
flask_api

Rubén Rodríguez Ramírez

Jan 05, 2021

CONTENTS

1	Installation	3
1.1	1. Linux packages	3
1.2	2. RabbitMQ configuration	3
1.3	3. Python dependencies	3
1.4	4. Domain configuration	4
1.5	5. Environment configuration	4
1.6	6. uWSGI configuration	4
1.7	7. Nginx configuration	4
1.8	8. Supervisor configuration	4
1.9	9. Log directories	5
1.10	10. Supervisor systemd unit file	5
1.11	How to usage	6
1.12	Optional installation	6
2	Skeleton app structure	7
2.1	app	8
2.2	database	261
2.3	tests	288
2.4	config	302
3	Flask command line	335
4	Changelog	337
4.1	2.0.0 (2020-10-27)	337
4.2	1.4.0 (2020-10-04)	338
4.3	1.3.0 (2020-09-20)	338
4.4	1.2.0 (2020-09-18)	338
4.5	1.1.0 (2020-05-31)	338
4.6	1.0.0 (2020-05-17)	339
4.7	0.8.0 (2020-04-29)	339
4.8	0.7.0 (2020-04-23)	340
5	Note	341
6	Indices and tables	343
	Python Module Index	345
	Index	347

Flask-api is a small API project for creating users and files (Microsoft Word and PDF). These files contain data about users registered in the project.

The project is developed in Python 3.7 and use next main libraries:

- : microframework.
- : SQL database engine.
- : simple and small ORM.
- : asynchronous task queue/job.
- : message broker.
- : web server, reverse proxy, etc.
- : Web Server Gateway Interface (WSGI) server implementation.
- : monitoring and administrating Celery clusters.
- : client/server system that allows its users to monitor and control a number of processes on UNIX-like operating systems.

INSTALLATION

1.1 1. Linux packages

These packages are required for the project installation:

```
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
sudo apt-get install autoconf build-essential cmake libcap-dev libffi-dev libpcre3-
↳dev librabbitmq-dev libreoffice-writer libtool libxml2-dev libxslt1-dev libxslt1.1_
↳pkg-config magic nginx python3-distutils python3.7 python3.7-dev python3.7-venv_
↳rabbitmq-server uuid-dev uwsgi uwsgi-src
sudo reboot
```

1.2 2. RabbitMQ configuration

Required plugins for monitoring our brokers in Flower:

```
sudo rabbitmq-plugins enable rabbitmq_management
sudo service rabbitmq-server restart
```

1.3 3. Python dependencies

Install Python dependencies:

```
python3.7 -m venv venv
source venv/bin/activate
pip install -r requirements.txt --no-cache-dir
```

1.4 4. Domain configuration

Add local domain to our */etc/hosts* file:

```
127.0.0.1 flask-api.prod
```

1.5 5. Environment configuration

Create a new **.env** file based on *.env.example* file.

1.6 6. uWSGI configuration

Create a new **uwsgi.ini** file based on *uwsgi.ini.example*.

username and *project_path* must to be filled with appropriate values.

www-data group must to be added to your user:

```
sudo usermod -a -G www-data username
```

1.7 7. Nginx configuration

Create a new **flask_api** file based on *docs/examples/flask_api.nginx.example* file.

Replace *uwsgi_pass* variable with the value in *socket* variable from **uwsgi.ini** file.

Move **flask_api** file to */etc/nginx/sites-available* directory:

```
sudo mv docs/examples/flask_api /etc/nginx/sites-available
sudo ln -s /etc/nginx/sites-available/flask_api /etc/nginx/sites-enabled/flask_api
sudo systemctl restart nginx
```

1.8 8. Supervisor configuration

1.8.1 8.1 Main configuration

Create a new **supervisord.conf** file based on *docs/examples/supervisor/supervisord.conf.example* file in the root project.

command, *directory* and *username* variables must to be filled with appropriate values. These variables are below *Mr Developer* comment.

1.8.2 8.2 Other configurations

Create a new directory named *supervisor* in the root path and create next files based on *docs/examples/supervisor* example files:

1. celery.conf
2. flower.conf
3. uwsgi.conf

username and *path* variables must to be replaced with appropriate values.

1.9 9. Log directories

Create next log directories:

1. log/app
2. log/celery
3. log/flower
4. log/uwsgi

1.10 10. Supervisor systemd unit file

Create a new **flask_api_supervisor.service** file based on *docs/examples/flask_api_supervisor.service.example* file.

username, *usergroup* and *path* variables must to be filled with appropriate values.

Move file to */etc/systemd/system* directory and we run next commands:

```
sudo systemctl enable flask_api_supervisor.service
sudo systemctl daemon-reload
sudo systemctl start flask_api_supervisor.service
```

The systemd unit file start up the project if the system is reboot or shutdown.

For checking process status in command line:

```
sudo systemctl status flask_api_supervisor.service
```

For restart all processes in command line:

```
sudo systemctl restart flask_api_supervisor.service
```

This command reread the supervisor configuration files, stop all processes and start them again.

1.11 How to usage

The setup is finished, we only need to create the database tables and fill them with fake data. We open a terminal in the root project and run next commands:

```
./venv/bin/flask init-db
./venv/bin/flask migrate
./venv/bin/flask seed
```

You can use an API client such as Insomnia or Postman and starting to consume the API!

You can see the processes status here: <http://flask-api.prod/supervisor>

The credentials are user:123 by default, you can change the credentials in supervisord.conf file in *inet_http_server* section.

You can management the Celery tasks status here: <http://flask-api.prod:5555/flower/>

1.12 Optional installation

This project use for logging configuration. The config file is already defined you only need to do these steps:

1. Create new **flask_api.logrotate** file based on *docs/examples/flask_api.logrotate.example*.
2. *path*, *username* and *usergroup* variables must to be filled with appropriate values.
3. Move flask_api_logrotate to */etc/logrotate.d*:

```
sudo mv docs/examples/flask_api.logrotate /etc/logrotate.d
```

4. Restart logrotate service:

```
sudo service log rotate restart
```

A new log file will be created every day.

SKELETON APP STRUCTURE

The project structure looks like this:

```
flask_api
├── /app
│   ├── /blueprints
│   │   └── ...
│   ├── /celery
│   │   └── ...
│   ├── /models
│   │   └── ...
│   ├── /templates
│   │   └── ...
│   ├── /utils
│   │   └── ...
│   ├── __init__.py
│   ├── extensions.py
│   └── middleware.py
├── /database
│   ├── factories
│   │   └── ...
│   ├── migrations
│   │   └── ...
│   ├── seeds
│   │   └── ...
│   └── __init__.py
├── /log
│   └── ...
├── /storage
│   └── ...
├── /tests
│   └── ...
├── config.py
├── manage.py
└── requirements.txt
```

<i>app</i>	Package for building a Flask application.
<i>database</i>	Package for managing the database.
<i>tests</i>	Package for testing the application.
<i>config</i>	Module loads the application's configuration.

2.1 app

Description

Package for building a Flask application.

The app package loads application configuration and registers middleware, blueprints, database models, etc.

Modules

<i>app.blueprints</i>	Registers Flask blueprints.
<i>app.celery</i>	Registers Celery tasks.
<i>app.extensions</i>	Registers third party extensions.
<i>app.middleware</i>	WSGI middleware for validating requests content type.
<i>app.models</i>	Registers database models.
<i>app.utils</i>	Collection of functions and classes which make common patterns shorter and easier.

2.1.1 app.blueprints

Description

Registers Flask blueprints.

Modules

<i>app.blueprints.auth</i>
<i>app.blueprints.base</i>
<i>app.blueprints.documents</i>
<i>app.blueprints.roles</i>
<i>app.blueprints.tasks</i>
<i>app.blueprints.users</i>

app.blueprints.auth

Description

Classes

<i>AuthUserLoginResource([api])</i>
<i>AuthUserLogoutResource([api])</i>
<i>RequestResetPasswordResource([api])</i>
<i>ResetPasswordResource([api])</i>

app.blueprints.auth.AuthUserLoginResource

```
class app.blueprints.auth.AuthUserLoginResource (api=None, *args, **kwargs)
    Bases: flask_restx.resource.Resource
```

Attributes

*AuthUserLoginResource.
decorators*

*AuthUserLoginResource.
method_decorators*

AuthUserLoginResource.methods

*AuthUserLoginResource.
provide_automatic_options*

*AuthUserLoginResource.
representations*

app.blueprints.auth.AuthUserLoginResource.decorators

```
AuthUserLoginResource.decorators = ()
```

app.blueprints.auth.AuthUserLoginResource.method_decorators

```
AuthUserLoginResource.method_decorators = []
```

app.blueprints.auth.AuthUserLoginResource.methods

```
AuthUserLoginResource.methods = {'POST'}
```

app.blueprints.auth.AuthUserLoginResource.provide_automatic_options

```
AuthUserLoginResource.provide_automatic_options = None
```

app.blueprints.auth.AuthUserLoginResource.representations

```
AuthUserLoginResource.representations = None
```

Methods

<code>AuthUserLoginResource.__init__([api])</code>	Initialize self.
<code>AuthUserLoginResource.as_view(name,...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>AuthUserLoginResource.dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>AuthUserLoginResource.post()</code>	
<code>AuthUserLoginResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

`app.blueprints.auth.AuthUserLoginResource.__init__`

`AuthUserLoginResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

`app.blueprints.auth.AuthUserLoginResource.as_view`

classmethod `AuthUserLoginResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

`app.blueprints.auth.AuthUserLoginResource.dispatch_request`

`AuthUserLoginResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

`app.blueprints.auth.AuthUserLoginResource.post`

`AuthUserLoginResource.post()` → tuple

`app.blueprints.auth.AuthUserLoginResource.validate_payload`

`AuthUserLoginResource.validate_payload(func)`
Perform a payload validation on expected model if necessary

app.blueprints.auth.AuthUserLogoutResource

```
class app.blueprints.auth.AuthUserLogoutResource (api=None, *args, **kwargs)
    Bases: flask_restx.resource.Resource
```

Attributes

*AuthUserLogoutResource.
decorators*

*AuthUserLogoutResource.
method_decorators*

AuthUserLogoutResource.methods

*AuthUserLogoutResource.
provide_automatic_options*

*AuthUserLogoutResource.
representations*

app.blueprints.auth.AuthUserLogoutResource.decorators

```
AuthUserLogoutResource.decorators = ()
```

app.blueprints.auth.AuthUserLogoutResource.method_decorators

```
AuthUserLogoutResource.method_decorators = []
```

app.blueprints.auth.AuthUserLogoutResource.methods

```
AuthUserLogoutResource.methods = {'POST'}
```

app.blueprints.auth.AuthUserLogoutResource.provide_automatic_options

```
AuthUserLogoutResource.provide_automatic_options = None
```

app.blueprints.auth.AuthUserLogoutResource.representations

```
AuthUserLogoutResource.representations = None
```

Methods

<code>AuthUserLogoutResource.__init__([api])</code>	Initialize self.
<code>AuthUserLogoutResource.as_view(name,...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>AuthUserLogoutResource.dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>AuthUserLogoutResource.post()</code>	
<code>AuthUserLogoutResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

`app.blueprints.auth.AuthUserLogoutResource.__init__`

`AuthUserLogoutResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

`app.blueprints.auth.AuthUserLogoutResource.as_view`

classmethod `AuthUserLogoutResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

`app.blueprints.auth.AuthUserLogoutResource.dispatch_request`

`AuthUserLogoutResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

`app.blueprints.auth.AuthUserLogoutResource.post`

`AuthUserLogoutResource.post()` → tuple

`app.blueprints.auth.AuthUserLogoutResource.validate_payload`

`AuthUserLogoutResource.validate_payload(func)`
Perform a payload validation on expected model if necessary

app.blueprints.auth.RequestResetPasswordResource

```
class app.blueprints.auth.RequestResetPasswordResource (api=None, *args,
                                                    **kwargs)
```

```
Bases: flask_restx.resource.Resource
```

Attributes

```
RequestResetPasswordResource.  
decorators
```

```
RequestResetPasswordResource.  
method_decorators
```

```
RequestResetPasswordResource.  
methods
```

```
RequestResetPasswordResource.  
provide_automatic_options
```

```
RequestResetPasswordResource.  
representations
```

app.blueprints.auth.RequestResetPasswordResource.decorators

```
RequestResetPasswordResource.decorators = ()
```

app.blueprints.auth.RequestResetPasswordResource.method_decorators

```
RequestResetPasswordResource.method_decorators = []
```

app.blueprints.auth.RequestResetPasswordResource.methods

```
RequestResetPasswordResource.methods = {'POST'}
```

app.blueprints.auth.RequestResetPasswordResource.provide_automatic_options

```
RequestResetPasswordResource.provide_automatic_options = None
```

app.blueprints.auth.RequestResetPasswordResource.representations

```
RequestResetPasswordResource.representations = None
```

Methods

<code>RequestResetPasswordResource.__init__([api])</code>	Initialize self.
<code>RequestResetPasswordResource.as_view(name,...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>RequestResetPasswordResource.dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>RequestResetPasswordResource.post()</code>	
<code>RequestResetPasswordResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

`app.blueprints.auth.RequestResetPasswordResource.__init__`

`RequestResetPasswordResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

`app.blueprints.auth.RequestResetPasswordResource.as_view`

classmethod `RequestResetPasswordResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

`app.blueprints.auth.RequestResetPasswordResource.dispatch_request`

`RequestResetPasswordResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

`app.blueprints.auth.RequestResetPasswordResource.post`

`RequestResetPasswordResource.post()` → tuple

`app.blueprints.auth.RequestResetPasswordResource.validate_payload`

`RequestResetPasswordResource.validate_payload(func)`
Perform a payload validation on expected model if necessary

app.blueprints.auth.ResetPasswordResource

```
class app.blueprints.auth.ResetPasswordResource (api=None, *args, **kwargs)
    Bases: flask_restx.resource.Resource
```

Attributes

*ResetPasswordResource.
decorators*

*ResetPasswordResource.
method_decorators*

ResetPasswordResource.methods

*ResetPasswordResource.
provide_automatic_options*

*ResetPasswordResource.
representations*

app.blueprints.auth.ResetPasswordResource.decorators

```
ResetPasswordResource.decorators = ()
```

app.blueprints.auth.ResetPasswordResource.method_decorators

```
ResetPasswordResource.method_decorators = []
```

app.blueprints.auth.ResetPasswordResource.methods

```
ResetPasswordResource.methods = {'GET', 'POST'}
```

app.blueprints.auth.ResetPasswordResource.provide_automatic_options

```
ResetPasswordResource.provide_automatic_options = None
```

app.blueprints.auth.ResetPasswordResource.representations

```
ResetPasswordResource.representations = None
```

Methods

<code>ResetPasswordResource.__init__([api])</code>	Initialize self.
<code>ResetPasswordResource.as_view(name,...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>ResetPasswordResource.dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>ResetPasswordResource.get(token)</code>	
<code>ResetPasswordResource.post(token)</code>	
<code>ResetPasswordResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

`app.blueprints.auth.ResetPasswordResource.__init__`

`ResetPasswordResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

`app.blueprints.auth.ResetPasswordResource.as_view`

classmethod `ResetPasswordResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

`app.blueprints.auth.ResetPasswordResource.dispatch_request`

`ResetPasswordResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

`app.blueprints.auth.ResetPasswordResource.get`

`ResetPasswordResource.get(token: str) → tuple`

`app.blueprints.auth.ResetPasswordResource.post`

`ResetPasswordResource.post(token: str) → tuple`

app.blueprints.auth.ResetPasswordResource.validate_payload

`ResetPasswordResource.validate_payload(func)`

Perform a payload validation on expected model if necessary

```
class app.blueprints.auth.AuthUserLoginResource (api=None, *args, **kwargs)
```

classmethod as_view (name, *class_args, **class_kwargs)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

decorators = ()

dispatch_request (*args, **kwargs)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

method_decorators = []

methods = {'POST'}

post () → tuple

provide_automatic_options = None

representations = None

validate_payload (func)

Perform a payload validation on expected model if necessary

```
class app.blueprints.auth.AuthUserLogoutResource (api=None, *args, **kwargs)
```

classmethod as_view (name, *class_args, **class_kwargs)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

decorators = ()

dispatch_request (*args, **kwargs)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

method_decorators = []

methods = {'POST'}

post () → tuple

provide_automatic_options = None

representations = None

validate_payload (func)

Perform a payload validation on expected model if necessary

```
class app.blueprints.auth.RequestResetPasswordResource (api=None, *args,
                                                         **kwargs)
```

classmethod as_view (*name*, **class_args*, ***class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

decorators = ()

dispatch_request (**args*, ***kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

method_decorators = []

methods = {'POST'}

post () → tuple

provide_automatic_options = None

representations = None

validate_payload (*func*)

Perform a payload validation on expected model if necessary

class app.blueprints.auth.**ResetPasswordResource** (*api=None*, **args*, ***kwargs*)

classmethod as_view (*name*, **class_args*, ***class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

decorators = ()

dispatch_request (**args*, ***kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

get (*token: str*) → tuple

method_decorators = []

methods = {'GET', 'POST'}

post (*token: str*) → tuple

provide_automatic_options = None

representations = None

validate_payload (*func*)

Perform a payload validation on expected model if necessary

app.blueprints.base**Description****Classes**

BaseResource([api])

WelcomeResource([api])

app.blueprints.base.BaseResource

class app.blueprints.base.**BaseResource** (*api=None, *args, **kwargs*)
Bases: flask_restx.resource.Resource

Attributes

BaseResource.decorators

BaseResource.method_decorators

BaseResource.methods

BaseResource.

provide_automatic_options

BaseResource.representations

app.blueprints.base.BaseResource.decorators

BaseResource.**decorators** = ()

app.blueprints.base.BaseResource.method_decorators

BaseResource.**method_decorators** = []

app.blueprints.base.BaseResource.methods

BaseResource.**methods** = None

app.blueprints.base.BaseResource.provide_automatic_options

BaseResource.**provide_automatic_options** = None

app.blueprints.base.BaseResource.representations

`BaseResource.representations = None`

Methods

<code>BaseResource.__init__([api])</code>	Initialize self.
<code>BaseResource.as_view(name, *class_args, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>BaseResource.create_search_query(query, ...)</code>	
<code>BaseResource.dispatch_request(*args, **kwargs)</code>	Subclasses have to override this method to implement the actual view function code.
<code>BaseResource.get_request_query_fields(...)</code>	
<code>BaseResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.base.BaseResource.__init__

`BaseResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.base.BaseResource.as_view

classmethod `BaseResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.base.BaseResource.create_search_query

`BaseResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.base.BaseResource.dispatch_request

`BaseResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.base.BaseResource.get_request_query_fields

`BaseResource.get_request_query_fields(request_data: dict) → tuple`

app.blueprints.base.BaseResource.validate_payload

`BaseResource.validate_payload(func)`
 Perform a payload validation on expected model if necessary

app.blueprints.base.WelcomeResource

class `app.blueprints.base.WelcomeResource` (*api=None, *args, **kwargs*)
 Bases: `flask_restx.resource.Resource`

Attributes

<code>WelcomeResource.decorators</code>
<code>WelcomeResource.method_decorators</code>
<code>WelcomeResource.methods</code>
<code>WelcomeResource.provide_automatic_options</code>
<code>WelcomeResource.representations</code>

app.blueprints.base.WelcomeResource.decorators

`WelcomeResource.decorators = ()`

app.blueprints.base.WelcomeResource.method_decorators

`WelcomeResource.method_decorators = []`

app.blueprints.base.WelcomeResource.methods

`WelcomeResource.methods = {'GET'}`

app.blueprints.base.WelcomeResource.provide_automatic_options

`WelcomeResource.provide_automatic_options = None`

app.blueprints.base>WelcomeResource.representations

WelcomeResource.**representations** = None

Methods

<i>WelcomeResource.__init__([api])</i>	Initialize self.
<i>WelcomeResource.as_view(name, *class_args, ...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>WelcomeResource.dispatch_request(*args, **kwargs)</i>	Subclasses have to override this method to implement the actual view function code.
<i>WelcomeResource.get()</i>	
<i>WelcomeResource.validate_payload(func)</i>	Perform a payload validation on expected model if necessary

app.blueprints.base>WelcomeResource.__init__

WelcomeResource.**__init__**(*api=None, *args, **kwargs*)
Initialize self. See help(type(self)) for accurate signature.

app.blueprints.base>WelcomeResource.as_view

classmethod WelcomeResource.**as_view**(*name, *class_args, **class_kwargs*)
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the View on each request and call the *dispatch_request()* method on it.

The arguments passed to *as_view()* are forwarded to the constructor of the class.

app.blueprints.base>WelcomeResource.dispatch_request

WelcomeResource.**dispatch_request**(**args, **kwargs*)
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.base>WelcomeResource.get

WelcomeResource.**get**() → tuple

app.blueprints.base.WelcomeResource.validate_payload

`WelcomeResource.validate_payload(func)`

Perform a payload validation on expected model if necessary

```
class app.blueprints.base.BaseResource (api=None, *args, **kwargs)
```

```
classmethod as_view (name, *class_args, **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the View on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

```
create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect
```

```
db_model
```

alias of `playhouse.flask_utils.FlaskDB.get_model_class.<locals>.BaseModel`

```
decorators = ()
```

```
dispatch_request (*args, **kwargs)
```

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

```
get_request_query_fields (request_data: dict) → tuple
```

```
method_decorators = []
```

```
methods = None
```

```
provide_automatic_options = None
```

```
representations = None
```

```
validate_payload (func)
```

Perform a payload validation on expected model if necessary

```
class app.blueprints.base.WelcomeResource (api=None, *args, **kwargs)
```

```
classmethod as_view (name, *class_args, **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the View on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

```
decorators = ()
```

```
dispatch_request (*args, **kwargs)
```

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

```
get () → tuple
```

```
method_decorators = []
```

```
methods = {'GET'}
```

```
provide_automatic_options = None
```

```
representations = None
```

validate_payload (*func*)

Perform a payload validation on expected model if necessary

app.blueprints.documents

Description

Classes

DocumentBaseResource([api])

DocumentResource([api])

NewDocumentResource([api])

SearchDocumentResource([api])

app.blueprints.documents.DocumentBaseResource

class app.blueprints.documents.**DocumentBaseResource** (*api=None, *args, **kwargs*)

Bases: *app.blueprints.base.BaseResource*

Attributes

DocumentBaseResource.decorators

DocumentBaseResource.

document_serializer

DocumentBaseResource.

method_decorators

DocumentBaseResource.methods

DocumentBaseResource.

provide_automatic_options

DocumentBaseResource.

representations

DocumentBaseResource.

request_field_name

app.blueprints.documents.DocumentBaseResource.decorators

DocumentBaseResource.**decorators** = ()

app.blueprints.documents.DocumentBaseResource.document_serializer

```
DocumentBaseResource.document_serializer = <DocumentSchema (many=False)>
```

app.blueprints.documents.DocumentBaseResource.method_decorators

```
DocumentBaseResource.method_decorators = []
```

app.blueprints.documents.DocumentBaseResource.methods

```
DocumentBaseResource.methods = None
```

app.blueprints.documents.DocumentBaseResource.provide_automatic_options

```
DocumentBaseResource.provide_automatic_options = None
```

app.blueprints.documents.DocumentBaseResource.representations

```
DocumentBaseResource.representations = None
```

app.blueprints.documents.DocumentBaseResource.request_field_name

```
DocumentBaseResource.request_field_name = 'document'
```

Methods

<i>DocumentBaseResource. __init__([api])</i>	Initialize self.
<i>DocumentBaseResource. as_view(name, ...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>DocumentBaseResource. create_search_query(...)</i>	
<i>DocumentBaseResource. dispatch_request(*args, ...)</i>	Subclasses have to override this method to implement the actual view function code.
<i>DocumentBaseResource. get_document_content(...)</i>	
<i>DocumentBaseResource. get_document_data(...)</i>	
<i>DocumentBaseResource. get_request_file()</i>	
<i>DocumentBaseResource. get_request_query_fields(...)</i>	
<i>DocumentBaseResource. validate_payload(func)</i>	Perform a payload validation on expected model if necessary

app.blueprints.documents.DocumentBaseResource.__init__

`DocumentBaseResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.documents.DocumentBaseResource.as_view

classmethod `DocumentBaseResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.documents.DocumentBaseResource.create_search_query

`DocumentBaseResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.documents.DocumentBaseResource.dispatch_request

`DocumentBaseResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.documents.DocumentBaseResource.get_document_content

static `DocumentBaseResource.get_document_content(document_id: int)`

app.blueprints.documents.DocumentBaseResource.get_document_data

`DocumentBaseResource.get_document_data(document_id: int) → tuple`

app.blueprints.documents.DocumentBaseResource.get_request_file

`DocumentBaseResource.get_request_file() → dict`

app.blueprints.documents.DocumentBaseResource.get_request_query_fields

`DocumentBaseResource.get_request_query_fields` (*request_data: dict*) → tuple

app.blueprints.documents.DocumentBaseResource.validate_payload

`DocumentBaseResource.validate_payload` (*func*)
Perform a payload validation on expected model if necessary

app.blueprints.documents.DocumentResource

class `app.blueprints.documents.DocumentResource` (*api=None, *args, **kwargs*)
Bases: `app.blueprints.documents.DocumentBaseResource`

Attributes

`DocumentResource.decorators`

`DocumentResource.`
`document_serializer`

`DocumentResource.`
`method_decorators`

`DocumentResource.methods`

`DocumentResource.`
`provide_automatic_options`

`DocumentResource.`
`representations`

`DocumentResource.`
`request_field_name`

app.blueprints.documents.DocumentResource.decorators

`DocumentResource.decorators` = ()

app.blueprints.documents.DocumentResource.document_serializer

```
DocumentResource.document_serializer = <DocumentSchema(many=False)>
```

app.blueprints.documents.DocumentResource.method_decorators

```
DocumentResource.method_decorators = []
```

app.blueprints.documents.DocumentResource.methods

```
DocumentResource.methods = {'DELETE', 'GET', 'PUT'}
```

app.blueprints.documents.DocumentResource.provide_automatic_options

```
DocumentResource.provide_automatic_options = None
```

app.blueprints.documents.DocumentResource.representations

```
DocumentResource.representations = None
```

app.blueprints.documents.DocumentResource.request_field_name

```
DocumentResource.request_field_name = 'document'
```

Methods

<i>DocumentResource.__init__([api])</i>	Initialize self.
<i>DocumentResource.as_view(name, *class_args, ...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>DocumentResource.create_search_query(query, ...)</i>	
<i>DocumentResource.delete(document_id)</i>	
<i>DocumentResource.dispatch_request(*args, ...)</i>	Subclasses have to override this method to implement the actual view function code.
<i>DocumentResource.get(document_id)</i>	
<i>DocumentResource.get_document_content(...)</i>	
<i>DocumentResource.get_document_data(document_id)</i>	
<i>DocumentResource.get_request_file()</i>	
<i>DocumentResource.get_request_query_fields(...)</i>	
<i>DocumentResource.put(document_id)</i>	

continues on next page

Table 22 – continued from previous page

<code>DocumentResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary
--	---

app.blueprints.documents.DocumentResource.__init__

`DocumentResource.__init__(api=None, *args, **kwargs)`
 Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.documents.DocumentResource.as_view

classmethod `DocumentResource.as_view(name, *class_args, **class_kwargs)`
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.documents.DocumentResource.create_search_query

`DocumentResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.documents.DocumentResource.delete

`DocumentResource.delete(document_id: int)`

app.blueprints.documents.DocumentResource.dispatch_request

`DocumentResource.dispatch_request(*args, **kwargs)`
 Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.documents.DocumentResource.get

`DocumentResource.get(document_id: int) → tuple`

app.blueprints.documents.DocumentResource.get_document_content

static `DocumentResource.get_document_content(document_id: int)`

app.blueprints.documents.DocumentResource.get_document_data

DocumentResource.get_document_data (document_id: int) → tuple

app.blueprints.documents.DocumentResource.get_request_file

DocumentResource.get_request_file () → dict

app.blueprints.documents.DocumentResource.get_request_query_fields

DocumentResource.get_request_query_fields (request_data: dict) → tuple

app.blueprints.documents.DocumentResource.put

DocumentResource.put (document_id: int) → tuple

app.blueprints.documents.DocumentResource.validate_payload

DocumentResource.validate_payload (func)
Perform a payload validation on expected model if necessary

app.blueprints.documents.NewDocumentResource

class app.blueprints.documents.NewDocumentResource (api=None, *args, **kwargs)

Bases: *app.blueprints.documents.DocumentBaseResource*

Attributes

NewDocumentResource.decorators

NewDocumentResource.
document_serializer

NewDocumentResource.
method_decorators

NewDocumentResource.methods

NewDocumentResource.parser

NewDocumentResource.
provide_automatic_options

NewDocumentResource.
representations

NewDocumentResource.
request_field_name

app.blueprints.documents.NewDocumentResource.decorators

```
NewDocumentResource.decorators = ()
```

app.blueprints.documents.NewDocumentResource.document_serializer

```
NewDocumentResource.document_serializer = <DocumentSchema(many=False)>
```

app.blueprints.documents.NewDocumentResource.method_decorators

```
NewDocumentResource.method_decorators = []
```

app.blueprints.documents.NewDocumentResource.methods

```
NewDocumentResource.methods = {'POST'}
```

app.blueprints.documents.NewDocumentResource.parser

```
NewDocumentResource.parser = <flask_restx.reqparse.RequestParser object>
```

app.blueprints.documents.NewDocumentResource.provide_automatic_options

```
NewDocumentResource.provide_automatic_options = None
```

app.blueprints.documents.NewDocumentResource.representations

```
NewDocumentResource.representations = None
```

app.blueprints.documents.NewDocumentResource.request_field_name

```
NewDocumentResource.request_field_name = 'document'
```

Methods

<i>NewDocumentResource.__init__([api])</i>	Initialize self.
<i>NewDocumentResource.as_view(name, ...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>NewDocumentResource.create_search_query(...)</i>	
<i>NewDocumentResource.dispatch_request(*args, ...)</i>	Subclasses have to override this method to implement the actual view function code.
<i>NewDocumentResource.get_document_content(...)</i>	

continues on next page

Table 24 – continued from previous page

<code>NewDocumentResource.</code>	
<code>get_document_data(...)</code>	
<code>NewDocumentResource.</code>	
<code>get_request_file()</code>	
<code>NewDocumentResource.</code>	
<code>get_request_query_fields(...)</code>	
<code>NewDocumentResource.post()</code>	
<code>NewDocumentResource.</code>	Perform a payload validation on expected
<code>validate_payload(func)</code>	model if necessary

app.blueprints.documents.NewDocumentResource.__init__

`NewDocumentResource.__init__` (*api=None*, *args, **kwargs)
 Initialize self. See help(type(self)) for accurate signature.

app.blueprints.documents.NewDocumentResource.as_view

classmethod `NewDocumentResource.as_view` (*name*, *class_args, **class_kwargs)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.documents.NewDocumentResource.create_search_query

`NewDocumentResource.create_search_query` (*query: peewee.ModelSelect*,
request_data: dict) → *peewee.ModelSelect*

app.blueprints.documents.NewDocumentResource.dispatch_request

`NewDocumentResource.dispatch_request` (*args, **kwargs)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.documents.NewDocumentResource.get_document_content

static `NewDocumentResource.get_document_content` (*document_id: int*)

app.blueprints.documents.NewDocumentResource.get_document_data

`NewDocumentResource.get_document_data (document_id: int) → tuple`

app.blueprints.documents.NewDocumentResource.get_request_file

`NewDocumentResource.get_request_file () → dict`

app.blueprints.documents.NewDocumentResource.get_request_query_fields

`NewDocumentResource.get_request_query_fields (request_data: dict) → tuple`

app.blueprints.documents.NewDocumentResource.post

`NewDocumentResource.post ()`

app.blueprints.documents.NewDocumentResource.validate_payload

`NewDocumentResource.validate_payload (func)`
Perform a payload validation on expected model if necessary

app.blueprints.documents.SearchDocumentResource

class `app.blueprints.documents.SearchDocumentResource (api=None, *args, **kwargs)`
Bases: `app.blueprints.documents.DocumentBaseResource`

Attributes

`SearchDocumentResource.decorators`

`SearchDocumentResource.document_serializer`

`SearchDocumentResource.method_decorators`

`SearchDocumentResource.methods`

`SearchDocumentResource.provide_automatic_options`

`SearchDocumentResource.representations`

`SearchDocumentResource.request_field_name`

app.blueprints.documents.SearchDocumentResource.decorators

```
SearchDocumentResource.decorators = ()
```

app.blueprints.documents.SearchDocumentResource.document_serializer

```
SearchDocumentResource.document_serializer = <DocumentSchema(many=False)>
```

app.blueprints.documents.SearchDocumentResource.method_decorators

```
SearchDocumentResource.method_decorators = []
```

app.blueprints.documents.SearchDocumentResource.methods

```
SearchDocumentResource.methods = {'POST'}
```

app.blueprints.documents.SearchDocumentResource.provide_automatic_options

```
SearchDocumentResource.provide_automatic_options = None
```

app.blueprints.documents.SearchDocumentResource.representations

```
SearchDocumentResource.representations = None
```

app.blueprints.documents.SearchDocumentResource.request_field_name

```
SearchDocumentResource.request_field_name = 'document'
```

Methods

<i>SearchDocumentResource. __init__([api])</i>	Initialize self.
<i>SearchDocumentResource. as_view(name, ...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>SearchDocumentResource. create_search_query(...)</i>	
<i>SearchDocumentResource. dispatch_request(...)</i>	Subclasses have to override this method to implement the actual view function code.
<i>SearchDocumentResource. get_document_content(...)</i>	
<i>SearchDocumentResource. get_document_data(...)</i>	
<i>SearchDocumentResource. get_request_file()</i>	

continues on next page

Table 26 – continued from previous page

<code>SearchDocumentResource.get_request_query_fields(...)</code>	
<code>SearchDocumentResource.post()</code>	
<code>SearchDocumentResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.documents.SearchDocumentResource.__init__

`SearchDocumentResource.__init__(api=None, *args, **kwargs)`
 Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.documents.SearchDocumentResource.as_view

classmethod `SearchDocumentResource.as_view(name, *class_args, **class_kwargs)`
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
 The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.documents.SearchDocumentResource.create_search_query

`SearchDocumentResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.documents.SearchDocumentResource.dispatch_request

`SearchDocumentResource.dispatch_request(*args, **kwargs)`
 Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.documents.SearchDocumentResource.get_document_content

static `SearchDocumentResource.get_document_content(document_id: int)`

app.blueprints.documents.SearchDocumentResource.get_document_data

`SearchDocumentResource.get_document_data(document_id: int) → tuple`

app.blueprints.documents.SearchDocumentResource.get_request_file

`SearchDocumentResource.get_request_file()` → dict

app.blueprints.documents.SearchDocumentResource.get_request_query_fields

`SearchDocumentResource.get_request_query_fields(request_data: dict)` → tuple

app.blueprints.documents.SearchDocumentResource.post

`SearchDocumentResource.post()`

app.blueprints.documents.SearchDocumentResource.validate_payload

`SearchDocumentResource.validate_payload(func)`

Perform a payload validation on expected model if necessary

class `app.blueprints.documents.DocumentBaseResource` (*api=None, *args, **kwargs*)

classmethod `as_view` (*name, *class_args, **class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

create_search_query (*query: peewee.ModelSelect, request_data: dict*) → `peewee.ModelSelect`

db_model

alias of `app.models.document.Document`

decorators = ()

dispatch_request (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

document_serializer = `<DocumentSchema(many=False)>`

static `get_document_content` (*document_id: int*)

`get_document_data` (*document_id: int*) → tuple

`get_request_file()` → dict

`get_request_query_fields` (*request_data: dict*) → tuple

method_decorators = []

methods = None

provide_automatic_options = None

representations = None

request_field_name = 'document'


```

validate_payload (func)
    Perform a payload validation on expected model if necessary

class app.blueprints.documents.DocumentResource (api=None, *args, **kwargs)

    _parser = <flask_restx.reqparse.RequestParser object>

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

    db_model
        alias of app.models.document.Document

    decorators = ()

    delete (document_id: int)

    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    document_serializer = <DocumentSchema (many=False)>

    get (document_id: int) → tuple

    static get_document_content (document_id: int)

    get_document_data (document_id: int) → tuple

    get_request_file () → dict

    get_request_query_fields (request_data: dict) → tuple

    method_decorators = []

    methods = {'DELETE', 'GET', 'PUT'}

    provide_automatic_options = None

    put (document_id: int) → tuple

    representations = None

    request_field_name = 'document'

    validate_payload (func)
        Perform a payload validation on expected model if necessary

class app.blueprints.documents.NewDocumentResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

```

```
db_model
    alias of app.models.document.Document

decorators = ()

dispatch_request (*args, **kwargs)
    Subclasses have to override this method to implement the actual view function code. This method is called
    with all the arguments from the URL rule.

document_serializer = <DocumentSchema (many=False)>

static get_document_content (document_id: int)

get_document_data (document_id: int) → tuple

get_request_file () → dict

get_request_query_fields (request_data: dict) → tuple

method_decorators = []

methods = {'POST'}

parser = <flask_restx.reqparse.RequestParser object>

post ()

provide_automatic_options = None

representations = None

request_field_name = 'document'

validate_payload (func)
    Perform a payload validation on expected model if necessary

class app.blueprints.documents.SearchDocumentResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

    db_model
        alias of app.models.document.Document

    decorators = ()

    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    document_serializer = <DocumentSchema (many=False)>

    static get_document_content (document_id: int)

    get_document_data (document_id: int) → tuple

    get_request_file () → dict

    get_request_query_fields (request_data: dict) → tuple

    method_decorators = []
```

```

methods = {'POST'}

post()

provide_automatic_options = None

representations = None

request_field_name = 'document'

validate_payload(func)
    Perform a payload validation on expected model if necessary

```

app.blueprints.roles

Description

Classes

<i>NewRoleResource</i> ([api])
<i>RoleBaseResource</i> ([api])
<i>RoleResource</i> ([api])
<i>RolesSearchResource</i> ([api])

app.blueprints.roles.NewRoleResource

class app.blueprints.roles.**NewRoleResource** (*api=None, *args, **kwargs*)
 Bases: *app.blueprints.roles.RoleBaseResource*

Attributes

<i>NewRoleResource.decorators</i>
<i>NewRoleResource.method_decorators</i>
<i>NewRoleResource.methods</i>
<i>NewRoleResource.provide_automatic_options</i>
<i>NewRoleResource.representations</i>
<i>NewRoleResource.role_serializer</i>

app.blueprints.roles.NewRoleResource.decorators

```
NewRoleResource.decorators = ()
```

app.blueprints.roles.NewRoleResource.method_decorators

```
NewRoleResource.method_decorators = []
```

app.blueprints.roles.NewRoleResource.methods

```
NewRoleResource.methods = {'POST'}
```

app.blueprints.roles.NewRoleResource.provide_automatic_options

```
NewRoleResource.provide_automatic_options = None
```

app.blueprints.roles.NewRoleResource.representations

```
NewRoleResource.representations = None
```

app.blueprints.roles.NewRoleResource.role_serializer

```
NewRoleResource.role_serializer = <RoleSchema(many=False)>
```

Methods

<code>NewRoleResource.__init__([api])</code>	Initialize self.
<code>NewRoleResource.as_view(name, *class_args, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>NewRoleResource.create_search_query(query, ...)</code>	
<code>NewRoleResource.deserialize_request_data(...)</code>	
<code>NewRoleResource.dispatch_request(*args, **kwargs)</code>	Subclasses have to override this method to implement the actual view function code.
<code>NewRoleResource.get_request_query_fields(...)</code>	
<code>NewRoleResource.post()</code>	
<code>NewRoleResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.roles.NewRoleResource.__init__

`NewRoleResource.__init__` (*api=None, *args, **kwargs*)
 Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.roles.NewRoleResource.as_view

classmethod `NewRoleResource.as_view` (*name, *class_args, **class_kwargs*)
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.roles.NewRoleResource.create_search_query

`NewRoleResource.create_search_query` (*query: peewee.ModelSelect, request_data: dict*) → `peewee.ModelSelect`

app.blueprints.roles.NewRoleResource.deserialize_request_data

`NewRoleResource.deserialize_request_data` (***kwargs: dict*) → `dict`

app.blueprints.roles.NewRoleResource.dispatch_request

`NewRoleResource.dispatch_request` (**args, **kwargs*)
 Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.roles.NewRoleResource.get_request_query_fields

`NewRoleResource.get_request_query_fields` (*request_data: dict*) → `tuple`

app.blueprints.roles.NewRoleResource.post

`NewRoleResource.post` () → `tuple`

app.blueprints.roles.NewRoleResource.validate_payload

`NewRoleResource.validate_payload` (*func*)
 Perform a payload validation on expected model if necessary

app.blueprints.roles.RoleBaseResource

```
class app.blueprints.roles.RoleBaseResource (api=None, *args, **kwargs)
    Bases: app.blueprints.base.BaseResource
```

Attributes

RoleBaseResource.decorators

RoleBaseResource.
method_decorators

RoleBaseResource.methods

RoleBaseResource.
provide_automatic_options

RoleBaseResource.
representations

RoleBaseResource.
role_serializer

app.blueprints.roles.RoleBaseResource.decorators

```
RoleBaseResource.decorators = ()
```

app.blueprints.roles.RoleBaseResource.method_decorators

```
RoleBaseResource.method_decorators = []
```

app.blueprints.roles.RoleBaseResource.methods

```
RoleBaseResource.methods = None
```

app.blueprints.roles.RoleBaseResource.provide_automatic_options

```
RoleBaseResource.provide_automatic_options = None
```

app.blueprints.roles.RoleBaseResource.representations

```
RoleBaseResource.representations = None
```

app.blueprints.roles.RoleBaseResource.role_serializer

```
RoleBaseResource.role_serializer = <RoleSchema(many=False)>
```

Methods

<i>RoleBaseResource.__init__</i> ([api])	Initialize self.
<i>RoleBaseResource.as_view</i> (name, *class_args, ...)	Converts the class into an actual view function that can be used with the routing system.
<i>RoleBaseResource.create_search_query</i> (query, ...)	
<i>RoleBaseResource.deserialize_request_data</i> (...)	
<i>RoleBaseResource.dispatch_request</i> (*args, ...)	Subclasses have to override this method to implement the actual view function code.
<i>RoleBaseResource.get_request_query_fields</i> (...)	
<i>RoleBaseResource.validate_payload</i> (func)	Perform a payload validation on expected model if necessary

app.blueprints.roles.RoleBaseResource.__init__

```
RoleBaseResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.roles.RoleBaseResource.as_view

classmethod *RoleBaseResource.as_view*(name, *class_args, **class_kwargs)
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the *View* on each request and call the *dispatch_request()* method on it.

The arguments passed to *as_view()* are forwarded to the constructor of the class.

app.blueprints.roles.RoleBaseResource.create_search_query

```
RoleBaseResource.create_search_query(query: peewee.ModelSelect,
                                     request_data: dict) → peewee.ModelSelect
```

app.blueprints.roles.RoleBaseResource.deserialize_request_data

```
RoleBaseResource.deserialize_request_data(**kwargs: dict) → dict
```

app.blueprints.roles.RoleBaseResource.dispatch_request

```
RoleBaseResource.dispatch_request(*args, **kwargs)
```

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.roles.RoleBaseResource.get_request_query_fields

```
RoleBaseResource.get_request_query_fields(request_data: dict) → tuple
```

app.blueprints.roles.RoleBaseResource.validate_payload

```
RoleBaseResource.validate_payload(func)
```

Perform a payload validation on expected model if necessary

app.blueprints.roles.RoleResource

```
class app.blueprints.roles.RoleResource(api=None, *args, **kwargs)
```

Bases: *app.blueprints.roles.RoleBaseResource*

Attributes

RoleResource.decorators

RoleResource.method_decorators

RoleResource.methods

RoleResource.

provide_automatic_options

RoleResource.representations

RoleResource.role_serializer

app.blueprints.roles.RoleResource.decorators

```
RoleResource.decorators = ()
```

app.blueprints.roles.RoleResource.method_decorators

```
RoleResource.method_decorators = []
```

app.blueprints.roles.RoleResource.methods

```
RoleResource.methods = {'DELETE', 'GET', 'PUT'}
```

app.blueprints.roles.RoleResource.provide_automatic_options

```
RoleResource.provide_automatic_options = None
```

app.blueprints.roles.RoleResource.representations

```
RoleResource.representations = None
```

app.blueprints.roles.RoleResource.role_serializer

```
RoleResource.role_serializer = <RoleSchema(many=False)>
```

Methods

<i>RoleResource.__init__</i> ([api])	Initialize self.
<i>RoleResource.as_view</i> (name, *class_args, ...)	Converts the class into an actual view function that can be used with the routing system.
<i>RoleResource.create_search_query</i> (query, ...)	
<i>RoleResource.delete</i> (role_id)	
<i>RoleResource.deserialize_request_data</i> (**kwargs)	
<i>RoleResource.dispatch_request</i> (*args, **kwargs)	Subclasses have to override this method to implement the actual view function code.
<i>RoleResource.get</i> (role_id)	
<i>RoleResource.get_request_query_fields</i> (...)	
<i>RoleResource.put</i> (role_id)	
<i>RoleResource.validate_payload</i> (func)	Perform a payload validation on expected model if necessary

app.blueprints.roles.RoleResource.__init__

`RoleResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.roles.RoleResource.as_view

classmethod `RoleResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.roles.RoleResource.create_search_query

`RoleResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.roles.RoleResource.delete

`RoleResource.delete(role_id: int) → tuple`

app.blueprints.roles.RoleResource.deserialize_request_data

`RoleResource.deserialize_request_data(**kwargs: dict) → dict`

app.blueprints.roles.RoleResource.dispatch_request

`RoleResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.roles.RoleResource.get

`RoleResource.get(role_id: int) → tuple`

app.blueprints.roles.RoleResource.get_request_query_fields

`RoleResource.get_request_query_fields(request_data: dict) → tuple`

app.blueprints.roles.RoleResource.put

RoleResource.**put** (*role_id: int*) → tuple

app.blueprints.roles.RoleResource.validate_payload

RoleResource.**validate_payload** (*func*)
 Perform a payload validation on expected model if necessary

app.blueprints.roles.RolesSearchResource

class app.blueprints.roles.**RolesSearchResource** (*api=None, *args, **kwargs*)
 Bases: *app.blueprints.roles.RoleBaseResource*

Attributes

RolesSearchResource.decorators

*RolesSearchResource.
 method_decorators*

RolesSearchResource.methods

*RolesSearchResource.
 provide_automatic_options*

*RolesSearchResource.
 representations*

*RolesSearchResource.
 role_serializer*

app.blueprints.roles.RolesSearchResource.decorators

RolesSearchResource.**decorators** = ()

app.blueprints.roles.RolesSearchResource.method_decorators

RolesSearchResource.**method_decorators** = []

app.blueprints.roles.RolesSearchResource.methods

RolesSearchResource.**methods** = {'POST'}

app.blueprints.roles.RolesSearchResource.provide_automatic_options

```
RolesSearchResource.provide_automatic_options = None
```

app.blueprints.roles.RolesSearchResource.representations

```
RolesSearchResource.representations = None
```

app.blueprints.roles.RolesSearchResource.role_serializer

```
RolesSearchResource.role_serializer = <RoleSchema(many=False)>
```

Methods

<i>RolesSearchResource.</i> <i>__init__</i> (<i>[api]</i>)	Initialize self.
<i>RolesSearchResource.</i> <i>as_view</i> (<i>name</i> ,...)	Converts the class into an actual view function that can be used with the routing system.
<i>RolesSearchResource.</i> <i>create_search_query</i> (...)	
<i>RolesSearchResource.</i> <i>deserialize_request_data</i> (...)	
<i>RolesSearchResource.</i> <i>dispatch_request</i> (*args,...)	Subclasses have to override this method to implement the actual view function code.
<i>RolesSearchResource.</i> <i>get_request_query_fields</i> (...)	
<i>RolesSearchResource.</i> <i>post</i> ()	
<i>RolesSearchResource.</i> <i>validate_payload</i> (<i>func</i>)	Perform a payload validation on expected model if necessary

app.blueprints.roles.RolesSearchResource.__init__

```
RolesSearchResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.roles.RolesSearchResource.as_view

```
classmethod RolesSearchResource.as_view(name, *class_args,
                                         **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.roles.RolesSearchResource.create_search_query

`RolesSearchResource.create_search_query` (*query: peewee.ModelSelect, request_data: dict*) → *peewee.ModelSelect*

app.blueprints.roles.RolesSearchResource.deserialize_request_data

`RolesSearchResource.deserialize_request_data` (***kwargs: dict*) → *dict*

app.blueprints.roles.RolesSearchResource.dispatch_request

`RolesSearchResource.dispatch_request` (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.roles.RolesSearchResource.get_request_query_fields

`RolesSearchResource.get_request_query_fields` (*request_data: dict*) → *tuple*

app.blueprints.roles.RolesSearchResource.post

`RolesSearchResource.post` () → *tuple*

app.blueprints.roles.RolesSearchResource.validate_payload

`RolesSearchResource.validate_payload` (*func*)

Perform a payload validation on expected model if necessary

class `app.blueprints.roles.NewRoleResource` (*api=None, *args, **kwargs*)

classmethod `as_view` (*name, *class_args, **class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

create_search_query (*query: peewee.ModelSelect, request_data: dict*) → *peewee.ModelSelect*

db_model

alias of `app.models.role.Role`

decorators = ()

deserialize_request_data (***kwargs: dict*) → *dict*

dispatch_request (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

```
get_request_query_fields (request_data: dict) → tuple
method_decorators = []
methods = {'POST'}
post () → tuple
provide_automatic_options = None
representations = None
role_serializer = <RoleSchema (many=False)>
validate_payload (func)
    Perform a payload validation on expected model if necessary
class app.blueprints.roles.RoleBaseResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect
    db_model
        alias of app.models.role.Role
    decorators = ()
    deserialize_request_data (**kwargs: dict) → dict
    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    get_request_query_fields (request_data: dict) → tuple
    method_decorators = []
    methods = None
    provide_automatic_options = None
    representations = None
    role_serializer = <RoleSchema (many=False)>
    validate_payload (func)
        Perform a payload validation on expected model if necessary
class app.blueprints.roles.RoleResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect
```

```

db_model
    alias of app.models.role.Role

decorators = ()

delete (role_id: int) → tuple

deserialize_request_data (**kwargs: dict) → dict

dispatch_request (*args, **kwargs)
    Subclasses have to override this method to implement the actual view function code. This method is called
    with all the arguments from the URL rule.

get (role_id: int) → tuple

get_request_query_fields (request_data: dict) → tuple

method_decorators = []

methods = {'DELETE', 'GET', 'PUT'}

provide_automatic_options = None

put (role_id: int) → tuple

representations = None

role_serializer = <RoleSchema (many=False)>

validate_payload (func)
    Perform a payload validation on expected model if necessary

class app.blueprints.roles.RolesSearchResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

    db_model
        alias of app.models.role.Role

    decorators = ()

    deserialize_request_data (**kwargs: dict) → dict

    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    get_request_query_fields (request_data: dict) → tuple

    method_decorators = []

    methods = {'POST'}

    post () → tuple

    provide_automatic_options = None

    representations = None

    role_serializer = <RoleSchema (many=False)>

```

validate_payload (*func*)

Perform a payload validation on expected model if necessary

app.blueprints.tasks

Description

Classes

TaskResource([api])

TaskStatusResource([api])

app.blueprints.tasks.TaskResource

class app.blueprints.tasks.**TaskResource** (*api=None, *args, **kwargs*)

Bases: flask_restx.resource.Resource

Attributes

TaskResource.decorators

TaskResource.method_decorators

TaskResource.methods

TaskResource.

provide_automatic_options

TaskResource.representations

app.blueprints.tasks.TaskResource.decorators

TaskResource.**decorators** = ()

app.blueprints.tasks.TaskResource.method_decorators

TaskResource.**method_decorators** = []

app.blueprints.tasks.TaskResource.methods

`TaskResource.methods = None`

app.blueprints.tasks.TaskResource.provide_automatic_options

`TaskResource.provide_automatic_options = None`

app.blueprints.tasks.TaskResource.representations

`TaskResource.representations = None`

Methods

<code>TaskResource.__init__([api])</code>	Initialize self.
<code>TaskResource.as_view(name, *class_args, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>TaskResource.dispatch_request(*args, **kwargs)</code>	Subclasses have to override this method to implement the actual view function code.
<code>TaskResource.get_task(task_id)</code>	
<code>TaskResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.tasks.TaskResource.__init__

`TaskResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.tasks.TaskResource.as_view

classmethod `TaskResource.as_view(name, *class_args, **class_kwargs)`

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.tasks.TaskResource.dispatch_request

`TaskResource.dispatch_request(*args, **kwargs)`

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.tasks.TaskResource.get_task

static TaskResource.get_task(task_id: str) → celery.local.PromiseProxy

app.blueprints.tasks.TaskResource.validate_payload

TaskResource.validate_payload(func)
Perform a payload validation on expected model if necessary

app.blueprints.tasks.TaskStatusResource

class app.blueprints.tasks.TaskStatusResource(api=None, *args, **kwargs)
Bases: *app.blueprints.tasks.TaskResource*

Attributes

TaskStatusResource.decorators

TaskStatusResource.

method_decorators

TaskStatusResource.methods

TaskStatusResource.

provide_automatic_options

TaskStatusResource.

representations

app.blueprints.tasks.TaskStatusResource.decorators

TaskStatusResource.decorators = ()

app.blueprints.tasks.TaskStatusResource.method_decorators

TaskStatusResource.method_decorators = []

app.blueprints.tasks.TaskStatusResource.methods

```
TaskStatusResource.methods = {'GET'}
```

app.blueprints.tasks.TaskStatusResource.provide_automatic_options

```
TaskStatusResource.provide_automatic_options = None
```

app.blueprints.tasks.TaskStatusResource.representations

```
TaskStatusResource.representations = None
```

Methods

<code>TaskStatusResource.__init__([api])</code>	Initialize self.
<code>TaskStatusResource.as_view(name, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>TaskStatusResource.dispatch_request(*args, ...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>TaskStatusResource.get(task_id)</code>	
<code>TaskStatusResource.get_task(task_id)</code>	
<code>TaskStatusResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.tasks.TaskStatusResource.__init__

```
TaskStatusResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.tasks.TaskStatusResource.as_view

classmethod `TaskStatusResource.as_view(name, *class_args, **class_kwargs)`
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.tasks.TaskStatusResource.dispatch_request

`TaskStatusResource.dispatch_request(*args, **kwargs)`

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.tasks.TaskStatusResource.get

`TaskStatusResource.get(task_id: str)`

app.blueprints.tasks.TaskStatusResource.get_task

static `TaskStatusResource.get_task(task_id: str) → celery.local.PromiseProxy`

app.blueprints.tasks.TaskStatusResource.validate_payload

`TaskStatusResource.validate_payload(func)`

Perform a payload validation on expected model if necessary

class `app.blueprints.tasks.TaskResource(api=None, *args, **kwargs)`

classmethod `as_view(name, *class_args, **class_kwargs)`

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

decorators = ()

dispatch_request (*args, **kwargs)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

static `get_task(task_id: str) → celery.local.PromiseProxy`

method_decorators = []

methods = None

provide_automatic_options = None

representations = None

validate_payload (func)

Perform a payload validation on expected model if necessary

class `app.blueprints.tasks.TaskStatusResource(api=None, *args, **kwargs)`

classmethod `as_view(name, *class_args, **class_kwargs)`

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

```

decorators = ()

dispatch_request (*args, **kwargs)
    Subclasses have to override this method to implement the actual view function code. This method is called
    with all the arguments from the URL rule.

get (task_id: str)

static get_task (task_id: str) → celery.local.PromiseProxy

method_decorators = []

methods = {'GET'}

provide_automatic_options = None

representations = None

validate_payload (func)
    Perform a payload validation on expected model if necessary

```

app.blueprints.users

Description

Classes

ExportUsersExcelAndWordResource([api])

ExportUsersExcelResource([api])

ExportUsersWordResource([api])

NewUserResource([api])

UserBaseResource([api])

UserResource([api])

UsersSearchResource([api])

app.blueprints.users.ExportUsersExcelAndWordResource

```

class app.blueprints.users.ExportUsersExcelAndWordResource (api=None, *args,
                                                             **kwargs)
    Bases: app.blueprints.users.UserBaseResource

```

Attributes

ExportUsersExcelAndWordResource.
decorators

ExportUsersExcelAndWordResource.
method_decorators

ExportUsersExcelAndWordResource.
methods

ExportUsersExcelAndWordResource.
provide_automatic_options

ExportUsersExcelAndWordResource.
representations

continues on next page

Table 42 – continued from previous page

ExportUsersExcelAndWordResource.
user_serializer

app.blueprints.users.ExportUsersExcelAndWordResource.decorators

`ExportUsersExcelAndWordResource.decorators = ()`

app.blueprints.users.ExportUsersExcelAndWordResource.method_decorators

`ExportUsersExcelAndWordResource.method_decorators = []`

app.blueprints.users.ExportUsersExcelAndWordResource.methods

`ExportUsersExcelAndWordResource.methods = {'POST'}`

app.blueprints.users.ExportUsersExcelAndWordResource.provide_automatic_options

`ExportUsersExcelAndWordResource.provide_automatic_options = None`

app.blueprints.users.ExportUsersExcelAndWordResource.representations

`ExportUsersExcelAndWordResource.representations = None`

app.blueprints.users.ExportUsersExcelAndWordResource.user_serializer

`ExportUsersExcelAndWordResource.user_serializer = <UserSchema (many=False)>`

Methods

<i>ExportUsersExcelAndWordResource.</i> <i>__init__([api])</i>	Initialize self.
<i>ExportUsersExcelAndWordResource.</i> <i>as_view(...)</i>	Converts the class into an actual view function that can be used with the routing system.
<i>ExportUsersExcelAndWordResource.</i> <i>create_search_query(...)</i>	
<i>ExportUsersExcelAndWordResource.</i> <i>deserialize_request_data(...)</i>	
<i>ExportUsersExcelAndWordResource.</i> <i>dispatch_request(...)</i>	Subclasses have to override this method to implement the actual view function code.
<i>ExportUsersExcelAndWordResource.</i> <i>get_request_query_fields(...)</i>	
<i>ExportUsersExcelAndWordResource.</i> <i>post()</i>	

continues on next page

Table 43 – continued from previous page

<code>ExportUsersExcelAndWordResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary
---	---

app.blueprints.users.ExportUsersExcelAndWordResource.__init__

`ExportUsersExcelAndWordResource.__init__(api=None, *args, **kwargs)`
 Initialize self. See help(type(self)) for accurate signature.

app.blueprints.users.ExportUsersExcelAndWordResource.as_view

classmethod `ExportUsersExcelAndWordResource.as_view(name, *class_args, **class_kwargs)`
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
 The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.ExportUsersExcelAndWordResource.create_search_query

`ExportUsersExcelAndWordResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.users.ExportUsersExcelAndWordResource.deserialize_request_data

`ExportUsersExcelAndWordResource.deserialize_request_data(**kwargs: dict) → dict`

app.blueprints.users.ExportUsersExcelAndWordResource.dispatch_request

`ExportUsersExcelAndWordResource.dispatch_request(*args, **kwargs)`
 Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.ExportUsersExcelAndWordResource.get_request_query_fields

`ExportUsersExcelAndWordResource.get_request_query_fields` (*request_data*:
dict) →
tuple

app.blueprints.users.ExportUsersExcelAndWordResource.post

`ExportUsersExcelAndWordResource.post` () → *tuple*

app.blueprints.users.ExportUsersExcelAndWordResource.validate_payload

`ExportUsersExcelAndWordResource.validate_payload` (*func*)
Perform a payload validation on expected model if necessary

app.blueprints.users.ExportUsersExcelResource

class `app.blueprints.users.ExportUsersExcelResource` (*api=None, *args, **kwargs*)
Bases: `app.blueprints.users.UserBaseResource`

Attributes

`ExportUsersExcelResource.
decorators`

`ExportUsersExcelResource.
method_decorators`

`ExportUsersExcelResource.
methods`

`ExportUsersExcelResource.
provide_automatic_options`

`ExportUsersExcelResource.
representations`

`ExportUsersExcelResource.
user_serializer`

app.blueprints.users.ExportUsersExcelResource.decorators

`ExportUsersExcelResource.decorators` = ()

app.blueprints.users.ExportUsersExcelResource.method_decorators

```
ExportUsersExcelResource.method_decorators = []
```

app.blueprints.users.ExportUsersExcelResource.methods

```
ExportUsersExcelResource.methods = {'POST'}
```

app.blueprints.users.ExportUsersExcelResource.provide_automatic_options

```
ExportUsersExcelResource.provide_automatic_options = None
```

app.blueprints.users.ExportUsersExcelResource.representations

```
ExportUsersExcelResource.representations = None
```

app.blueprints.users.ExportUsersExcelResource.user_serializer

```
ExportUsersExcelResource.user_serializer = <UserSchema(many=False)>
```

Methods

<code>ExportUsersExcelResource.__init__(api)</code>	Initialize self.
<code>ExportUsersExcelResource.as_view(name, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>ExportUsersExcelResource.create_search_query(...)</code>	
<code>ExportUsersExcelResource.deserialize_request_data(...)</code>	
<code>ExportUsersExcelResource.dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>ExportUsersExcelResource.get_request_query_fields(...)</code>	
<code>ExportUsersExcelResource.post()</code>	
<code>ExportUsersExcelResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.users.ExportUsersExcelResource.__init__

`ExportUsersExcelResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.users.ExportUsersExcelResource.as_view

classmethod `ExportUsersExcelResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.
The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.ExportUsersExcelResource.create_search_query

`ExportUsersExcelResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.users.ExportUsersExcelResource.deserialize_request_data

`ExportUsersExcelResource.deserialize_request_data(**kwargs: dict) → dict`

app.blueprints.users.ExportUsersExcelResource.dispatch_request

`ExportUsersExcelResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.ExportUsersExcelResource.get_request_query_fields

`ExportUsersExcelResource.get_request_query_fields(request_data: dict) → tuple`

app.blueprints.users.ExportUsersExcelResource.post

`ExportUsersExcelResource.post() → tuple`

app.blueprints.users.ExportUsersExcelResource.validate_payload

`ExportUsersExcelResource.validate_payload(func)`

Perform a payload validation on expected model if necessary

app.blueprints.users.ExportUsersWordResource

class `app.blueprints.users.ExportUsersWordResource` (*api=None, *args, **kwargs*)

Bases: `app.blueprints.users.UserBaseResource`

Attributes

`ExportUsersWordResource.decorators`

`ExportUsersWordResource.method_decorators`

`ExportUsersWordResource.methods`

`ExportUsersWordResource.provide_automatic_options`

`ExportUsersWordResource.representations`

`ExportUsersWordResource.user_serializer`

app.blueprints.users.ExportUsersWordResource.decorators

`ExportUsersWordResource.decorators = ()`

app.blueprints.users.ExportUsersWordResource.method_decorators

`ExportUsersWordResource.method_decorators = []`

app.blueprints.users.ExportUsersWordResource.methods

`ExportUsersWordResource.methods = {'POST'}`

app.blueprints.users.ExportUsersWordResource.provide_automatic_options

`ExportUsersWordResource.provide_automatic_options = None`

app.blueprints.users.ExportUsersWordResource.representations

```
ExportUsersWordResource.representations = None
```

app.blueprints.users.ExportUsersWordResource.user_serializer

```
ExportUsersWordResource.user_serializer = <UserSchema(many=False)>
```

Methods

<code>ExportUsersWordResource. __init__([api])</code>	Initialize self.
<code>ExportUsersWordResource. as_view(name,...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>ExportUsersWordResource. create_search_query(...)</code>	
<code>ExportUsersWordResource. deserialize_request_data(...)</code>	
<code>ExportUsersWordResource. dispatch_request(...)</code>	Subclasses have to override this method to implement the actual view function code.
<code>ExportUsersWordResource. get_request_query_fields(...)</code>	
<code>ExportUsersWordResource.post()</code>	
<code>ExportUsersWordResource. validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.users.ExportUsersWordResource.__init__

```
ExportUsersWordResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.users.ExportUsersWordResource.as_view

```
classmethod ExportUsersWordResource.as_view(name, *class_args,
                                              **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.ExportUsersWordResource.create_search_query

`ExportUsersWordResource.create_search_query` (*query: peewee.ModelSelect, request_data: dict*) → *peewee.ModelSelect*

app.blueprints.users.ExportUsersWordResource.deserialize_request_data

`ExportUsersWordResource.deserialize_request_data` (***kwargs: dict*) → *dict*

app.blueprints.users.ExportUsersWordResource.dispatch_request

`ExportUsersWordResource.dispatch_request` (**args, **kwargs*)
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.ExportUsersWordResource.get_request_query_fields

`ExportUsersWordResource.get_request_query_fields` (*request_data: dict*) → *tuple*

app.blueprints.users.ExportUsersWordResource.post

`ExportUsersWordResource.post` () → *tuple*

app.blueprints.users.ExportUsersWordResource.validate_payload

`ExportUsersWordResource.validate_payload` (*func*)
Perform a payload validation on expected model if necessary

app.blueprints.users.NewUserResource

class `app.blueprints.users.NewUserResource` (*api=None, *args, **kwargs*)

Bases: `app.blueprints.users.UserBaseResource`

Attributes

NewUserResource.decorators

NewUserResource.method_decorators

NewUserResource.methods

NewUserResource.provide_automatic_options

NewUserResource.representations

NewUserResource.user_serializer

app.blueprints.users.NewUserResource.decorators

```
NewUserResource.decorators = ()
```

app.blueprints.users.NewUserResource.method_decorators

```
NewUserResource.method_decorators = []
```

app.blueprints.users.NewUserResource.methods

```
NewUserResource.methods = {'POST'}
```

app.blueprints.users.NewUserResource.provide_automatic_options

```
NewUserResource.provide_automatic_options = None
```

app.blueprints.users.NewUserResource.representations

```
NewUserResource.representations = None
```

app.blueprints.users.NewUserResource.user_serializer

```
NewUserResource.user_serializer = <UserSchema(many=False)>
```

Methods

<code>NewUserResource.__init__([api])</code>	Initialize self.
<code>NewUserResource.as_view(name, *class_args, ...)</code>	Converts the class into an actual view function that can be used with the routing system.
<code>NewUserResource.create_search_query(query, ...)</code>	
<code>NewUserResource.deserialize_request_data(...)</code>	
<code>NewUserResource.dispatch_request(*args, **kwargs)</code>	Subclasses have to override this method to implement the actual view function code.
<code>NewUserResource.get_request_query_fields(...)</code>	
<code>NewUserResource.post()</code>	
<code>NewUserResource.validate_payload(func)</code>	Perform a payload validation on expected model if necessary

app.blueprints.users.NewUserResource.__init__

`NewUserResource.__init__ (api=None, *args, **kwargs)`
 Initialize self. See help(type(self)) for accurate signature.

app.blueprints.users.NewUserResource.as_view

classmethod `NewUserResource.as_view (name, *class_args, **class_kwargs)`
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.NewUserResource.create_search_query

`NewUserResource.create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.users.NewUserResource.deserialize_request_data

`NewUserResource.deserialize_request_data (**kwargs: dict) → dict`

app.blueprints.users.NewUserResource.dispatch_request

`NewUserResource.dispatch_request (*args, **kwargs)`
 Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.NewUserResource.get_request_query_fields

`NewUserResource.get_request_query_fields (request_data: dict) → tuple`

app.blueprints.users.NewUserResource.post

`NewUserResource.post () → tuple`

app.blueprints.users.NewUserResource.validate_payload

`NewUserResource.validate_payload (func)`
 Perform a payload validation on expected model if necessary

app.blueprints.users.UserBaseResource

```
class app.blueprints.users.UserBaseResource (api=None, *args, **kwargs)
    Bases: app.blueprints.base.BaseResource
```

Attributes

UserBaseResource.decorators

UserBaseResource.
method_decorators

UserBaseResource.methods

UserBaseResource.
provide_automatic_options

UserBaseResource.
representations

UserBaseResource.
user_serializer

app.blueprints.users.UserBaseResource.decorators

```
UserBaseResource.decorators = ()
```

app.blueprints.users.UserBaseResource.method_decorators

```
UserBaseResource.method_decorators = []
```

app.blueprints.users.UserBaseResource.methods

```
UserBaseResource.methods = None
```

app.blueprints.users.UserBaseResource.provide_automatic_options

```
UserBaseResource.provide_automatic_options = None
```

app.blueprints.users.UserBaseResource.representations

```
UserBaseResource.representations = None
```


app.blueprints.users.UserBaseResource.user_serializer

```
UserBaseResource.user_serializer = <UserSchema(many=False)>
```

Methods

<i>UserBaseResource.__init__</i> ([api])	Initialize self.
<i>UserBaseResource.as_view</i> (name, *class_args, ...)	Converts the class into an actual view function that can be used with the routing system.
<i>UserBaseResource.create_search_query</i> (query, ...)	
<i>UserBaseResource.deserialize_request_data</i> (...)	
<i>UserBaseResource.dispatch_request</i> (*args, ...)	Subclasses have to override this method to implement the actual view function code.
<i>UserBaseResource.get_request_query_fields</i> (...)	
<i>UserBaseResource.validate_payload</i> (func)	Perform a payload validation on expected model if necessary

app.blueprints.users.UserBaseResource.__init__

```
UserBaseResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.users.UserBaseResource.as_view

classmethod *UserBaseResource.as_view*(name, *class_args, **class_kwargs)
 Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the *View* on each request and call the *dispatch_request()* method on it.

The arguments passed to *as_view()* are forwarded to the constructor of the class.

app.blueprints.users.UserBaseResource.create_search_query

`UserBaseResource.create_search_query` (*query: peewee.ModelSelect,*
request_data: dict) → *peewee.ModelSelect*

app.blueprints.users.UserBaseResource.deserialize_request_data

`UserBaseResource.deserialize_request_data` (***kwargs: dict*) → *dict*

app.blueprints.users.UserBaseResource.dispatch_request

`UserBaseResource.dispatch_request` (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.UserBaseResource.get_request_query_fields

`UserBaseResource.get_request_query_fields` (*request_data: dict*) → *tuple*

app.blueprints.users.UserBaseResource.validate_payload

`UserBaseResource.validate_payload` (*func*)

Perform a payload validation on expected model if necessary

app.blueprints.users.UserResource

class `app.blueprints.users.UserResource` (*api=None, *args, **kwargs*)

Bases: `app.blueprints.users.UserBaseResource`

Attributes

`UserResource.decorators`

`UserResource.method_decorators`

`UserResource.methods`

`UserResource.`

`provide_automatic_options`

`UserResource.representations`

`UserResource.user_serializer`

app.blueprints.users.UserResource.decorators

```
UserResource.decorators = ()
```

app.blueprints.users.UserResource.method_decorators

```
UserResource.method_decorators = []
```

app.blueprints.users.UserResource.methods

```
UserResource.methods = {'DELETE', 'GET', 'PUT'}
```

app.blueprints.users.UserResource.provide_automatic_options

```
UserResource.provide_automatic_options = None
```

app.blueprints.users.UserResource.representations

```
UserResource.representations = None
```

app.blueprints.users.UserResource.user_serializer

```
UserResource.user_serializer = <UserSchema(many=False)>
```

Methods

<i>UserResource.__init__</i> ([api])	Initialize self.
<i>UserResource.as_view</i> (name, *class_args, ...)	Converts the class into an actual view function that can be used with the routing system.
<i>UserResource.create_search_query</i> (query, ...)	
<i>UserResource.delete</i> (user_id)	
<i>UserResource.deserialize_request_data</i> (**kwargs)	
<i>UserResource.dispatch_request</i> (*args, **kwargs)	Subclasses have to override this method to implement the actual view function code.
<i>UserResource.get</i> (user_id)	
<i>UserResource.get_request_query_fields</i> (...)	
<i>UserResource.put</i> (user_id)	
<i>UserResource.validate_payload</i> (func)	Perform a payload validation on expected model if necessary

app.blueprints.users.UserResource.__init__

`UserResource.__init__(api=None, *args, **kwargs)`
Initialize self. See `help(type(self))` for accurate signature.

app.blueprints.users.UserResource.as_view

classmethod `UserResource.as_view(name, *class_args, **class_kwargs)`
Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.UserResource.create_search_query

`UserResource.create_search_query(query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect`

app.blueprints.users.UserResource.delete

`UserResource.delete(user_id: int) → tuple`

app.blueprints.users.UserResource.deserialize_request_data

`UserResource.deserialize_request_data(**kwargs: dict) → dict`

app.blueprints.users.UserResource.dispatch_request

`UserResource.dispatch_request(*args, **kwargs)`
Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.UserResource.get

`UserResource.get(user_id: int) → tuple`

app.blueprints.users.UserResource.get_request_query_fields

`UserResource.get_request_query_fields(request_data: dict) → tuple`

app.blueprints.users.UserResource.put

UserResource.**put** (*user_id: int*) → tuple

app.blueprints.users.UserResource.validate_payload

UserResource.**validate_payload** (*func*)

Perform a payload validation on expected model if necessary

app.blueprints.users.UsersSearchResource

class app.blueprints.users.**UsersSearchResource** (*api=None, *args, **kwargs*)

Bases: *app.blueprints.users.UserBaseResource*

Attributes

UsersSearchResource.decorators

*UsersSearchResource.
method_decorators*

UsersSearchResource.methods

*UsersSearchResource.
provide_automatic_options*

*UsersSearchResource.
representations*

*UsersSearchResource.
user_serializer*

app.blueprints.users.UsersSearchResource.decorators

UsersSearchResource.**decorators** = ()

app.blueprints.users.UsersSearchResource.method_decorators

UsersSearchResource.**method_decorators** = []

app.blueprints.users.UsersSearchResource.methods

UsersSearchResource.**methods** = {'POST'}

app.blueprints.users.UsersSearchResource.provide_automatic_options

```
UsersSearchResource.provide_automatic_options = None
```

app.blueprints.users.UsersSearchResource.representations

```
UsersSearchResource.representations = None
```

app.blueprints.users.UsersSearchResource.user_serializer

```
UsersSearchResource.user_serializer = <UserSchema(many=False)>
```

Methods

<i>UsersSearchResource.</i> <i>__init__</i> ([api])	Initialize self.
<i>UsersSearchResource.</i> <i>as_view</i> (name,...)	Converts the class into an actual view function that can be used with the routing system.
<i>UsersSearchResource.</i> <i>create_search_query</i> (...)	
<i>UsersSearchResource.</i> <i>deserialize_request_data</i> (...)	
<i>UsersSearchResource.</i> <i>dispatch_request</i> (*args,...)	Subclasses have to override this method to implement the actual view function code.
<i>UsersSearchResource.</i> <i>get_request_query_fields</i> (...)	
<i>UsersSearchResource.</i> <i>post</i> ()	
<i>UsersSearchResource.</i> <i>validate_payload</i> (func)	Perform a payload validation on expected model if necessary

app.blueprints.users.UsersSearchResource.__init__

```
UsersSearchResource.__init__(api=None, *args, **kwargs)
    Initialize self. See help(type(self)) for accurate signature.
```

app.blueprints.users.UsersSearchResource.as_view

```
classmethod UsersSearchResource.as_view(name, *class_args,
                                         **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

app.blueprints.users.UsersSearchResource.create_search_query

`UsersSearchResource.create_search_query` (*query: peewee.ModelSelect, request_data: dict*) → *peewee.ModelSelect*

app.blueprints.users.UsersSearchResource.deserialize_request_data

`UsersSearchResource.deserialize_request_data` (***kwargs: dict*) → *dict*

app.blueprints.users.UsersSearchResource.dispatch_request

`UsersSearchResource.dispatch_request` (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

app.blueprints.users.UsersSearchResource.get_request_query_fields

`UsersSearchResource.get_request_query_fields` (*request_data: dict*) → *tuple*

app.blueprints.users.UsersSearchResource.post

`UsersSearchResource.post` () → *tuple*

app.blueprints.users.UsersSearchResource.validate_payload

`UsersSearchResource.validate_payload` (*func*)

Perform a payload validation on expected model if necessary

```
class app.blueprints.users.ExportUsersExcelAndWordResource (api=None, *args, **kwargs)
```

```
classmethod as_view (name, *class_args, **class_kwargs)
```

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the View on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

```
create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect
```

```
db_model
```

alias of `app.models.user.User`

```
decorators = ()
```

```
deserialize_request_data (**kwargs: dict) → dict
```

```
dispatch_request (*args, **kwargs)
```

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

```
get_request_query_fields (request_data: dict) → tuple
method_decorators = []
methods = {'POST'}
post () → tuple
provide_automatic_options = None
representations = None
user_serializer = <UserSchema (many=False)>
validate_payload (func)
    Perform a payload validation on expected model if necessary
```

class app.blueprints.users.**ExportUsersExcelResource** (api=None, *args, **kwargs)

classmethod **as_view** (name, *class_args, **class_kwargs)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

db_model

alias of `app.models.user.User`

decorators = ()

deserialize_request_data (**kwargs: dict) → dict

dispatch_request (*args, **kwargs)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

get_request_query_fields (request_data: dict) → tuple

method_decorators = []

methods = {'POST'}

post () → tuple

provide_automatic_options = None

representations = None

user_serializer = <UserSchema (many=False)>

validate_payload (func)

Perform a payload validation on expected model if necessary

class app.blueprints.users.**ExportUsersWordResource** (api=None, *args, **kwargs)

classmethod **as_view** (name, *class_args, **class_kwargs)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.


```

create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

db_model
    alias of app.models.user.User

decorators = ()

deserialize_request_data (**kwargs: dict) → dict

dispatch_request (*args, **kwargs)
    Subclasses have to override this method to implement the actual view function code. This method is called
    with all the arguments from the URL rule.

get_request_query_fields (request_data: dict) → tuple

method_decorators = []

methods = {'POST'}

post () → tuple

provide_automatic_options = None

representations = None

user_serializer = <UserSchema (many=False)>

validate_payload (func)
    Perform a payload validation on expected model if necessary

class app.blueprints.users.NewUserResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect

    db_model
        alias of app.models.user.User

    decorators = ()

    deserialize_request_data (**kwargs: dict) → dict

    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    get_request_query_fields (request_data: dict) → tuple

    method_decorators = []

    methods = {'POST'}

    post () → tuple

    provide_automatic_options = None

    representations = None

    user_serializer = <UserSchema (many=False)>

```

validate_payload (*func*)

Perform a payload validation on expected model if necessary

class app.blueprints.users.**UserBaseResource** (*api=None, *args, **kwargs*)

classmethod **as_view** (*name, *class_args, **class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

create_search_query (*query: peewee.ModelSelect, request_data: dict*) → `peewee.ModelSelect`

db_model

alias of `app.models.user.User`

decorators = ()

deserialize_request_data (***kwargs: dict*) → `dict`

dispatch_request (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

get_request_query_fields (*request_data: dict*) → `tuple`

method_decorators = []

methods = `None`

provide_automatic_options = `None`

representations = `None`

user_serializer = `<UserSchema(many=False)>`

validate_payload (*func*)

Perform a payload validation on expected model if necessary

class app.blueprints.users.**UserResource** (*api=None, *args, **kwargs*)

classmethod **as_view** (*name, *class_args, **class_kwargs*)

Converts the class into an actual view function that can be used with the routing system. Internally this generates a function on the fly which will instantiate the `View` on each request and call the `dispatch_request()` method on it.

The arguments passed to `as_view()` are forwarded to the constructor of the class.

create_search_query (*query: peewee.ModelSelect, request_data: dict*) → `peewee.ModelSelect`

db_model

alias of `app.models.user.User`

decorators = ()

delete (*user_id: int*) → `tuple`

deserialize_request_data (***kwargs: dict*) → `dict`

dispatch_request (**args, **kwargs*)

Subclasses have to override this method to implement the actual view function code. This method is called with all the arguments from the URL rule.

```

get (user_id: int) → tuple
get_request_query_fields (request_data: dict) → tuple
method_decorators = []
methods = {'DELETE', 'GET', 'PUT'}
provide_automatic_options = None
put (user_id: int) → tuple
representations = None
user_serializer = <UserSchema (many=False)>
validate_payload (func)
    Perform a payload validation on expected model if necessary
class app.blueprints.users.UsersSearchResource (api=None, *args, **kwargs)

    classmethod as_view (name, *class_args, **class_kwargs)
        Converts the class into an actual view function that can be used with the routing system. Internally
        this generates a function on the fly which will instantiate the View on each request and call the
        dispatch_request() method on it.

        The arguments passed to as_view() are forwarded to the constructor of the class.

    create_search_query (query: peewee.ModelSelect, request_data: dict) → peewee.ModelSelect
    db_model
        alias of app.models.user.User
    decorators = ()
    deserialize_request_data (**kwargs: dict) → dict
    dispatch_request (*args, **kwargs)
        Subclasses have to override this method to implement the actual view function code. This method is called
        with all the arguments from the URL rule.

    get_request_query_fields (request_data: dict) → tuple
    method_decorators = []
    methods = {'POST'}
    post () → tuple
    provide_automatic_options = None
    representations = None
    user_serializer = <UserSchema (many=False)>
    validate_payload (func)
        Perform a payload validation on expected model if necessary

```

2.1.2 app.celery

Description

Registers Celery tasks.

Modules

`app.celery.excel`

`app.celery.tasks`

`app.celery.word`

app.celery.excel

Description

Modules

`app.celery.excel.tasks`

app.celery.excel.tasks

Description

`app.celery.excel.tasks._add_excel_autofilter` (*worksheet:* `xl-writer.worksheet.Worksheet`)

`app.celery.excel.tasks._adjust_each_column_width` (*rows:* `list`, *worksheet:* `xl-writer.worksheet.Worksheet`, *excel_longest_word:* `int`) → `None`

`app.celery.excel.tasks._get_excel_column_names` (*excel_rows:* `list`) → `None`

`app.celery.excel.tasks._get_excel_user_data` (*users:* `list`, *excel_rows:* `list`) → `None`

`app.celery.excel.tasks._get_user_data` (*request_data:* `dict`) → `list`

`app.celery.excel.tasks._parse_user_data` (*users:* `list`)

(task) `app.celery.excel.tasks.export_user_data_in_excel` (*created_by:* `int`, *request_data:* `dict`)

Export User Data in Excel task

Parameters

- **self** –
- **created_by** –
- **request_data** –

Returns

app.celery.tasks

Description

(task) `app.celery.tasks.create_user_email(email_data)` → bool

Proxy that evaluates object once.

Proxy will evaluate the object each time, while the promise will only evaluate it once.

(task) `app.celery.tasks.create_word_and_excel_documents(created_by: int, request_data: dict, to_pdf: int)` → bool

Proxy that evaluates object once.

Proxy will evaluate the object each time, while the promise will only evaluate it once.

(task) `app.celery.tasks.reset_password_email(email_data)` → bool

Proxy that evaluates object once.

Proxy will evaluate the object each time, while the promise will only evaluate it once.

(task) `app.celery.tasks.send_email_with_attachments(task_data: list)` → bool

Proxy that evaluates object once.

Proxy will evaluate the object each time, while the promise will only evaluate it once.

app.celery.word

Description

Modules

[`app.celery.word.tasks`](#)

app.celery.word.tasks

Description

`app.celery.word.tasks._add_table_column_names(rows: list, original_column_names: set)` → None

`app.celery.word.tasks._add_table_user_data(users_query: list, rows: list)` → None

`app.celery.word.tasks._get_user_data(request_data: dict)` → list

(task) `app.celery.word.tasks.export_user_data_in_word(created_by: int, request_data: dict, to_pdf: int)`

Export User Data in Word

Parameters

- **self** –
- **created_by** –
- **request_data** –
- **to_pdf** –

Returns

Classes

ContextTask()

MyCelery([main, loader, backend, amqp, ...])

app.celery.ContextTask

class app.celery.**ContextTask**
Bases: celery.app.task.Task

Attributes

ContextTask.Request

ContextTask.Strategy

ContextTask.abstract

ContextTask.acks_late

ContextTask.acks_on_failure_or_timeout

ContextTask.app

ContextTask.autoregister

ContextTask.backend

ContextTask.default_retry_delay

ContextTask.expires

ContextTask.from_config

ContextTask.ignore_result

ContextTask.max_retries

ContextTask.name

ContextTask.priority

ContextTask.rate_limit

ContextTask.reject_on_worker_lost

ContextTask.request Get current request object.

ContextTask.request_stack

ContextTask.resultrepr_maxsize

ContextTask.send_events

ContextTask.serializer

ContextTask.soft_time_limit

ContextTask.store_errors_even_if_ignored

ContextTask.throws

ContextTask.time_limit

ContextTask.track_started

ContextTask.trail

ContextTask.typing

app.celery.ContextTask.Request

```
ContextTask.Request = 'celery.worker.request:Request'
```

app.celery.ContextTask.Strategy

```
ContextTask.Strategy = 'celery.worker.strategy:default'
```

app.celery.ContextTask.abstract

```
ContextTask.abstract = True
```

app.celery.ContextTask.acks_late

```
ContextTask.acks_late = False
```

app.celery.ContextTask.acks_on_failure_or_timeout

```
ContextTask.acks_on_failure_or_timeout = True
```

app.celery.ContextTask.app

```
ContextTask.app = <MyCelery __main__>
```

app.celery.ContextTask.autoregister

```
ContextTask.autoregister = True
```

app.celery.ContextTask.backend

```
property ContextTask.backend
```

app.celery.ContextTask.default_retry_delay

```
ContextTask.default_retry_delay = 180
```

app.celery.ContextTask.expires

```
ContextTask.expires = None
```

app.celery.ContextTask.from_config

```
ContextTask.from_config = (('serializer', 'task_serializer'), ('rate_limit', 'task_
```

app.celery.ContextTask.ignore_result

```
ContextTask.ignore_result = False
```

app.celery.ContextTask.max_retries

```
ContextTask.max_retries = 3
```

app.celery.ContextTask.name

```
ContextTask.name = None
```

app.celery.ContextTask.priority

```
ContextTask.priority = None
```

app.celery.ContextTask.rate_limit

```
ContextTask.rate_limit = None
```

app.celery.ContextTask.reject_on_worker_lost

```
ContextTask.reject_on_worker_lost = None
```

app.celery.ContextTask.request

```
property ContextTask.request  
    Get current request object.
```


app.celery.ContextTask.request_stack

```
ContextTask.request_stack = <celery.utils.threads._LocalStack object>
```

app.celery.ContextTask.resultrepr_maxsize

```
ContextTask.resultrepr_maxsize = 1024
```

app.celery.ContextTask.send_events

```
ContextTask.send_events = True
```

app.celery.ContextTask.serializer

```
ContextTask.serializer = 'json'
```

app.celery.ContextTask.soft_time_limit

```
ContextTask.soft_time_limit = None
```

app.celery.ContextTask.store_errors_even_if_ignored

```
ContextTask.store_errors_even_if_ignored = False
```

app.celery.ContextTask.raises

```
ContextTask.raises = ()
```

app.celery.ContextTask.time_limit

```
ContextTask.time_limit = None
```

app.celery.ContextTask.track_started

```
ContextTask.track_started = False
```

app.celery.ContextTask.trail

```
ContextTask.trail = True
```

app.celery.ContextTask.typing

```
ContextTask.typing = True
```

Methods

<i>ContextTask.AsyncResult</i> (task_id, **kwargs)	Get AsyncResult instance for the specified task.
<i>ContextTask.__init__</i> ()	Initialize self.
<i>ContextTask.add_around</i> (attr, around)	
<i>ContextTask.add_to_chord</i> (sig[, lazy])	Add signature to the chord the current task is a member of.
<i>ContextTask.add_trail</i> (result)	
<i>ContextTask.after_return</i> (status, retval, ...)	Handler called after the task returns.
<i>ContextTask.annotate</i> ()	
<i>ContextTask.apply</i> ([args, kwargs, link, ...])	Execute this task locally, by blocking until the task returns.
<i>ContextTask.apply_async</i> ([args, kwargs, ...])	Apply tasks asynchronously by sending a message.
<i>ContextTask.bind</i> (app)	
<i>ContextTask.chunks</i> (it, n)	Create a chunks task for this task.
<i>ContextTask.delay</i> (*args, **kwargs)	Star argument version of <i>apply_async()</i> .
<i>ContextTask.map</i> (it)	Create a xmap task from it.
<i>ContextTask.on_bound</i> (app)	Called when the task is bound to an app.
<i>ContextTask.on_failure</i> (exc, task_id, args, ...)	param exc The exception raised by the task.
<i>ContextTask.on_retry</i> (exc, task_id, args, ...)	Retry handler.
<i>ContextTask.on_success</i> (retval, task_id, ...)	Success handler.
<i>ContextTask.pop_request</i> ()	
<i>ContextTask.push_request</i> (*args, **kwargs)	
<i>ContextTask.replace</i> (sig)	Replace this task, with a new task inheriting the task id.
<i>ContextTask.retry</i> ([args, kwargs, exc, ...])	Retry the task, adding it to the back of the queue.
<i>ContextTask.run</i> (*args, **kwargs)	The body of the task executed by workers.
<i>ContextTask.s</i> (*args, **kwargs)	Create signature.
<i>ContextTask.send_event</i> (type[, retry, ...])	Send monitoring event message.

continues on next page

Table 61 – continued from previous page

<code>ContextTask.shadow_name(args, kwargs, options)</code>	Override for custom task name in worker logs/monitoring.
<code>ContextTask.si(*args, **kwargs)</code>	Create immutable signature.
<code>ContextTask.signature([args])</code>	Create signature.
<code>ContextTask.signature_from_request([...])</code>	
<code>ContextTask.starmap(it)</code>	Create a xstarmap task from it.
<code>ContextTask.start_strategy(app, consumer, ...)</code>	
<code>ContextTask.subtask([args])</code>	Create signature.
<code>ContextTask.subtask_from_request([request, ...])</code>	
<code>ContextTask.update_state([task_id, state, meta])</code>	Update task state.

app.celery.ContextTask.AsyncResult

`ContextTask.AsyncResult(task_id, **kwargs)`

Get AsyncResult instance for the specified task.

Parameters `task_id` (*str*) – Task id to get result for.

app.celery.ContextTask.__init__

`ContextTask.__init__()`

Initialize self. See `help(type(self))` for accurate signature.

app.celery.ContextTask.add_around

classmethod `ContextTask.add_around(attr, around)`

app.celery.ContextTask.add_to_chord

`ContextTask.add_to_chord(sig, lazy=False)`

Add signature to the chord the current task is a member of.

New in version 4.0.

Currently only supported by the Redis result backend.

Parameters

- **sig** (*~@Signature*) – Signature to extend chord with.
- **lazy** (*bool*) – If enabled the new task won't actually be called, and `sig.delay()` must be called manually.

app.celery.ContextTask.add_trail

ContextTask.**add_trail** (*result*)

app.celery.ContextTask.after_return

ContextTask.**after_return** (*status, retval, task_id, args, kwargs, einfo*)

Handler called after the task returns.

Parameters

- **status** (*str*) – Current task state.
- **retval** (*Any*) – Task return value/exception.
- **task_id** (*str*) – Unique id of the task.
- **args** (*Tuple*) – Original arguments for the task.
- **kwargs** (*Dict*) – Original keyword arguments for the task.
- **einfo** (*ExceptionInfo*) – Exception information.

Returns The return value of this handler is ignored.

Return type None

app.celery.ContextTask.annotate

classmethod ContextTask.**annotate** ()

app.celery.ContextTask.apply

ContextTask.**apply** (*args=None, kwargs=None, link=None, link_error=None, task_id=None, retries=None, throw=None, logfile=None, loglevel=None, headers=None, **options*)

Execute this task locally, by blocking until the task returns.

Parameters

- **args** (*Tuple*) – positional arguments passed on to the task.
- **kwargs** (*Dict*) – keyword arguments passed on to the task.
- **throw** (*bool*) – Re-raise task exceptions. Defaults to the **:setting: `task_eager_propagates`** setting.

Returns pre-evaluated result.

Return type celery.result.EagerResult

app.celery.ContextTask.apply_async

ContextTask.**apply_async** (*args=None, kwargs=None, task_id=None, producer=None, link=None, link_error=None, shadow=None, **options*)

Apply tasks asynchronously by sending a message.

Parameters

- **args** (*Tuple*) – The positional arguments to pass on to the task.
- **kwargs** (*Dict*) – The keyword arguments to pass on to the task.
- **countdown** (*float*) – Number of seconds into the future that the task should execute. Defaults to immediate execution.
- **eta** (*datetime*) – Absolute time and date of when the task should be executed. May not be specified if *countdown* is also supplied.

- **expires** (*float, datetime*) – Datetime or seconds in the future for the task should expire. The task won't be executed after the expiration time.
- **shadow** (*str*) – Override task name used in logs/monitoring. Default is retrieved from `shadow_name()`.
- **connection** (*kombu.Connection*) – Re-use existing broker connection instead of acquiring one from the connection pool.
- **retry** (*bool*) – If enabled sending of the task message will be retried in the event of connection loss or failure. Default is taken from the `:setting:`task_publish_retry`` setting. Note that you need to handle the producer/connection manually for this to work.
- **retry_policy** (*Mapping*) – Override the retry policy used. See the `:setting:`task_publish_retry_policy`` setting.
- **queue** (*str, kombu.Queue*) – The queue to route the task to. This must be a key present in `:setting:`task_queues``, or `:setting:`task_create_missing_queues`` must be enabled. See guide-routing for more information.
- **exchange** (*str, kombu.Exchange*) – Named custom exchange to send the task to. Usually not used in combination with the `queue` argument.
- **routing_key** (*str*) – Custom routing key used to route the task to a worker server. If in combination with a `queue` argument only used to specify custom routing keys to topic exchanges.
- **priority** (*int*) – The task priority, a number between 0 and 9. Defaults to the `priority` attribute.
- **serializer** (*str*) – Serialization method to use. Can be `pickle`, `json`, `yaml`, `msgpack` or any custom serialization method that's been registered with `kombu.serialization.registry`. Defaults to the `serializer` attribute.
- **compression** (*str*) – Optional compression method to use. Can be one of `zlib`, `bzip2`, or any custom compression methods registered with `kombu.compression.register()`. Defaults to the `:setting:`task_compression`` setting.
- **link** (*Signature*) – A single, or a list of tasks signatures to apply if the task returns successfully.
- **link_error** (*Signature*) – A single, or a list of task signatures to apply if an error occurs while executing the task.
- **producer** (*kombu.Producer*) – custom producer to use when publishing the task.
- **add_to_parent** (*bool*) – If set to `True` (default) and the task is applied while executing another task, then the result will be appended to the parent tasks `request.children` attribute. Trailing can also be disabled by default using the `trail` attribute
- **publisher** (*kombu.Producer*) – Deprecated alias to `producer`.
- **headers** (*Dict*) – Message headers to be included in the message.

Returns Promise of future evaluation.

Return type `celery.result.AsyncResult`

Raises

- **TypeError** – If not enough arguments are passed, or too many arguments are passed. Note that signature checks may be disabled by specifying `@task(typing=False)`.
- **kombu.exceptions.OperationalError** – If a connection to the transport cannot be made, or if the connection is lost.

Note: Also supports all keyword arguments supported by `kombu.Producer.publish()`.

app.celery.ContextTask.bind

classmethod ContextTask.**bind**(app)

app.celery.ContextTask.chunks

ContextTask.**chunks**(it, n)
Create a chunks task for this task.

app.celery.ContextTask.delay

ContextTask.**delay**(*args, **kwargs)
Star argument version of [apply_async\(\)](#).
Does not support the extra options enabled by [apply_async\(\)](#).
Parameters

- ***args** (Any) – Positional arguments passed on to the task.
- ****kwargs** (Any) – Keyword arguments passed on to the task.

Returns Future promise.
Return type celery.result.AsyncResult

app.celery.ContextTask.map

ContextTask.**map**(it)
Create a xmap task from it.

app.celery.ContextTask.on_bound

classmethod ContextTask.**on_bound**(app)
Called when the task is bound to an app.

Note: This class method can be defined to do additional actions when the task class is bound to an app.

app.celery.ContextTask.on_failure

ContextTask.**on_failure**(exc, task_id, args, kwargs, einfo) → None
Parameters

- **exc** – The exception raised by the task.
- **task_id** – Unique id of the failed task.
- **args** – Original arguments for the task that failed.
- **kwargs** – Original keyword arguments for the task that failed.
- **einfo** – ExceptionInfo instance, containing the traceback

Returns None

app.celery.ContextTask.on_retry

`ContextTask.on_retry(exc, task_id, args, kwargs, einfo)`

Retry handler.

This is run by the worker when the task is to be retried.

Parameters

- **exc** (*Exception*) – The exception sent to `retry()`.
- **task_id** (*str*) – Unique id of the retried task.
- **args** (*Tuple*) – Original arguments for the retried task.
- **kwargs** (*Dict*) – Original keyword arguments for the retried task.
- **einfo** (*ExceptionInfo*) – Exception information.

Returns The return value of this handler is ignored.

Return type None

app.celery.ContextTask.on_success

`ContextTask.on_success(retval, task_id, args, kwargs)`

Success handler.

Run by the worker if the task executes successfully.

Parameters

- **retval** (*Any*) – The return value of the task.
- **task_id** (*str*) – Unique id of the executed task.
- **args** (*Tuple*) – Original arguments for the executed task.
- **kwargs** (*Dict*) – Original keyword arguments for the executed task.

Returns The return value of this handler is ignored.

Return type None

app.celery.ContextTask.pop_request

`ContextTask.pop_request()`

app.celery.ContextTask.push_request

`ContextTask.push_request(*args, **kwargs)`

app.celery.ContextTask.replace

`ContextTask.replace(sig)`

Replace this task, with a new task inheriting the task id.

Execution of the host task ends immediately and no subsequent statements will be run.

New in version 4.0.

Parameters **sig** (*~@Signature*) – signature to replace with.

Raises

- **~@Ignore** – This is always raised when called in asynchronous context.
- **It is best to always use `return self.replace(..)` to convey –**
- **to the reader that the task won't continue after being replaced. –**

app.celery.ContextTask.retry

ContextTask.**retry**(args=None, kwargs=None, exc=None, throw=True, eta=None, countdown=None, max_retries=None, **options)

Retry the task, adding it to the back of the queue.

Example

```
>>> from imaginary_twitter_lib import Twitter
>>> from proj.celery import app

>>> @app.task(bind=True)
... def tweet(self, auth, message):
...     twitter = Twitter(oauth=auth)
...     try:
...         twitter.post_status_update(message)
...     except twitter.FailWhale as exc:
...         # Retry in 5 minutes.
...         self.retry(countdown=60 * 5, exc=exc)
```

Note: Although the task will never return above as *retry* raises an exception to notify the worker, we use *raise* in front of the *retry* to convey that the rest of the block won't be executed.

Parameters

- **args** (*Tuple*) – Positional arguments to retry with.
- **kwargs** (*Dict*) – Keyword arguments to retry with.
- **exc** (*Exception*) – Custom exception to report when the max retry limit has been exceeded (default: @MaxRetriesExceededError).

If this argument is set and *retry* is called while an exception was raised (`sys.exc_info()` is set) it will attempt to re-raise the current exception.

If no exception was raised it will raise the `exc` argument provided.

- **countdown** (*float*) – Time in seconds to delay the retry for.
- **eta** (*datetime*) – Explicit time and date to run the retry at.
- **max_retries** (*int*) – If set, overrides the default retry limit for this execution. Changes to this parameter don't propagate to subsequent task retry attempts. A value of `None`, means “use the default”, so if you want infinite retries you'd have to set the `max_retries` attribute of the task to `None` first.
- **time_limit** (*int*) – If set, overrides the default time limit.
- **soft_time_limit** (*int*) – If set, overrides the default soft time limit.
- **throw** (*bool*) – If this is `False`, don't raise the @Retry exception, that tells the worker to mark the task as being retried. Note that this means the task will be marked as failed if the task raises an exception, or successful if it returns after the retry call.
- ****options** (*Any*) – Extra options to pass on to `apply_async()`.

Raises `celery.exceptions.Retry` – To tell the worker that the task has been re-sent for retry. This always happens, unless the *throw* keyword argument has been explicitly set to `False`, and is considered normal operation.

app.celery.ContextTask.run

`ContextTask.run(*args, **kwargs)`

The body of the task executed by workers.

app.celery.ContextTask.s

`ContextTask.s(*args, **kwargs)`

Create signature.

Shortcut for `.s(*a, **k) -> .signature(a, k)`.

app.celery.ContextTask.send_event

`ContextTask.send_event(type_, retry=True, retry_policy=None, **fields)`

Send monitoring event message.

This can be used to add custom event types in `:pypi:Flower`` and other monitors.

Parameters `type` (*str*) – Type of event, e.g. "task-failed".

Keyword Arguments

- **retry** (*bool*) – Retry sending the message if the connection is lost. Default is taken from the `:setting:task_publish_retry`` setting.
- **retry_policy** (*Mapping*) – Retry settings. Default is taken from the `:setting:task_publish_retry_policy`` setting.
- ****fields** (*Any*) – Map containing information about the event. Must be JSON serializable.

app.celery.ContextTask.shadow_name

`ContextTask.shadow_name(args, kwargs, options)`

Override for custom task name in worker logs/monitoring.

Example

```
from celery.utils.imports import qualname

def shadow_name(task, args, kwargs, options):
    return qualname(args[0])

@app.task(shadow_name=shadow_name, serializer='pickle')
def apply_function_async(fun, *args, **kwargs):
    return fun(*args, **kwargs)
```

Parameters

- **args** (*Tuple*) – Task positional arguments.
- **kwargs** (*Dict*) – Task keyword arguments.
- **options** (*Dict*) – Task execution options.

app.celery.ContextTask.si

ContextTask.**si** (*args, **kwargs)

Create immutable signature.

Shortcut for `.si(*a, **k) -> .signature(a, k, immutable=True)`.

app.celery.ContextTask.signature

ContextTask.**signature** (args=None, *starargs, **starkwargs)

Create signature.

Returns

object for this task, wrapping arguments and execution options for a single task invocation.

Return type signature

app.celery.ContextTask.signature_from_request

ContextTask.**signature_from_request** (request=None, args=None, kwargs=None, queue=None, **extra_options)

app.celery.ContextTask.starmap

ContextTask.**starmap** (it)

Create a xstarmap task from it.

app.celery.ContextTask.start_strategy

ContextTask.**start_strategy** (app, consumer, **kwargs)

app.celery.ContextTask.subtask

ContextTask.**subtask** (args=None, *starargs, **starkwargs)

Create signature.

Returns

object for this task, wrapping arguments and execution options for a single task invocation.

Return type signature

app.celery.ContextTask.subtask_from_request

`ContextTask.subtask_from_request` (*request=None, args=None, kwargs=None, queue=None, **extra_options*)

app.celery.ContextTask.update_state

`ContextTask.update_state` (*task_id=None, state=None, meta=None, **kwargs*)

Update task state.

Parameters

- **task_id** (*str*) – Id of the task to update. Defaults to the id of the current task.
- **state** (*str*) – New state.
- **meta** (*Dict*) – State meta-data.

app.celery.MyCelery

class `app.celery.MyCelery` (*main=None, loader=None, backend=None, amqp=None, events=None, log=None, control=None, set_as_current=True, tasks=None, broker=None, include=None, changes=None, config_source=None, fixups=None, task_cls=None, autofinalize=True, namespace=None, strict_typing=True, **kwargs*)

Bases: `celery.app.base.Celery`

Attributes

<code>MyCelery.AsyncResult</code>	Create new result instance.
<code>MyCelery.Beat</code>	celery beat scheduler application.
<code>MyCelery.GroupResult</code>	Create new group result instance.
<code>MyCelery.IS_WINDOWS</code>	
<code>MyCelery.IS_macOS</code>	
<code>MyCelery.ResultSet</code>	
<code>MyCelery.SYSTEM</code>	
<code>MyCelery.Task</code>	Base task class for this app.
<code>MyCelery.WorkController</code>	Embeddable worker.
<code>MyCelery.Worker</code>	Worker application.
<code>MyCelery.amqp</code>	@amqp.
<code>MyCelery.amqp_cls</code>	
<code>MyCelery.annotations</code>	
<code>MyCelery.backend</code>	Current backend instance.
<code>MyCelery.backend_cls</code>	
<code>MyCelery.builtin_fixups</code>	
<code>MyCelery.conf</code>	Current configuration.
<code>MyCelery.control</code>	@control.
<code>MyCelery.control_cls</code>	
<code>MyCelery.current_task</code>	Instance of task being executed, or None.
<code>MyCelery.current_worker_task</code>	The task currently being executed by a worker or None.
<code>MyCelery.events</code>	@events.
<code>MyCelery.events_cls</code>	

continues on next page

Table 62 – continued from previous page

<i>MyCelery.loader</i>	Current loader instance.
<i>MyCelery.loader_cls</i>	
<i>MyCelery.log</i>	@log.
<i>MyCelery.log_cls</i>	
<i>MyCelery.main</i>	
<i>MyCelery.oid</i>	Universally unique identifier for this app.
<i>MyCelery.on_after_configure</i>	
<i>MyCelery.on_after_finalize</i>	
<i>MyCelery.on_after_fork</i>	
<i>MyCelery.on_configure</i>	
<i>MyCelery.pool</i>	@pool.
<i>MyCelery.producer_pool</i>	
<i>MyCelery.registry_cls</i>	
<i>MyCelery.steps</i>	
<i>MyCelery.task_cls</i>	
<i>MyCelery.tasks</i>	Task registry.
<i>MyCelery.timezone</i>	Current timezone for this app.
<i>MyCelery.user_options</i>	

app.celery.MyCelery.AsyncResult

MyCelery.AsyncResult

Create new result instance.

See also:

`celery.result.AsyncResult`.

app.celery.MyCelery.Beat

MyCelery.Beat

celery beat scheduler application.

See also:

@Beat.

app.celery.MyCelery.GroupResult

MyCelery.GroupResult

Create new group result instance.

See also:

`celery.result.GroupResult`.

app.celery.MyCelery.IS_WINDOWS

`MyCelery.IS_WINDOWS = False`

app.celery.MyCelery.IS_macOS

`MyCelery.IS_macOS = False`

app.celery.MyCelery.ResultSet

`MyCelery.ResultSet`

app.celery.MyCelery.SYSTEM

`MyCelery.SYSTEM = 'Linux'`

app.celery.MyCelery.Task

`MyCelery.Task`

Base task class for this app.

app.celery.MyCelery.WorkController

`MyCelery.WorkController`

Embeddable worker.

See also:

`@WorkController.`

app.celery.MyCelery.Worker

`MyCelery.Worker`

Worker application.

See also:

`@Worker.`

app.celery.MyCelery.amqp

`MyCelery.amqp`

`@amqp.`

Type AMQP related functionality

app.celery.MyCelery.amqp_cls

```
MyCelery.amqp_cls = 'celery.app.amqp:AMQP'
```

app.celery.MyCelery.annotations

```
MyCelery.annotations
```

app.celery.MyCelery.backend

```
MyCelery.backend  
    Current backend instance.
```

app.celery.MyCelery.backend_cls

```
MyCelery.backend_cls = None
```

app.celery.MyCelery.builtin_fixups

```
MyCelery.builtin_fixups = {'celery.fixups.django:fixup'}
```

app.celery.MyCelery.conf

```
property MyCelery.conf  
    Current configuration.
```

app.celery.MyCelery.control

```
MyCelery.control  
    @control.  
    Type Remote control
```

app.celery.MyCelery.control_cls

```
MyCelery.control_cls = 'celery.app.control:Control'
```

app.celery.MyCelery.current_task

```
property MyCelery.current_task  
    Instance of task being executed, or None.
```

app.celery.MyCelery.current_worker_task

property MyCelery.current_worker_task

The task currently being executed by a worker or None.

Differs from `current_task` in that it's not affected by tasks calling other tasks directly, or eagerly.

app.celery.MyCelery.events

MyCelery.events

@events.

Type Consuming and sending events

app.celery.MyCelery.events_cls

MyCelery.events_cls = 'celery.app.events:Events'

app.celery.MyCelery.loader

MyCelery.loader

Current loader instance.

app.celery.MyCelery.loader_cls

MyCelery.loader_cls = None

app.celery.MyCelery.log

MyCelery.log

@log.

Type Logging

app.celery.MyCelery.log_cls

MyCelery.log_cls = 'celery.app.log:Logging'

app.celery.MyCelery.main

MyCelery.main = None

app.celery.MyCelery.oid**MyCelery.oid**

Universally unique identifier for this app.

app.celery.MyCelery.on_after_configure**MyCelery.on_after_configure = None****app.celery.MyCelery.on_after_finalize****MyCelery.on_after_finalize = None****app.celery.MyCelery.on_after_fork****MyCelery.on_after_fork = None****app.celery.MyCelery.on_configure****MyCelery.on_configure = None****app.celery.MyCelery.pool****property** **MyCelery.pool**

@pool.

Note: This attribute is not related to the workers concurrency pool.

Type Broker connection pool**app.celery.MyCelery.producer_pool****property** **MyCelery.producer_pool****app.celery.MyCelery.registry_cls****MyCelery.registry_cls = 'celery.app.registry:TaskRegistry'**

app.celery.MyCelery.steps

MyCelery.steps = None

app.celery.MyCelery.task_cls

MyCelery.task_cls = 'celery.app.task:Task'

app.celery.MyCelery.tasks

MyCelery.tasks

Task registry.

Warning: Accessing this attribute will also auto-finalize the app.

app.celery.MyCelery.timezone

MyCelery.timezone

Current timezone for this app.

This is a cached property taking the time zone from the **:setting:`timezone`** setting.

app.celery.MyCelery.user_options

MyCelery.user_options = None

Methods

<code>MyCelery.__init__([main, loader, back- end, ...])</code>	Initialize self.
<code>MyCelery.add_defaults(fun)</code>	Add default configuration from dict d.
<code>MyCelery.add_periodic_task(schedule, sig[, ...])</code>	
<code>MyCelery.autodiscover_tasks([packageA, ...])</code>	Auto-discover task modules.
<code>MyCelery.broker_connection([hostname, ...])</code>	Establish a connection to the message broker.
<code>MyCelery.bugreport()</code>	Return information useful in bug reports.
<code>MyCelery.close()</code>	Clean up after the application.
<code>MyCelery.config_from_cmdline(argv[, namespace])</code>	
<code>MyCelery.config_from_envvar(variable, Read ...])</code>	Read configuration from environment variable.
<code>MyCelery.config_from_object(obj[, silent, ...])</code>	Read configuration from object.

continues on next page

Table 63 – continued from previous page

<code>MyCelery.connection([hostname, userid, ...])</code>	Establish a connection to the message broker.
<code>MyCelery.connection_for_read([url])</code>	Establish connection used for consuming.
<code>MyCelery.connection_for_write([url])</code>	Establish connection used for producing.
<code>MyCelery.connection_or_acquire([connection, ...])</code>	Context used to acquire a connection from the pool.
<code>MyCelery.create_task_cls()</code>	Create a base task class bound to this app.
<code>MyCelery.default_connection([connection, pool])</code>	Context used to acquire a connection from the pool.
<code>MyCelery.default_producer([producer])</code>	Context used to acquire a producer from the pool.
<code>MyCelery.either(default_key, *defaults)</code>	Get key from configuration or use default values.
<code>MyCelery.finalize([auto])</code>	Finalize the app.
<code>MyCelery.gen_task_name(name, module)</code>	New task default automatic naming.
<code>MyCelery.now()</code>	Return the current time and date as a datetime.
<code>MyCelery.on_init()</code>	Optional callback called at init.
<code>MyCelery.prepare_config(c)</code>	Prepare configuration before it is merged with the defaults.
<code>MyCelery.producer_or_acquire([producer, pool])</code>	Context used to acquire a producer from the pool.
<code>MyCelery.register_task(task)</code>	Utility for registering a task-based class.
<code>MyCelery.select_queues([queues])</code>	Select subset of queues.
<code>MyCelery.send_task(name[, args, kwargs, ...])</code>	Send task by name.
<code>MyCelery.set_current()</code>	Make this the current app for this thread.
<code>MyCelery.set_default()</code>	Make this the default app for all threads.
<code>MyCelery.setup_security([...])</code>	Setup the message-signing serializer.
<code>MyCelery.signature(*args, **kwargs)</code>	Return a new <code>Signature</code> bound to this app.
<code>MyCelery.subclass_with_self(Class[, name, ...])</code>	Subclass an app-compatible class.
<code>MyCelery.task(*args, **opts)</code>	Decorator to create a task class out of any callable.
<code>MyCelery.uses_utc_timezone()</code>	Check if the application uses the UTC time-zone.

app.celery.MyCelery.__init__

`MyCelery.__init__(main=None, loader=None, backend=None, amqp=None, events=None, log=None, control=None, set_as_current=True, tasks=None, broker=None, include=None, changes=None, config_source=None, fixups=None, task_cls=None, autofinalize=True, namespace=None, strict_typing=True, **kwargs)`

Initialize self. See `help(type(self))` for accurate signature.

app.celery.MyCelery.add_defaults

MyCelery.**add_defaults** (*fun*)

Add default configuration from dict *d*.

If the argument is a callable function then it will be regarded as a promise, and it won't be loaded until the configuration is actually needed.

This method can be compared to:

```
>>> celery.conf.update(d)
```

with a difference that 1) no copy will be made and 2) the dict will not be transferred when the worker spawns child processes, so it's important that the same configuration happens at import time when pickle restores the object on the other side.

app.celery.MyCelery.add_periodic_task

MyCelery.**add_periodic_task** (*schedule*, *sig*, *args=()*, *kwargs=()*, *name=None*, ***opts*)

app.celery.MyCelery.autodiscover_tasks

MyCelery.**autodiscover_tasks** (*packages=None*, *related_name='tasks'*, *force=False*)

Auto-discover task modules.

Searches a list of packages for a “tasks.py” module (or use *related_name* argument).

If the name is empty, this will be delegated to fix-ups (e.g., Django).

For example if you have a directory layout like this:

```
foo/__init__.py
    tasks.py
    models.py

bar/__init__.py
    tasks.py
    models.py

baz/__init__.py
    models.py
```

Then calling `app.autodiscover_tasks(['foo', 'bar', 'baz'])` will result in the modules `foo.tasks` and `bar.tasks` being imported.

Parameters

- **packages** (*List[str]*) – List of packages to search. This argument may also be a callable, in which case the value returned is used (for lazy evaluation).
- **related_name** (*Optional[str]*) – The name of the module to find. Defaults to “tasks”: meaning “look for ‘module.tasks’ for every module in packages.”. If *None* will only try to import the package, i.e. “look for ‘module’”.
- **force** (*bool*) – By default this call is lazy so that the actual auto-discovery won't happen until an application imports the default modules. Forcing will cause the auto-discovery to happen immediately.

app.celery.MyCelery.broker_connection

```
MyCelery.broker_connection(hostname=None, userid=None, password=None,
                           virtual_host=None, port=None, ssl=None, connect_timeout=None,
                           transport=None, transport_options=None, heartbeat=None, login_method=None,
                           failover_strategy=None, **kwargs)
```

Establish a connection to the message broker.

Please use `connection_for_read()` and `connection_for_write()` instead, to convey the intent of use for this connection.

Parameters

- **url** – Either the URL or the hostname of the broker to use.
- **hostname** (*str*) – URL, Hostname/IP-address of the broker. If a URL is used, then the other argument below will be taken from the URL instead.
- **userid** (*str*) – Username to authenticate as.
- **password** (*str*) – Password to authenticate with.
- **virtual_host** (*str*) – Virtual host to use (domain).
- **port** (*int*) – Port to connect to.
- **ssl** (*bool*, *Dict*) – Defaults to the `:setting:`broker_use_ssl`` setting.
- **transport** (*str*) – defaults to the `:setting:`broker_transport`` setting.
- **transport_options** (*Dict*) – Dictionary of transport specific options.
- **heartbeat** (*int*) – AMQP Heartbeat in seconds (pyamqp only).
- **login_method** (*str*) – Custom login method to use (AMQP only).
- **failover_strategy** (*str*, *Callable*) – Custom failover strategy.
- ****kwargs** – Additional arguments to `kombu.Connection`.

Returns the lazy connection instance.

Return type `kombu.Connection`

app.celery.MyCelery.bugreport

```
MyCelery.bugreport()
```

Return information useful in bug reports.

app.celery.MyCelery.close

```
MyCelery.close()
```

Clean up after the application.

Only necessary for dynamically created apps, and you should probably use the `with` statement instead.

Example

```
>>> with Celery(set_as_current=False) as app:
...     with app.connection_for_write() as conn:
...         pass
```

app.celery.MyCelery.config_from_cmdline

`MyCelery.config_from_cmdline(argv, namespace='celery')`

app.celery.MyCelery.config_from_envvar

`MyCelery.config_from_envvar(variable_name, silent=False, force=False)`

Read configuration from environment variable.

The value of the environment variable must be the name of a module to import.

Example

```
>>> os.environ['CELERY_CONFIG_MODULE'] = 'myapp.celeryconfig'
>>> celery.config_from_envvar('CELERY_CONFIG_MODULE')
```

app.celery.MyCelery.config_from_object

`MyCelery.config_from_object(obj, silent=False, force=False, namespace=None)`

Read configuration from object.

Object is either an actual object or the name of a module to import.

Example

```
>>> celery.config_from_object('myapp.celeryconfig')
```

```
>>> from myapp import celeryconfig
>>> celery.config_from_object(celeryconfig)
```

Parameters

- **silent** (*bool*) – If true then import errors will be ignored.
- **force** (*bool*) – Force reading configuration immediately. By default the configuration will be read only when required.

app.celery.MyCelery.connection

`MyCelery.connection` (*hostname=None, userid=None, password=None, virtual_host=None, port=None, ssl=None, connect_timeout=None, transport=None, transport_options=None, heartbeat=None, login_method=None, failover_strategy=None, **kwargs*)

Establish a connection to the message broker.

Please use `connection_for_read()` and `connection_for_write()` instead, to convey the intent of use for this connection.

Parameters

- **url** – Either the URL or the hostname of the broker to use.
- **hostname** (*str*) – URL, Hostname/IP-address of the broker. If a URL is used, then the other argument below will be taken from the URL instead.
- **userid** (*str*) – Username to authenticate as.
- **password** (*str*) – Password to authenticate with.
- **virtual_host** (*str*) – Virtual host to use (domain).
- **port** (*int*) – Port to connect to.
- **ssl** (*bool, Dict*) – Defaults to the `:setting:`broker_use_ssl`` setting.
- **transport** (*str*) – defaults to the `:setting:`broker_transport`` setting.
- **transport_options** (*Dict*) – Dictionary of transport specific options.
- **heartbeat** (*int*) – AMQP Heartbeat in seconds (pyamqp only).
- **login_method** (*str*) – Custom login method to use (AMQP only).
- **failover_strategy** (*str, Callable*) – Custom failover strategy.
- ****kwargs** – Additional arguments to `kombu.Connection`.

Returns the lazy connection instance.

Return type `kombu.Connection`

app.celery.MyCelery.connection_for_read

`MyCelery.connection_for_read` (*url=None, **kwargs*)

Establish connection used for consuming.

See also:

`connection()` for supported arguments.

app.celery.MyCelery.connection_for_write

`MyCelery.connection_for_write` (*url=None, **kwargs*)

Establish connection used for producing.

See also:

`connection()` for supported arguments.

app.celery.MyCelery.connection_or_acquire

`MyCelery.connection_or_acquire (connection=None, pool=True, *_ , ** __)`

Context used to acquire a connection from the pool.

For use within a `with` statement to get a connection from the pool if one is not already provided.

Parameters **connection** (*kombu.Connection*) – If not provided, a connection will be acquired from the connection pool.

app.celery.MyCelery.create_task_cls

`MyCelery.create_task_cls ()`

Create a base task class bound to this app.

app.celery.MyCelery.default_connection

`MyCelery.default_connection (connection=None, pool=True, *_ , ** __)`

Context used to acquire a connection from the pool.

For use within a `with` statement to get a connection from the pool if one is not already provided.

Parameters **connection** (*kombu.Connection*) – If not provided, a connection will be acquired from the connection pool.

app.celery.MyCelery.default_producer

`MyCelery.default_producer (producer=None)`

Context used to acquire a producer from the pool.

For use within a `with` statement to get a producer from the pool if one is not already provided

Parameters **producer** (*kombu.Producer*) – If not provided, a producer will be acquired from the producer pool.

app.celery.MyCelery.either

`MyCelery.either (default_key, *defaults)`

Get key from configuration or use default values.

Fallback to the value of a configuration key if none of the **values* are true.

app.celery.MyCelery.finalize

`MyCelery.finalize (auto=False)`

Finalize the app.

This loads built-in tasks, evaluates pending task decorators, reads configuration, etc.

app.celery.MyCelery.gen_task_name

MyCelery.gen_task_name(name, module)

New task default automatic naming.

The default gen_task_name method builds task names based on absolute imports, for example:

project / `/__init__.py` /moduleA/
 `/__init.py` /tasks.py

 /moduleB/ `/__init.py` /tasks.py

The default automatic naming is “project.moduleA.tasks.taskA”, “project.moduleA.tasks.taskB”, etc. This new default automatic naming forget “tasks” in all task names:

DEFAULT WAY	NEW WAY	project.moduleA.tasks.taskA	project.moduleA.taskA
project.moduleA.tasks.taskA		project.moduleA.taskB	project.moduleB.tasks.taskA
project.moduleB.taskA			

This method is only used when the tasks don’t have a name attribute defined, otherwise, the task name will be respect.

https://docs.celeryproject.org/en/stable/userguide/tasks.html?highlight=gen_task_name#changing-the-automatic-naming-behavior

app.celery.MyCelery.now

MyCelery.now()

Return the current time and date as a datetime.

app.celery.MyCelery.on_init

MyCelery.on_init()

Optional callback called at init.

app.celery.MyCelery.prepare_config

MyCelery.prepare_config(c)

Prepare configuration before it is merged with the defaults.

app.celery.MyCelery.producer_or_acquire

MyCelery.producer_or_acquire(producer=None)

Context used to acquire a producer from the pool.

For use within a with statement to get a producer from the pool if one is not already provided

Parameters **producer** (*kombu.Producer*) – If not provided, a producer will be acquired from the producer pool.

app.celery.MyCelery.register_task

`MyCelery.register_task(task)`
 Utility for registering a task-based class.

Note: This is here for compatibility with old Celery 1.0 style task classes, you should not need to use this for new projects.

app.celery.MyCelery.select_queues

`MyCelery.select_queues(queues=None)`
 Select subset of queues.
Parameters `queues` (*Sequence[str]*) – a list of queue names to keep.

app.celery.MyCelery.send_task

`MyCelery.send_task(name, args=None, kwargs=None, countdown=None, eta=None, task_id=None, producer=None, connection=None, router=None, result_cls=None, expires=None, publisher=None, link=None, link_error=None, add_to_parent=True, group_id=None, group_index=None, retries=0, chord=None, reply_to=None, time_limit=None, soft_time_limit=None, root_id=None, parent_id=None, route_name=None, shadow=None, chain=None, task_type=None, **options)`

Send task by name.

Supports the same arguments as `@-Task.apply_async()`.

Parameters

- **name** (*str*) – Name of task to call (e.g., “*tasks.add*”).
- **result_cls** (*AsyncResult*) – Specify custom result class.

app.celery.MyCelery.set_current

`MyCelery.set_current()`
 Make this the current app for this thread.

app.celery.MyCelery.set_default

`MyCelery.set_default()`
 Make this the default app for all threads.

app.celery.MyCelery.setup_security

`MyCelery.setup_security` (*allowed_serializers=None, key=None, cert=None, store=None, digest='sha256', serializer='json'*)

Setup the message-signing serializer.

This will affect all application instances (a global operation).

Disables untrusted serializers and if configured to use the `auth` serializer will register the `auth` serializer with the provided settings into the Kombu serializer registry.

Parameters

- **allowed_serializers** (*Set[str]*) – List of serializer names, or content_types that should be exempt from being disabled.
- **key** (*str*) – Name of private key file to use. Defaults to the `:setting:security_key` setting.
- **cert** (*str*) – Name of certificate file to use. Defaults to the `:setting:security_certificate` setting.
- **store** (*str*) – Directory containing certificates. Defaults to the `:setting:security_cert_store` setting.
- **digest** (*str*) – Digest algorithm used when signing messages. Default is `sha256`.
- **serializer** (*str*) – Serializer used to encode messages after they've been signed. See `:setting:task_serializer` for the serializers supported. Default is `json`.

app.celery.MyCelery.signature

`MyCelery.signature` (**args, **kwargs*)

Return a new Signature bound to this app.

app.celery.MyCelery.subclass_with_self

`MyCelery.subclass_with_self` (*Class, name=None, attribute='app', reverse=None, keep_reduce=False, **kw*)

Subclass an app-compatible class.

App-compatible means that the class has a class attribute that provides the default app it should use, for example: `class Foo: app = None`.

Parameters

- **Class** (*type*) – The app-compatible class to subclass.
- **name** (*str*) – Custom name for the target class.
- **attribute** (*str*) – Name of the attribute holding the app, Default is `'app'`.
- **reverse** (*str*) – Reverse path to this object used for pickling purposes. For example, to get `app.AsyncResult`, use `"AsyncResult"`.
- **keep_reduce** (*bool*) – If enabled a custom `__reduce__` implementation won't be provided.

app.celery.MyCelery.task

MyCelery.**task** (*args, **opts)

Decorator to create a task class out of any callable.

See Task options for a list of the arguments that can be passed to this decorator.

Examples

```
@app.task
def refresh_feed(url):
    store_feed(feedparser.parse(url))
```

with setting extra options:

```
@app.task(exchange='feeds')
def refresh_feed(url):
    return store_feed(feedparser.parse(url))
```

Note: App Binding: For custom apps the task decorator will return a proxy object, so that the act of creating the task is not performed until the task is used or the task registry is accessed.

If you're depending on binding to be deferred, then you must not access any attributes on the returned object until the application is fully set up (finalized).

app.celery.MyCelery.uses_utc_timezone

MyCelery.**uses_utc_timezone**()

Check if the application uses the UTC timezone.

Functions

init_celery(app)

app.celery.init_celery

app.celery.**init_celery** (app: flask.app.Flask) → celery.app.base.Celery

Exceptions

TaskFailure

app.celery.TaskFailure**exception** app.celery.TaskFailure**class** app.celery.ContextTask**AsyncResult** (*task_id*, ***kwargs*)

Get AsyncResult instance for the specified task.

Parameters **task_id** (*str*) – Task id to get result for.**exception** MaxRetriesExceededError (**args*, ***kwargs*)

The tasks max restart limit has been exceeded.

args**with_traceback** ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception OperationalError

Recoverable message transport connection error.

args**with_traceback** ()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

Request = 'celery.worker.request:Request'**Strategy** = 'celery.worker.strategy:default'**_app** = <MyCelery __main__>**_backend** = None**_default_request** = None**_exec_options** = None**classmethod** _get_app ()**_get_exec_options** ()**_get_request** ()

Get current request object.

abstract = True**acks_late** = False**acks_on_failure_or_timeout** = True**classmethod** add_around (*attr*, *around*)**add_to_chord** (*sig*, *lazy=False*)

Add signature to the chord the current task is a member of.

New in version 4.0.

Currently only supported by the Redis result backend.

Parameters

- **sig** (~@Signature) – Signature to extend chord with.
- **lazy** (*bool*) – If enabled the new task won't actually be called, and sig.delay() must be called manually.

add_trail (*result*)

after_return (*status, retval, task_id, args, kwargs, einfo*)

Handler called after the task returns.

Parameters

- **status** (*str*) – Current task state.
- **retval** (*Any*) – Task return value/exception.
- **task_id** (*str*) – Unique id of the task.
- **args** (*Tuple*) – Original arguments for the task.
- **kwargs** (*Dict*) – Original keyword arguments for the task.
- **einfo** (*ExceptionInfo*) – Exception information.

Returns The return value of this handler is ignored.

Return type None

classmethod annotate ()

app = <MyCelery __main__>

apply (*args=None, kwargs=None, link=None, link_error=None, task_id=None, retries=None, throw=None, logfile=None, loglevel=None, headers=None, **options*)

Execute this task locally, by blocking until the task returns.

Parameters

- **args** (*Tuple*) – positional arguments passed on to the task.
- **kwargs** (*Dict*) – keyword arguments passed on to the task.
- **throw** (*bool*) – Re-raise task exceptions. Defaults to the **:setting:task_eager_propagates** setting.

Returns pre-evaluated result.

Return type celery.result.EagerResult

apply_async (*args=None, kwargs=None, task_id=None, producer=None, link=None, link_error=None, shadow=None, **options*)

Apply tasks asynchronously by sending a message.

Parameters

- **args** (*Tuple*) – The positional arguments to pass on to the task.
- **kwargs** (*Dict*) – The keyword arguments to pass on to the task.
- **countdown** (*float*) – Number of seconds into the future that the task should execute. Defaults to immediate execution.
- **eta** (*datetime*) – Absolute time and date of when the task should be executed. May not be specified if *countdown* is also supplied.
- **expires** (*float, datetime*) – Datetime or seconds in the future for the task should expire. The task won't be executed after the expiration time.
- **shadow** (*str*) – Override task name used in logs/monitoring. Default is retrieved from `shadow_name()`.
- **connection** (*kombu.Connection*) – Re-use existing broker connection instead of acquiring one from the connection pool.

- **retry** (*bool*) – If enabled sending of the task message will be retried in the event of connection loss or failure. Default is taken from the `:setting:`task_publish_retry`` setting. Note that you need to handle the producer/connection manually for this to work.
- **retry_policy** (*Mapping*) – Override the retry policy used. See the `:setting:`task_publish_retry_policy`` setting.
- **queue** (*str*, *kombu.Queue*) – The queue to route the task to. This must be a key present in `:setting:`task_queues``, or `:setting:`task_create_missing_queues`` must be enabled. See guide-routing for more information.
- **exchange** (*str*, *kombu.Exchange*) – Named custom exchange to send the task to. Usually not used in combination with the `queue` argument.
- **routing_key** (*str*) – Custom routing key used to route the task to a worker server. If in combination with a `queue` argument only used to specify custom routing keys to topic exchanges.
- **priority** (*int*) – The task priority, a number between 0 and 9. Defaults to the `priority` attribute.
- **serializer** (*str*) – Serialization method to use. Can be *pickle*, *json*, *yaml*, *msgpack* or any custom serialization method that's been registered with `kombu.serialization.registry`. Defaults to the `serializer` attribute.
- **compression** (*str*) – Optional compression method to use. Can be one of *zlib*, *bzip2*, or any custom compression methods registered with `kombu.compression.register()`. Defaults to the `:setting:`task_compression`` setting.
- **link** (*Signature*) – A single, or a list of tasks signatures to apply if the task returns successfully.
- **link_error** (*Signature*) – A single, or a list of task signatures to apply if an error occurs while executing the task.
- **producer** (*kombu.Producer*) – custom producer to use when publishing the task.
- **add_to_parent** (*bool*) – If set to *True* (default) and the task is applied while executing another task, then the result will be appended to the parent tasks `request.children` attribute. Trailing can also be disabled by default using the `trail` attribute
- **publisher** (*kombu.Producer*) – Deprecated alias to `producer`.
- **headers** (*Dict*) – Message headers to be included in the message.

Returns Promise of future evaluation.

Return type `celery.result.AsyncResult`

Raises

- **TypeError** – If not enough arguments are passed, or too many arguments are passed. Note that signature checks may be disabled by specifying `@task(typing=False)`.
- **kombu.exceptions.OperationalError** – If a connection to the transport cannot be made, or if the connection is lost.

Note: Also supports all keyword arguments supported by `kombu.Producer.publish()`.

autoregister = True

property backend

classmethod `bind(app)`

chunks (*it*, *n*)

Create a `chunks` task for this task.

default_retry_delay = 180

delay (**args*, ***kwargs*)

Star argument version of `apply_async()`.

Does not support the extra options enabled by `apply_async()`.

Parameters

- ***args** (*Any*) – Positional arguments passed on to the task.
- ****kwargs** (*Any*) – Keyword arguments passed on to the task.

Returns Future promise.

Return type `celery.result.AsyncResult`

expires = None

from_config = (('serializer', 'task_serializer'), ('rate_limit', 'task_default_rate_li

ignore_result = False

map (*it*)

Create a `xmap` task from *it*.

max_retries = 3

name = None

classmethod `on_bound(app)`

Called when the task is bound to an app.

Note: This class method can be defined to do additional actions when the task class is bound to an app.

on_failure (*exc*, *task_id*, *args*, *kwargs*, *info*) → None

Parameters

- **exc** – The exception raised by the task.
- **task_id** – Unique id of the failed task.
- **args** – Original arguments for the task that failed.
- **kwargs** – Original keyword arguments for the task that failed.
- **info** – `ExceptionInfo` instance, containing the traceback

Returns None

on_retry (*exc*, *task_id*, *args*, *kwargs*, *info*)

Retry handler.

This is run by the worker when the task is to be retried.

Parameters

- **exc** (*Exception*) – The exception sent to `retry()`.
- **task_id** (*str*) – Unique id of the retried task.

- **args** (*Tuple*) – Original arguments for the retried task.
- **kwargs** (*Dict*) – Original keyword arguments for the retried task.
- **einfo** (*ExceptionInfo*) – Exception information.

Returns The return value of this handler is ignored.

Return type None

on_success (*retval, task_id, args, kwargs*)

Success handler.

Run by the worker if the task executes successfully.

Parameters

- **retval** (*Any*) – The return value of the task.
- **task_id** (*str*) – Unique id of the executed task.
- **args** (*Tuple*) – Original arguments for the executed task.
- **kwargs** (*Dict*) – Original keyword arguments for the executed task.

Returns The return value of this handler is ignored.

Return type None

pop_request ()

priority = None

push_request (**args, **kwargs*)

rate_limit = None

reject_on_worker_lost = None

replace (*sig*)

Replace this task, with a new task inheriting the task id.

Execution of the host task ends immediately and no subsequent statements will be run.

New in version 4.0.

Parameters **sig** (~@Signature) – signature to replace with.

Raises

- **~@Ignore** – This is always raised when called in asynchronous context.
- **It is best to always use `return self.replace(..)` to convey –**
- **to the reader that the task won't continue after being replaced. –**

property request

Get current request object.

request_stack = <celery.utils.threads._LocalStack object>

resultrepr_maxsize = 1024

retry (*args=None, kwargs=None, exc=None, throw=True, eta=None, countdown=None, max_retries=None, **options*)

Retry the task, adding it to the back of the queue.

Example

```
>>> from imaginary_twitter_lib import Twitter
>>> from proj.celery import app

>>> @app.task(bind=True)
... def tweet(self, auth, message):
...     twitter = Twitter(oauth=auth)
...     try:
...         twitter.post_status_update(message)
...     except twitter.FailWhale as exc:
...         # Retry in 5 minutes.
...         self.retry(countdown=60 * 5, exc=exc)
```

Note: Although the task will never return above as `retry` raises an exception to notify the worker, we use `raise` in front of the `retry` to convey that the rest of the block won't be executed.

Parameters

- **args** (*Tuple*) – Positional arguments to retry with.
- **kwargs** (*Dict*) – Keyword arguments to retry with.
- **exc** (*Exception*) – Custom exception to report when the max retry limit has been exceeded (default: `@MaxRetriesExceededError`).

If this argument is set and `retry` is called while an exception was raised (`sys.exc_info()` is set) it will attempt to re-raise the current exception.

If no exception was raised it will raise the `exc` argument provided.

- **countdown** (*float*) – Time in seconds to delay the retry for.
- **eta** (*datetime*) – Explicit time and date to run the retry at.
- **max_retries** (*int*) – If set, overrides the default retry limit for this execution. Changes to this parameter don't propagate to subsequent task retry attempts. A value of `None`, means “use the default”, so if you want infinite retries you'd have to set the `max_retries` attribute of the task to `None` first.
- **time_limit** (*int*) – If set, overrides the default time limit.
- **soft_time_limit** (*int*) – If set, overrides the default soft time limit.
- **throw** (*bool*) – If this is `False`, don't raise the `@Retry` exception, that tells the worker to mark the task as being retried. Note that this means the task will be marked as failed if the task raises an exception, or successful if it returns after the retry call.
- ****options** (*Any*) – Extra options to pass on to `apply_async()`.

Raises `celery.exceptions.Retry` – To tell the worker that the task has been re-sent for retry. This always happens, unless the `throw` keyword argument has been explicitly set to `False`, and is considered normal operation.

run (**args, **kwargs*)

The body of the task executed by workers.

s (**args, **kwargs*)

Create signature.

Shortcut for `.s(*a, **k) -> .signature(a, k)`.

send_event (*type_*, *retry=True*, *retry_policy=None*, ***fields*)
Send monitoring event message.

This can be used to add custom event types in **:pypi:Flower** and other monitors.

Parameters *type* (*str*) – Type of event, e.g. "task-failed".

Keyword Arguments

- **retry** (*bool*) – Retry sending the message if the connection is lost. Default is taken from the **:setting:task_publish_retry** setting.
- **retry_policy** (*Mapping*) – Retry settings. Default is taken from the **:setting:task_publish_retry_policy** setting.
- ****fields** (*Any*) – Map containing information about the event. Must be JSON serializable.

send_events = `True`

serializer = `'json'`

shadow_name (*args*, *kwargs*, *options*)
Override for custom task name in worker logs/monitoring.

Example

```
from celery.utils.imports import qualname

def shadow_name(task, args, kwargs, options):
    return qualname(args[0])

@app.task(shadow_name=shadow_name, serializer='pickle')
def apply_function_async(fun, *args, **kwargs):
    return fun(*args, **kwargs)
```

Parameters

- **args** (*Tuple*) – Task positional arguments.
- **kwargs** (*Dict*) – Task keyword arguments.
- **options** (*Dict*) – Task execution options.

si (**args*, ***kwargs*)
Create immutable signature.

Shortcut for `.si(*a, **k) -> .signature(a, k, immutable=True)`.

signature (*args=None*, **starargs*, ***starkwargs*)
Create signature.

Returns

object for this task, wrapping arguments and execution options for a single task invocation.

Return type `signature`

signature_from_request (*request=None*, *args=None*, *kwargs=None*, *queue=None*, ***extra_options*)

soft_time_limit = None

starmap (*it*)

Create a `xstarmap` task from *it*.

start_strategy (*app*, *consumer*, ***kwargs*)

store_errors_even_if_ignored = False

subtask (*args=None*, **starargs*, ***starkwargs*)

Create signature.

Returns

object for this task, wrapping arguments and execution options for a single task invocation.

Return type `signature`

subtask_from_request (*request=None*, *args=None*, *kwargs=None*, *queue=None*, ***extra_options*)

throws = ()

time_limit = None

track_started = False

trail = True

typing = True

update_state (*task_id=None*, *state=None*, *meta=None*, ***kwargs*)

Update task state.

Parameters

- **task_id** (*str*) – Id of the task to update. Defaults to the id of the current task.
- **state** (*str*) – New state.
- **meta** (*Dict*) – State meta-data.

class `app.celery.MyCelery` (*main=None*, *loader=None*, *backend=None*, *amqp=None*, *events=None*, *log=None*, *control=None*, *set_as_current=True*, *tasks=None*, *broker=None*, *include=None*, *changes=None*, *config_source=None*, *fixups=None*, *task_cls=None*, *autofinalize=True*, *namespace=None*, *strict_typing=True*, ***kwargs*)

AsyncResult

Create new result instance.

See also:

`celery.result.AsyncResult`.

Beat

celery beat scheduler application.

See also:

`@Beat`.

GroupResult

Create new group result instance.

See also:

`celery.result.GroupResult`.

```
IS_WINDOWS = False
IS_macOS = False
Pickler
    alias of celery.app.utils.AppPickler
ResultSet
SYSTEM = 'Linux'
Task
    Base task class for this app.
WorkController
    Embeddable worker.
    See also:
        @WorkController.
Worker
    Worker application.
    See also:
        @Worker.
_acquire_connection(pool=True)
    Helper for connection\_or\_acquire\(\).
_add_periodic_task(key, entry)
_after_fork()
_after_fork_registered = False
_autodiscover_tasks(packages, related_name, **kwargs)
_autodiscover_tasks_from_fixups(related_name)
_autodiscover_tasks_from_names(packages, related_name)
_canvas
_conf = None
_connection(url, userid=None, password=None, virtual_host=None, port=None, ssl=None, connect_timeout=None, transport=None, transport_options=None, heartbeat=None, login_method=None, failover_strategy=None, **kwargs)
_ensure_after_fork()
_finalize_pending_conf()
    Get config value by key and finalize loading the configuration.


---


Note:
    This is used by PendingConfiguration: as soon as you access a key the configuration is read.


---


_fixups = None
_get_backend()
_get_default_loader()
_load_config()
```

```

_pool = None
_rgetattr(path)
_sig_to_periodic_task_entry(schedule, sig, args=(), kwargs=None, name=None, **opts)
_task_from_fun(fun, name=None, base=None, bind=False, **options)
add_defaults(fun)

```

Add default configuration from dict d.

If the argument is a callable function then it will be regarded as a promise, and it won't be loaded until the configuration is actually needed.

This method can be compared to:

```
>>> celery.conf.update(d)
```

with a difference that 1) no copy will be made and 2) the dict will not be transferred when the worker spawns child processes, so it's important that the same configuration happens at import time when pickle restores the object on the other side.

```
add_periodic_task(schedule, sig, args=(), kwargs=(), name=None, **opts)
```

```
amqp
    @amqp.
```

Type AMQP related functionality

```
amqp_cls = 'celery.app.amqp:AMQP'
```

annotations

```
autodiscover_tasks(packages=None, related_name='tasks', force=False)
```

Auto-discover task modules.

Searches a list of packages for a “tasks.py” module (or use related_name argument).

If the name is empty, this will be delegated to fix-ups (e.g., Django).

For example if you have a directory layout like this:

```

foo/__init__.py
  tasks.py
  models.py

bar/__init__.py
  tasks.py
  models.py

baz/__init__.py
  models.py

```

Then calling `app.autodiscover_tasks(['foo', 'bar', 'baz'])` will result in the modules `foo.tasks` and `bar.tasks` being imported.

Parameters

- **packages** (*List[str]*) – List of packages to search. This argument may also be a callable, in which case the value returned is used (for lazy evaluation).
- **related_name** (*Optional[str]*) – The name of the module to find. Defaults to “tasks”: meaning “look for ‘module.tasks’ for every module in packages.”. If None will only try to import the package, i.e. “look for ‘module’”.

- **force** (*bool*) – By default this call is lazy so that the actual auto-discovery won't happen until an application imports the default modules. Forcing will cause the auto-discovery to happen immediately.

backend

Current backend instance.

backend_cls = `None`

broker_connection (*hostname=None, userid=None, password=None, virtual_host=None, port=None, ssl=None, connect_timeout=None, transport=None, transport_options=None, heartbeat=None, login_method=None, failover_strategy=None, **kwargs*)

Establish a connection to the message broker.

Please use `connection_for_read()` and `connection_for_write()` instead, to convey the intent of use for this connection.

Parameters

- **url** – Either the URL or the hostname of the broker to use.
- **hostname** (*str*) – URL, Hostname/IP-address of the broker. If a URL is used, then the other argument below will be taken from the URL instead.
- **userid** (*str*) – Username to authenticate as.
- **password** (*str*) – Password to authenticate with
- **virtual_host** (*str*) – Virtual host to use (domain).
- **port** (*int*) – Port to connect to.
- **ssl** (*bool, Dict*) – Defaults to the `:setting:`broker_use_ssl`` setting.
- **transport** (*str*) – defaults to the `:setting:`broker_transport`` setting.
- **transport_options** (*Dict*) – Dictionary of transport specific options.
- **heartbeat** (*int*) – AMQP Heartbeat in seconds (pyamqp only).
- **login_method** (*str*) – Custom login method to use (AMQP only).
- **failover_strategy** (*str, Callable*) – Custom failover strategy.
- ****kwargs** – Additional arguments to `kombu.Connection`.

Returns the lazy connection instance.

Return type `kombu.Connection`

bugreport ()

Return information useful in bug reports.

builtin_fixups = `{'celery.fixups.django:fixup'}`

close ()

Clean up after the application.

Only necessary for dynamically created apps, and you should probably use the `with` statement instead.

Example

```
>>> with Celery(set_as_current=False) as app:
...     with app.connection_for_write() as conn:
...         pass
```

property `conf`

Current configuration.

config_from_cmdline (*argv*, *namespace*='celery')

config_from_envvar (*variable_name*, *silent*=False, *force*=False)

Read configuration from environment variable.

The value of the environment variable must be the name of a module to import.

Example

```
>>> os.environ['CELERY_CONFIG_MODULE'] = 'myapp.celeryconfig'
>>> celery.config_from_envvar('CELERY_CONFIG_MODULE')
```

config_from_object (*obj*, *silent*=False, *force*=False, *namespace*=None)

Read configuration from object.

Object is either an actual object or the name of a module to import.

Example

```
>>> celery.config_from_object('myapp.celeryconfig')
```

```
>>> from myapp import celeryconfig
>>> celery.config_from_object(celeryconfig)
```

Parameters

- **silent** (*bool*) – If true then import errors will be ignored.
- **force** (*bool*) – Force reading configuration immediately. By default the configuration will be read only when required.

connection (*hostname*=None, *userid*=None, *password*=None, *virtual_host*=None, *port*=None, *ssl*=None, *connect_timeout*=None, *transport*=None, *transport_options*=None, *heartbeat*=None, *login_method*=None, *failover_strategy*=None, ***kwargs*)

Establish a connection to the message broker.

Please use `connection_for_read()` and `connection_for_write()` instead, to convey the intent of use for this connection.

Parameters

- **url** – Either the URL or the hostname of the broker to use.
- **hostname** (*str*) – URL, Hostname/IP-address of the broker. If a URL is used, then the other argument below will be taken from the URL instead.
- **userid** (*str*) – Username to authenticate as.
- **password** (*str*) – Password to authenticate with

- **virtual_host** (*str*) – Virtual host to use (domain).
- **port** (*int*) – Port to connect to.
- **ssl** (*bool*, *Dict*) – Defaults to the **:setting:`broker_use_ssl`** setting.
- **transport** (*str*) – defaults to the **:setting:`broker_transport`** setting.
- **transport_options** (*Dict*) – Dictionary of transport specific options.
- **heartbeat** (*int*) – AMQP Heartbeat in seconds (pyamqp only).
- **login_method** (*str*) – Custom login method to use (AMQP only).
- **failover_strategy** (*str*, *Callable*) – Custom failover strategy.
- ****kwargs** – Additional arguments to `kombu.Connection`.

Returns the lazy connection instance.

Return type `kombu.Connection`

connection_for_read (*url=None*, ***kwargs*)
Establish connection used for consuming.

See also:

`connection()` for supported arguments.

connection_for_write (*url=None*, ***kwargs*)
Establish connection used for producing.

See also:

`connection()` for supported arguments.

connection_or_acquire (*connection=None*, *pool=True*, ****_*, ***__*)
Context used to acquire a connection from the pool.

For use within a `with` statement to get a connection from the pool if one is not already provided.

Parameters **connection** (`kombu.Connection`) – If not provided, a connection will be acquired from the connection pool.

control
@control.

Type Remote control

control_cls = 'celery.app.control:Control'

create_task_cls ()
Create a base task class bound to this app.

property current_task
Instance of task being executed, or `None`.

property current_worker_task
The task currently being executed by a worker or `None`.

Differs from `current_task` in that it's not affected by tasks calling other tasks directly, or eagerly.

default_connection (*connection=None*, *pool=True*, ****_*, ***__*)
Context used to acquire a connection from the pool.

For use within a `with` statement to get a connection from the pool if one is not already provided.

Parameters connection (*kombu.Connection*) – If not provided, a connection will be acquired from the connection pool.

default_producer (*producer=None*)

Context used to acquire a producer from the pool.

For use within a `with` statement to get a producer from the pool if one is not already provided

Parameters producer (*kombu.Producer*) – If not provided, a producer will be acquired from the producer pool.

either (*default_key, *defaults*)

Get key from configuration or use default values.

Fallback to the value of a configuration key if none of the **values* are true.

events

@events.

Type Consuming and sending events

events_cls = 'celery.app.events:Events'

finalize (*auto=False*)

Finalize the app.

This loads built-in tasks, evaluates pending task decorators, reads configuration, etc.

gen_task_name (*name, module*)

New task default automatic naming.

The default `gen_task_name` method builds task names based on absolute imports, for example:

project / `/__init__.py` /moduleA/

`/__init__.py` /tasks.py

/moduleB/ `/__init__.py` /tasks.py

The default automatic naming is “project.moduleA.tasks.taskA”, “project.moduleA.tasks.taskB”, etc. This new default automatic naming forget “tasks” in all task names:

DEFAULT	WAY	NEW	WAY	project.moduleA.tasks.taskA	project.moduleA.taskA
project.moduleA.tasks.taskA	project.moduleA.taskB	project.moduleB.tasks.taskA	project.moduleB.taskA		

This method is only used when the tasks don’t have a name attribute defined, otherwise, the task name will be respect.

https://docs.celeryproject.org/en/stable/userguide/tasks.html?highlight=gen_task_name#changing-the-automatic-naming-behavior

loader

Current loader instance.

loader_cls = None

log

@log.

Type Logging

log_cls = 'celery.app.log:Logging'

main = None

now()
Return the current time and date as a datetime.

oid
Universally unique identifier for this app.

on_after_configure = None

on_after_finalize = None

on_after_fork = None

on_configure = None

on_init()
Optional callback called at init.

property pool
@pool.

Note: This attribute is not related to the workers concurrency pool.

Type Broker connection pool

prepare_config(c)
Prepare configuration before it is merged with the defaults.

producer_or_acquire(producer=None)
Context used to acquire a producer from the pool.

For use within a `with` statement to get a producer from the pool if one is not already provided

Parameters producer (*kombu.Producer*) – If not provided, a producer will be acquired from the producer pool.

property producer_pool

register_task(task)
Utility for registering a task-based class.

Note: This is here for compatibility with old Celery 1.0 style task classes, you should not need to use this for new projects.

registry_cls = 'celery.app.registry:TaskRegistry'

select_queues(queues=None)
Select subset of queues.

Parameters queues (*Sequence[str]*) – a list of queue names to keep.

send_task(name, args=None, kwargs=None, countdown=None, eta=None, task_id=None, producer=None, connection=None, router=None, result_cls=None, expires=None, publisher=None, link=None, link_error=None, add_to_parent=True, group_id=None, group_index=None, retries=0, chord=None, reply_to=None, time_limit=None, soft_time_limit=None, root_id=None, parent_id=None, route_name=None, shadow=None, chain=None, task_type=None, **options)
Send task by name.

Supports the same arguments as `@-Task.apply_async()`.

Parameters

- **name** (*str*) – Name of task to call (e.g., “*tasks.add*”).
- **result_cls** (*AsyncResult*) – Specify custom result class.

set_current ()

Make this the current app for this thread.

set_default ()

Make this the default app for all threads.

setup_security (*allowed_serializers=None, key=None, cert=None, store=None, digest='sha256', serializer='json'*)

Setup the message-signing serializer.

This will affect all application instances (a global operation).

Disables untrusted serializers and if configured to use the `auth` serializer will register the `auth` serializer with the provided settings into the Kombu serializer registry.

Parameters

- **allowed_serializers** (*Set[str]*) – List of serializer names, or content_types that should be exempt from being disabled.
- **key** (*str*) – Name of private key file to use. Defaults to the `:setting:`security_key`` setting.
- **cert** (*str*) – Name of certificate file to use. Defaults to the `:setting:`security_certificate`` setting.
- **store** (*str*) – Directory containing certificates. Defaults to the `:setting:`security_cert_store`` setting.
- **digest** (*str*) – Digest algorithm used when signing messages. Default is `sha256`.
- **serializer** (*str*) – Serializer used to encode messages after they’ve been signed. See `:setting:`task_serializer`` for the serializers supported. Default is `json`.

signature (*args, **kwargs)

Return a new Signature bound to this app.

steps = None

subclass_with_self (*Class, name=None, attribute='app', reverse=None, keep_reduce=False, **kw*)

Subclass an app-compatible class.

App-compatible means that the class has a class attribute that provides the default app it should use, for example: `class Foo: app = None`.

Parameters

- **Class** (*type*) – The app-compatible class to subclass.
- **name** (*str*) – Custom name for the target class.
- **attribute** (*str*) – Name of the attribute holding the app, Default is ‘app’.
- **reverse** (*str*) – Reverse path to this object used for pickling purposes. For example, to get `app.AsyncResult`, use “`AsyncResult`”.
- **keep_reduce** (*bool*) – If enabled a custom `__reduce__` implementation won’t be provided.

task (*args, **opts)

Decorator to create a task class out of any callable.

See Task options for a list of the arguments that can be passed to this decorator.

Examples

```
@app.task
def refresh_feed(url):
    store_feed(feedparser.parse(url))
```

with setting extra options:

```
@app.task(exchange='feeds')
def refresh_feed(url):
    return store_feed(feedparser.parse(url))
```

Note: App Binding: For custom apps the task decorator will return a proxy object, so that the act of creating the task is not performed until the task is used or the task registry is accessed.

If you're depending on binding to be deferred, then you must not access any attributes on the returned object until the application is fully set up (finalized).

task_cls = 'celery.app.task:Task'

tasks

Task registry.

Warning: Accessing this attribute will also auto-finalize the app.

timezone

Current timezone for this app.

This is a cached property taking the time zone from the **:setting: `timezone`** setting.

user_options = None

uses_utc_timezone()

Check if the application uses the UTC timezone.

exception app.celery.TaskFailure

args

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

app.celery.**init_celery**(app: flask.app.Flask) → celery.app.base.Celery

2.1.3 app.extensions

Description

Registers third party extensions.

Functions

<code>init_app(app)</code>

app.extensions.init_app

`app.extensions.init_app(app: flask.app.Flask) → None`

`app.extensions.init_app(app: flask.app.Flask) → None`

2.1.4 app.middleware

Description

WSGI middleware for validating requests content type.

Classes

<code>Middleware(app)</code>	WSGI middleware for checking if the request has a valid content type.
------------------------------	---

app.middleware.Middleware

class `app.middleware.Middleware(app: flask.app.Flask)`

Bases: `object`

WSGI middleware for checking if the request has a valid content type.

Methods

<code>Middleware.__init__(app)</code>	Initialize self.
---------------------------------------	------------------

app.middleware.Middleware.__init__

`Middleware.__init__(app: flask.app.Flask)`
Initialize self. See `help(type(self))` for accurate signature.

class `app.middleware.Middleware` (`app: flask.app.Flask`)
WSGI middleware for checking if the request has a valid content type.

static `_parse_content_type` (`request_content_type: any`) \rightarrow str
Content-Type := type "/" subtype *[";" parameter] <https://tools.ietf.org/html/rfc1341>

2.1.5 app.models

Description

Registers database models.

Modules

`app.models.base`

`app.models.document`

`app.models.role`

`app.models.user`

app.models.base

Description

Classes

`Base(*args, **kwargs)`

app.models.base.Base

class `app.models.base.Base` (`*args, **kwargs`)
Bases: `playhouse.flask_utils.FlaskDB.get_model_class.<locals>.BaseModel`

Attributes

`Base.dirty_fields`

`Base.id`

app.models.base.Base.dirty_fields**property** `Base.dirty_fields`**app.models.base.Base.id**`Base.id = <AutoField: Base.id>`**Methods**

<code>Base.__init__(*args, **kwargs)</code>	Initialize self.
<code>Base.add_index(*fields, **kwargs)</code>	
<code>Base.alias([alias])</code>	
<code>Base.bind(database[, bind_refs, bind_backrefs])</code>	
<code>Base.bind_ctx(database[, bind_refs, ...])</code>	
<code>Base.bulk_create(model_list[, batch_size])</code>	
<code>Base.bulk_update(model_list, fields[, ...])</code>	
<code>Base.clone()</code>	
<code>Base.coerce([_coerce])</code>	
<code>Base.copy(method)</code>	
<code>Base.create(**query)</code>	
<code>Base.create_table([safe])</code>	
<code>Base.delete()</code>	
<code>Base.delete_by_id(pk)</code>	
<code>Base.delete_instance([recursive, ...])</code>	
<code>Base.dependencies([search_nullable])</code>	
<code>Base.drop_table([safe, drop_sequences])</code>	
<code>Base.filter(*dq_nodes, **filters)</code>	
<code>Base.get(*query, **filters)</code>	
<code>Base.get_by_id(pk)</code>	
<code>Base.get_fields([exclude, include, sort_order])</code>	
<code>Base.get_id()</code>	
<code>Base.get_or_create(**kwargs)</code>	
<code>Base.get_or_none(*query, **filters)</code>	
<code>Base.index(*fields, **kwargs)</code>	
<code>Base.insert([_Model__data])</code>	
<code>Base.insert_from(query, fields)</code>	
<code>Base.insert_many(rows[, fields])</code>	
<code>Base.is_alias()</code>	
<code>Base.is_dirty()</code>	
<code>Base.noop()</code>	
<code>Base.raw(sql, *params)</code>	
<code>Base.replace([_Model__data])</code>	
<code>Base.replace_many(rows[, fields])</code>	

continues on next page

Table 72 – continued from previous page

<i>Base.save(*args, **kwargs)</i>
<i>Base.select(*fields)</i>
<i>Base.set_by_id(key, value)</i>
<i>Base.table_exists()</i>
<i>Base.truncate_table(**options)</i>
<i>Base.unwrap()</i>
<i>Base.update([_Model_data])</i>
<i>Base.validate_model()</i>

app.models.base.Base.__init__

Base.__init__ (*args, **kwargs)

Initialize self. See help(type(self)) for accurate signature.

app.models.base.Base.add_index

classmethod **Base.add_index** (*fields, **kwargs)

app.models.base.Base.alias

classmethod **Base.alias** (alias=None)

app.models.base.Base.bind

classmethod **Base.bind** (database, bind_refs=True, bind_backrefs=True)

app.models.base.Base.bind_ctx

classmethod **Base.bind_ctx** (database, bind_refs=True, bind_backrefs=True)

app.models.base.Base.bulk_create

classmethod **Base.bulk_create** (model_list, batch_size=None)

app.models.base.Base.bulk_update

classmethod **Base.bulk_update** (model_list, fields, batch_size=None)

app.models.base.Base.clone

`Base.clone()`

app.models.base.Base.coerce

`Base.coerce(_coerce=True)`

app.models.base.Base.copy

static `Base.copy(method)`

app.models.base.Base.create

classmethod `Base.create(**query)`

app.models.base.Base.create_table

classmethod `Base.create_table(safe=True, **options)`

app.models.base.Base.delete

classmethod `Base.delete()`

app.models.base.Base.delete_by_id

classmethod `Base.delete_by_id(pk)`

app.models.base.Base.delete_instance

`Base.delete_instance(recursive=False, delete_nullable=False)`

app.models.base.Base.dependencies

`Base.dependencies(search_nullable=False)`

app.models.base.Base.drop_table**classmethod** Base.**drop_table** (*safe=True, drop_sequences=True, **options*)**app.models.base.Base.filter****classmethod** Base.**filter** (**dq_nodes, **filters*)**app.models.base.Base.get****classmethod** Base.**get** (**query, **filters*)**app.models.base.Base.get_by_id****classmethod** Base.**get_by_id** (*pk*)**app.models.base.Base.get_fields****classmethod** Base.**get_fields** (*exclude: list = None, include: list = None, sort_order: list = None*) → set**app.models.base.Base.get_id**Base.**get_id** ()**app.models.base.Base.get_or_create****classmethod** Base.**get_or_create** (***kwargs*)**app.models.base.Base.get_or_none****classmethod** Base.**get_or_none** (**query, **filters*)**app.models.base.Base.index****classmethod** Base.**index** (**fields, **kwargs*)

app.models.base.Base.insert

```
classmethod Base.insert(_Model__data=None, **insert)
```

app.models.base.Base.insert_from

```
classmethod Base.insert_from(query, fields)
```

app.models.base.Base.insert_many

```
classmethod Base.insert_many(rows, fields=None)
```

app.models.base.Base.is_alias

```
Base.is_alias()
```

app.models.base.Base.is_dirty

```
Base.is_dirty()
```

app.models.base.Base.noop

```
classmethod Base.noop()
```

app.models.base.Base.raw

```
classmethod Base.raw(sql, *params)
```

app.models.base.Base.replace

```
classmethod Base.replace(_Model__data=None, **insert)
```

app.models.base.Base.replace_many

```
classmethod Base.replace_many(rows, fields=None)
```

app.models.base.Base.save

abstract Base.**save** (*args: list, **kwargs: dict) → int

app.models.base.Base.select

classmethod Base.**select** (*fields)

app.models.base.Base.set_by_id

classmethod Base.**set_by_id**(key, value)

app.models.base.Base.table_exists

classmethod Base.**table_exists**()

app.models.base.Base.truncate_table

classmethod Base.**truncate_table**(**options)

app.models.base.Base.unwrap

Base.**unwrap**()

app.models.base.Base.update

classmethod Base.**update** (_Model__data=None, **update)

app.models.base.Base.validate_model

classmethod Base.**validate_model**()

class app.models.base.**Base** (*args, **kwargs)

DoesNotExist

alias of BaseDoesNotExist

_coerce = True

_meta = <peewee.Metadata object>

classmethod **_normalize_data**(data, kwargs)

property **_pk**

_pk_expr()

_populate_unsaved_relations(field_dict)

_prune_fields(field_dict, only)

```

_schema = <peewee.SchemaManager object>
classmethod add_index (*fields, **kwargs)
classmethod alias (alias=None)
classmethod bind (database, bind_refs=True, bind_backrefs=True)
classmethod bind_ctx (database, bind_refs=True, bind_backrefs=True)
classmethod bulk_create (model_list, batch_size=None)
classmethod bulk_update (model_list, fields, batch_size=None)
clone ()
coerce (_coerce=True)
static copy (method)
classmethod create (**query)
classmethod create_table (safe=True, **options)
classmethod delete ()
classmethod delete_by_id (pk)
delete_instance (recursive=False, delete_nullable=False)
dependencies (search_nullable=False)
property dirty_fields
classmethod drop_table (safe=True, drop_sequences=True, **options)
classmethod filter (*dq_nodes, **filters)
classmethod get (*query, **filters)
classmethod get_by_id (pk)
classmethod get_fields (exclude: list = None, include: list = None, sort_order: list = None) →
    set
get_id ()
classmethod get_or_create (**kwargs)
classmethod get_or_none (*query, **filters)
id = <AutoField: Base.id>
classmethod index (*fields, **kwargs)
classmethod insert (_Model__data=None, **insert)
classmethod insert_from (query, fields)
classmethod insert_many (rows, fields=None)
is_alias ()
is_dirty ()
classmethod noop ()
classmethod raw (sql, *params)
classmethod replace (_Model__data=None, **insert)
classmethod replace_many (rows, fields=None)

```

```
abstract save (*args: list, **kwargs: dict) → int
classmethod select (*fields)
classmethod set_by_id (key, value)
classmethod table_exists ()
classmethod truncate_table (**options)
unwrap ()
classmethod update (_Model__data=None, **update)
classmethod validate_model ()
```

app.models.document

Description

Classes

Document(*args, **kwargs)

app.models.document.Document

```
class app.models.document.Document (*args, **kwargs)
    Bases: app.models.base.Base
```

Attributes

Document.created_at

Document.created_by

Document.created_by_id

Document.deleted_at

Document.directory_path

Document.dirty_fields

Document.id

Document.internal_filename

Document.mime_type

Document.name

Document.size

Document.updated_at

Document.url

app.models.document.Document.created_at

```
Document.created_at = <TimestampField: Document.created_at>
```

app.models.document.Document.created_by

```
Document.created_by = <ForeignKeyField: Document.created_by>
```

app.models.document.Document.created_by_id

```
Document.created_by_id = <ForeignKeyField: Document.created_by>
```

app.models.document.Document.deleted_at

```
Document.deleted_at = <TimestampField: Document.deleted_at>
```

app.models.document.Document.directory_path

```
Document.directory_path = <CharField: Document.directory_path>
```

app.models.document.Document.dirty_fields

```
property Document.dirty_fields
```

app.models.document.Document.id

```
Document.id = <AutoField: Document.id>
```

app.models.document.Document.internal_filename

```
Document.internal_filename = <CharField: Document.internal_filename>
```

app.models.document.Document.mime_type

```
Document.mime_type = <CharField: Document.mime_type>
```

app.models.document.Document.name

```
Document.name = <CharField: Document.name>
```

app.models.document.Document.size

```
Document.size = <IntegerField: Document.size>
```

app.models.document.Document.updated_at

```
Document.updated_at = <TimestampField: Document.updated_at>
```

app.models.document.Document.url

```
property Document.url
```

Methods

<i>Document.__init__</i> (*args, **kwargs)	Initialize self.
<i>Document.add_index</i> (*fields, **kwargs)	
<i>Document.alias</i> ([alias])	
<i>Document.bind</i> (database[, bind_refs, ...])	
<i>Document.bind_ctx</i> (database[, bind_refs, ...])	
<i>Document.bulk_create</i> (model_list[, batch_size])	
<i>Document.bulk_update</i> (model_list, fields[, ...])	
<i>Document.clone</i> ()	
<i>Document.coerce</i> ([_coerce])	
<i>Document.copy</i> (method)	
<i>Document.create</i> (**query)	
<i>Document.create_table</i> ([safe])	
<i>Document.delete</i> ()	
<i>Document.delete_by_id</i> (pk)	
<i>Document.delete_instance</i> ([recursive, ...])	
<i>Document.dependencies</i> ([search_nullable])	
<i>Document.drop_table</i> ([safe, drop_sequences])	
<i>Document.filter</i> (*dq_nodes, **filters)	
<i>Document.get</i> (*query, **filters)	
<i>Document.get_by_id</i> (pk)	
<i>Document.get_fields</i> ([exclude, include, ...])	
<i>Document.get_filepath</i> ()	
<i>Document.get_id</i> ()	

continues on next page

Table 75 – continued from previous page

<i>Document.get_or_create(**kwargs)</i>
<i>Document.get_or_none(*query, **filters)</i>
<i>Document.index(*fields, **kwargs)</i>
<i>Document.insert([_Model__data])</i>
<i>Document.insert_from(query, fields)</i>
<i>Document.insert_many(rows[, fields])</i>
<i>Document.is_alias()</i>
<i>Document.is_dirty()</i>
<i>Document.noop()</i>
<i>Document.raw(sql, *params)</i>
<i>Document.replace([_Model__data])</i>
<i>Document.replace_many(rows[, fields])</i>
<i>Document.save(*args, **kwargs)</i>
<i>Document.select(*fields)</i>
<i>Document.set_by_id(key, value)</i>
<i>Document.table_exists()</i>
<i>Document.truncate_table(**options)</i>
<i>Document.unwrap()</i>
<i>Document.update([_Model__data])</i>
<i>Document.validate_model()</i>

app.models.document.Document.__init__

`Document.__init__(*args, **kwargs)`
 Initialize self. See `help(type(self))` for accurate signature.

app.models.document.Document.add_index

classmethod `Document.add_index(*fields, **kwargs)`

app.models.document.Document.alias

classmethod `Document.alias(alias=None)`

app.models.document.Document.bind

classmethod `Document.bind(database, bind_refs=True, bind_backrefs=True)`

app.models.document.Document.bind_ctx

classmethod Document.**.bind_ctx** (*database, bind_refs=True, bind_backrefs=True*)

app.models.document.Document.bulk_create

classmethod Document.**.bulk_create** (*model_list, batch_size=None*)

app.models.document.Document.bulk_update

classmethod Document.**.bulk_update** (*model_list, fields, batch_size=None*)

app.models.document.Document.clone

Document.**.clone** ()

app.models.document.Document.coerce

Document.**.coerce** (*_coerce=True*)

app.models.document.Document.copy

static Document.**.copy** (*method*)

app.models.document.Document.create

classmethod Document.**.create** (***query*)

app.models.document.Document.create_table

classmethod Document.**.create_table** (*safe=True, **options*)

app.models.document.Document.delete

classmethod Document.**.delete** ()

app.models.document.Document.delete_by_id

classmethod Document.**delete_by_id**(pk)

app.models.document.Document.delete_instance

Document.**delete_instance**(recursive=False, delete_nullable=False)

app.models.document.Document.dependencies

Document.**dependencies**(search_nullable=False)

app.models.document.Document.drop_table

classmethod Document.**drop_table**(safe=True, drop_sequences=True, **options)

app.models.document.Document.filter

classmethod Document.**filter**(*dq_nodes, **filters)

app.models.document.Document.get

classmethod Document.**get**(*query, **filters)

app.models.document.Document.get_by_id

classmethod Document.**get_by_id**(pk)

app.models.document.Document.get_fields

classmethod Document.**get_fields**(exclude: list = None, include: list = None,
sort_order: list = None) → set

app.models.document.Document.get_filepath

Document.**get_filepath**()

app.models.document.Document.get_id

`Document.get_id()`

app.models.document.Document.get_or_create

classmethod `Document.get_or_create(**kwargs)`

app.models.document.Document.get_or_none

classmethod `Document.get_or_none(*query, **filters)`

app.models.document.Document.index

classmethod `Document.index(*fields, **kwargs)`

app.models.document.Document.insert

classmethod `Document.insert(_Model__data=None, **insert)`

app.models.document.Document.insert_from

classmethod `Document.insert_from(query, fields)`

app.models.document.Document.insert_many

classmethod `Document.insert_many(rows, fields=None)`

app.models.document.Document.is_alias

`Document.is_alias()`

app.models.document.Document.is_dirty

`Document.is_dirty()`

app.models.document.Document.noop**classmethod** `Document.noop()`**app.models.document.Document.raw****classmethod** `Document.raw(sql, *params)`**app.models.document.Document.replace****classmethod** `Document.replace(_Model__data=None, **insert)`**app.models.document.Document.replace_many****classmethod** `Document.replace_many(rows, fields=None)`**app.models.document.Document.save****abstract** `Document.save(*args: list, **kwargs: dict) → int`**app.models.document.Document.select****classmethod** `Document.select(*fields)`**app.models.document.Document.set_by_id****classmethod** `Document.set_by_id(key, value)`**app.models.document.Document.table_exists****classmethod** `Document.table_exists()`**app.models.document.Document.truncate_table****classmethod** `Document.truncate_table(**options)`

app.models.document.Document.unwrap

```
Document.unwrap()
```

app.models.document.Document.update

```
classmethod Document.update(_Model__data=None, **update)
```

app.models.document.Document.validate_model

```
classmethod Document.validate_model()
```

```
class app.models.document.Document(*args, **kwargs)
```

DoesNotExist

```
alias of DocumentDoesNotExist
```

```
_coerce = True
```

```
_meta = <peewee.Metadata object>
```

```
classmethod _normalize_data(data, kwargs)
```

```
property _pk
```

```
_pk_expr()
```

```
_populate_unsaved_relations(field_dict)
```

```
_prune_fields(field_dict, only)
```

```
_schema = <peewee.SchemaManager object>
```

```
classmethod add_index(*fields, **kwargs)
```

```
classmethod alias(alias=None)
```

```
classmethod bind(database, bind_refs=True, bind_backrefs=True)
```

```
classmethod bind_ctx(database, bind_refs=True, bind_backrefs=True)
```

```
classmethod bulk_create(model_list, batch_size=None)
```

```
classmethod bulk_update(model_list, fields, batch_size=None)
```

```
clone()
```

```
coerce(_coerce=True)
```

```
static copy(method)
```

```
classmethod create(**query)
```

```
classmethod create_table(safe=True, **options)
```

```
created_at = <TimestampField: Document.created_at>
```

```
created_by = <ForeignKeyField: Document.created_by>
```

```
created_by_id = <ForeignKeyField: Document.created_by>
```

```
classmethod delete()
```

```
classmethod delete_by_id(pk)
```

```

delete_instance (recursive=False, delete_nullable=False)
deleted_at = <TimestampField: Document.deleted_at>
dependencies (search_nullable=False)
directory_path = <CharField: Document.directory_path>
property dirty_fields
classmethod drop_table (safe=True, drop_sequences=True, **options)
classmethod filter (*dq_nodes, **filters)
classmethod get (*query, **filters)
classmethod get_by_id (pk)
classmethod get_fields (exclude: list = None, include: list = None, sort_order: list = None) →
    set
get_filepath()
get_id()
classmethod get_or_create (**kwargs)
classmethod get_or_none (*query, **filters)
id = <AutoField: Document.id>
classmethod index (*fields, **kwargs)
classmethod insert (_Model__data=None, **insert)
classmethod insert_from (query, fields)
classmethod insert_many (rows, fields=None)
internal_filename = <CharField: Document.internal_filename>
is_alias()
is_dirty()
mime_type = <CharField: Document.mime_type>
name = <CharField: Document.name>
classmethod noop()
classmethod raw (sql, *params)
classmethod replace (_Model__data=None, **insert)
classmethod replace_many (rows, fields=None)
abstract save (*args: list, **kwargs: dict) → int
classmethod select (*fields)
classmethod set_by_id (key, value)
size = <IntegerField: Document.size>
classmethod table_exists()
classmethod truncate_table (**options)
unwrap()
classmethod update (_Model__data=None, **update)

```

```
updated_at = <TimestampField: Document.updated_at>
property url
classmethod validate_model()
```

app.models.role

Description

Classes

*Role(*args, **kwargs)*

app.models.role.Role

```
class app.models.role.Role(*args, **kwargs)
    Bases: app.models.base.Base, flask_security.core.RoleMixin
```

Attributes

Role.created_at

Role.deleted_at

Role.description

Role.dirty_fields

Role.id

Role.label

Role.name

Role.roles

Role.updated_at

Role.userrolethrough_set

Role.users

app.models.role.Role.created_at

```
Role.created_at = <TimestampField: Role.created_at>
```


app.models.role.Role.deleted_at

```
Role.deleted_at = <TimestampField: Role.deleted_at>
```

app.models.role.Role.description

```
Role.description = <TextField: Role.description>
```

app.models.role.Role.dirty_fields

```
property Role.dirty_fields
```

app.models.role.Role.id

```
Role.id = <AutoField: Role.id>
```

app.models.role.Role.label

```
Role.label = <CharField: Role.label>
```

app.models.role.Role.name

```
Role.name = <CharField: Role.name>
```

app.models.role.Role.roles

```
Role.roles
```

app.models.role.Role.updated_at

```
Role.updated_at = <TimestampField: Role.updated_at>
```

app.models.role.Role.userrolethrough_set

```
Role.userrolethrough_set
```

app.models.role.Role.users

`Role.users = <ManyToManyField: Role.users>`

Methods

<code>Role.__init__(*args, **kwargs)</code>	Initialize self.
<code>Role.add_index(*fields, **kwargs)</code>	
<code>Role.add_permissions(permissions)</code>	Add one or more permissions to role.
<code>Role.alias([alias])</code>	
<code>Role.bind(database[, bind_refs, bind_backrefs])</code>	
<code>Role.bind_ctx(database[, bind_refs, ...])</code>	
<code>Role.bulk_create(model_list[, batch_size])</code>	
<code>Role.bulk_update(model_list, fields[, ...])</code>	
<code>Role.clone()</code>	
<code>Role.coerce([_coerce])</code>	
<code>Role.copy(method)</code>	
<code>Role.create(**query)</code>	
<code>Role.create_table([safe])</code>	
<code>Role.delete()</code>	
<code>Role.delete_by_id(pk)</code>	
<code>Role.delete_instance([recursive, ...])</code>	
<code>Role.dependencies([search_nullable])</code>	
<code>Role.drop_table([safe, drop_sequences])</code>	
<code>Role.filter(*dq_nodes, **filters)</code>	
<code>Role.get(*query, **filters)</code>	
<code>Role.get_by_id(pk)</code>	
<code>Role.get_fields([exclude, include, sort_order])</code>	
<code>Role.get_id()</code>	
<code>Role.get_or_create(**kwargs)</code>	
<code>Role.get_or_none(*query, **filters)</code>	
<code>Role.get_permissions()</code>	Return set of permissions associated with role.
<code>Role.index(*fields, **kwargs)</code>	
<code>Role.insert([_Model__data])</code>	
<code>Role.insert_from(query, fields)</code>	
<code>Role.insert_many(rows[, fields])</code>	
<code>Role.is_alias()</code>	
<code>Role.is_dirty()</code>	
<code>Role.noop()</code>	
<code>Role.raw(sql, *params)</code>	
<code>Role.remove_permissions(permissions)</code>	Remove one or more permissions from role.
<code>Role.replace([_Model__data])</code>	
<code>Role.replace_many(rows[, fields])</code>	
<code>Role.save(*args, **kwargs)</code>	
<code>Role.select(*fields)</code>	

continues on next page

Table 78 – continued from previous page

<code>Role.set_by_id(key, value)</code>
<code>Role.table_exists()</code>
<code>Role.truncate_table(**options)</code>
<code>Role.unwrap()</code>
<code>Role.update([_Model_data])</code>
<code>Role.validate_model()</code>

app.models.role.Role.__init__

`Role.__init__(*args, **kwargs)`

Initialize self. See help(type(self)) for accurate signature.

app.models.role.Role.add_index

classmethod `Role.add_index(*fields, **kwargs)`

app.models.role.Role.add_permissions

`Role.add_permissions(permissions)`

Add one or more permissions to role.

Parameters **permissions** – a set, list, or single string.

Caller must commit to DB.

New in version 3.3.0.

Deprecated	since	version	3.4.4:	Use	UserDatastore.
<code>remove_permissions_from_role()</code>					

app.models.role.Role.alias

classmethod `Role.alias(alias=None)`

app.models.role.Role.bind

classmethod `Role.bind(database, bind_refs=True, bind_backrefs=True)`

app.models.role.Role.bind_ctx

classmethod `Role.bind_ctx(database, bind_refs=True, bind_backrefs=True)`

app.models.role.Role.bulk_create

```
classmethod Role.bulk_create(model_list, batch_size=None)
```

app.models.role.Role.bulk_update

```
classmethod Role.bulk_update(model_list, fields, batch_size=None)
```

app.models.role.Role.clone

```
Role.clone()
```

app.models.role.Role.coerce

```
Role.coerce(_coerce=True)
```

app.models.role.Role.copy

```
static Role.copy(method)
```

app.models.role.Role.create

```
classmethod Role.create(**query)
```

app.models.role.Role.create_table

```
classmethod Role.create_table(safe=True, **options)
```

app.models.role.Role.delete

```
classmethod Role.delete()
```

app.models.role.Role.delete_by_id

```
classmethod Role.delete_by_id(pk)
```

app.models.role.Role.delete_instance

`Role.delete_instance` (*recursive=False, delete_nullable=False*)

app.models.role.Role.dependencies

`Role.dependencies` (*search_nullable=False*)

app.models.role.Role.drop_table

classmethod `Role.drop_table` (*safe=True, drop_sequences=True, **options*)

app.models.role.Role.filter

classmethod `Role.filter` (**dq_nodes, **filters*)

app.models.role.Role.get

classmethod `Role.get` (**query, **filters*)

app.models.role.Role.get_by_id

classmethod `Role.get_by_id` (*pk*)

app.models.role.Role.get_fields

classmethod `Role.get_fields` (*exclude: list = None, include: list = None, sort_order: list = None*) → set

app.models.role.Role.get_id

`Role.get_id` ()

app.models.role.Role.get_or_create

classmethod `Role.get_or_create` (***kwargs*)

app.models.role.Role.get_or_none

```
classmethod Role.get_or_none(*query, **filters)
```

app.models.role.Role.get_permissions

```
Role.get_permissions()
```

Return set of permissions associated with role.

Either takes a comma separated string of permissions or an iterable of strings if permissions are in their own table.

New in version 3.3.0.

app.models.role.Role.index

```
classmethod Role.index(*fields, **kwargs)
```

app.models.role.Role.insert

```
classmethod Role.insert(_Model__data=None, **insert)
```

app.models.role.Role.insert_from

```
classmethod Role.insert_from(query, fields)
```

app.models.role.Role.insert_many

```
classmethod Role.insert_many(rows, fields=None)
```

app.models.role.Role.is_alias

```
Role.is_alias()
```

app.models.role.Role.is_dirty

```
Role.is_dirty()
```

app.models.role.Role.noop

classmethod `Role.noop()`

app.models.role.Role.raw

classmethod `Role.raw(sql, *params)`

app.models.role.Role.remove_permissions

`Role.remove_permissions(permissions)`

Remove one or more permissions from role.

Parameters **permissions** – a set, list, or single string.

Caller must commit to DB.

New in version 3.3.0.

Deprecated	since	version	3.4.4:	Use	<code>UserDatastore.remove_permissions_from_role()</code>
------------	-------	---------	--------	-----	---

app.models.role.Role.replace

classmethod `Role.replace(_Model__data=None, **insert)`

app.models.role.Role.replace_many

classmethod `Role.replace_many(rows, fields=None)`

app.models.role.Role.save

abstract `Role.save(*args: list, **kwargs: dict) → int`

app.models.role.Role.select

classmethod `Role.select(*fields)`

app.models.role.Role.set_by_id

classmethod `Role.set_by_id(key, value)`

app.models.role.Role.table_exists

```
classmethod Role.table_exists()
```

app.models.role.Role.truncate_table

```
classmethod Role.truncate_table(**options)
```

app.models.role.Role.unwrap

```
Role.unwrap()
```

app.models.role.Role.update

```
classmethod Role.update(_Model__data=None, **update)
```

app.models.role.Role.validate_model

```
classmethod Role.validate_model()
```

```
class app.models.role.Role(*args, **kwargs)
```

DoesNotExist

alias of RoleDoesNotExist

_coerce = True

_meta = <peewee.Metadata object>

```
classmethod _normalize_data(data, kwargs)
```

property _pk

```
_pk_expr()
```

```
_populate_unsaved_relations(field_dict)
```

```
_prune_fields(field_dict, only)
```

_schema = <peewee.SchemaManager object>

```
classmethod add_index(*fields, **kwargs)
```

add_permissions(permissions)

Add one or more permissions to role.

Parameters permissions – a set, list, or single string.

Caller must commit to DB.

New in version 3.3.0.

Deprecated since version 3.4.4: Use `UserDatastore.remove_permissions_from_role()`

```
classmethod alias(alias=None)
```

```
classmethod bind(database, bind_refs=True, bind_backrefs=True)
```



```

classmethod bind_ctx (database, bind_refs=True, bind_backrefs=True)
classmethod bulk_create (model_list, batch_size=None)
classmethod bulk_update (model_list, fields, batch_size=None)
clone ()
coerce (_coerce=True)
static copy (method)
classmethod create (**query)
classmethod create_table (safe=True, **options)
created_at = <TimestampField: Role.created_at>
classmethod delete ()
classmethod delete_by_id (pk)
delete_instance (recursive=False, delete_nullable=False)
deleted_at = <TimestampField: Role.deleted_at>
dependencies (search_nullable=False)
description = <TextField: Role.description>
property dirty_fields
classmethod drop_table (safe=True, drop_sequences=True, **options)
classmethod filter (*dq_nodes, **filters)
classmethod get (*query, **filters)
classmethod get_by_id (pk)
classmethod get_fields (exclude: list = None, include: list = None, sort_order: list = None) →
    set
get_id ()
classmethod get_or_create (**kwargs)
classmethod get_or_none (*query, **filters)
get_permissions ()
    Return set of permissions associated with role.

    Either takes a comma separated string of permissions or an iterable of strings if permissions are in their
    own table.

    New in version 3.3.0.
id = <AutoField: Role.id>
classmethod index (*fields, **kwargs)
classmethod insert (_Model__data=None, **insert)
classmethod insert_from (query, fields)
classmethod insert_many (rows, fields=None)
is_alias ()
is_dirty ()

```

```
label = <CharField: Role.label>
name = <CharField: Role.name>
classmethod noop()
classmethod raw(sql, *params)
remove_permissions(permissions)
    Remove one or more permissions from role.

    Parameters permissions – a set, list, or single string.

    Caller must commit to DB.

    New in version 3.3.0.

    Deprecated since version 3.4.4: Use UserDatastore.remove_permissions_from_role()
classmethod replace(_Model__data=None, **insert)
classmethod replace_many(rows, fields=None)
roles
abstract save(*args: list, **kwargs: dict) → int
classmethod select(*fields)
classmethod set_by_id(key, value)
classmethod table_exists()
classmethod truncate_table(**options)
unwrap()
classmethod update(_Model__data=None, **update)
updated_at = <TimestampField: Role.updated_at>
userrolethrough_set
users = <ManyToManyField: Role.users>
classmethod validate_model()
```

app.models.user

Description

Classes

`User(*args, **kwargs)`

app.models.user.User

```
class app.models.user.User(*args, **kwargs)
    Bases: app.models.base.Base, flask_security.core.UserMixin
```

Attributes

<i>User.active</i>	
<i>User.birth_date</i>	
<i>User.children</i>	
<i>User.created_at</i>	
<i>User.created_by</i>	
<i>User.created_by_id</i>	
<i>User.deleted_at</i>	
<i>User.dirty_fields</i>	
<i>User.document_set</i>	
<i>User.email</i>	
<i>User.genre</i>	
<i>User.id</i>	
<i>User.is_active</i>	Returns <i>True</i> if the user is active.
<i>User.is_anonymous</i>	
<i>User.is_authenticated</i>	
<i>User.last_name</i>	
<i>User.name</i>	
<i>User.password</i>	
<i>User.roles</i>	
<i>User.updated_at</i>	
<i>User.userrolethrough_set</i>	

app.models.user.User.active

```
User.active = <BooleanField: User.active>
```

app.models.user.User.birth_date

```
User.birth_date = <DateField: User.birth_date>
```

app.models.user.User.children

```
User.children
```

app.models.user.User.created_at

```
User.created_at = <TimestampField: User.created_at>
```

app.models.user.User.created_by

```
User.created_by = <ForeignKeyField: User.created_by>
```

app.models.user.User.created_by_id

```
User.created_by_id = <ForeignKeyField: User.created_by>
```

app.models.user.User.deleted_at

```
User.deleted_at = <TimestampField: User.deleted_at>
```

app.models.user.User.dirty_fields

```
property User.dirty_fields
```

app.models.user.User.document_set

```
User.document_set
```

app.models.user.User.email

```
User.email = <CharField: User.email>
```

app.models.user.User.genre

```
User.genre = <FixedCharField: User.genre>
```

app.models.user.User.id

```
User.id = <AutoField: User.id>
```

app.models.user.User.is_active

property User.is_active
Returns *True* if the user is active.

app.models.user.User.is_anonymous

property User.is_anonymous

app.models.user.User.is_authenticated

property User.is_authenticated

app.models.user.User.last_name

User.last_name = <CharField: User.last_name>

app.models.user.User.name

User.name = <CharField: User.name>

app.models.user.User.password

User.password = <CharField: User.password>

app.models.user.User.roles

User.roles = <ManyToManyField: User.roles>

app.models.user.User.updated_at

User.updated_at = <TimestampField: User.updated_at>

app.models.user.User.userrolethrough_set

User.userrolethrough_set

Methods

<code>User.__init__(*args, **kwargs)</code>	Initialize self.
<code>User.add_index(*fields, **kwargs)</code>	
<code>User.alias([alias])</code>	
<code>User.bind(database[, bind_refs, bind_backrefs])</code>	
<code>User.bind_ctx(database[, bind_refs, ...])</code>	
<code>User.bulk_create(model_list[, batch_size])</code>	
<code>User.bulk_update(model_list, fields[, ...])</code>	
<code>User.calc_username()</code>	Come up with the best 'username' based on how the app is configured (via SECURITY_USER_IDENTITY_ATTRIBUTES).
<code>User.clone()</code>	
<code>User.coerce([_coerce])</code>	
<code>User.copy(method)</code>	
<code>User.create(**query)</code>	
<code>User.create_table([safe])</code>	
<code>User.delete()</code>	
<code>User.delete_by_id(pk)</code>	
<code>User.delete_instance([recursive, ...])</code>	
<code>User.dependencies([search_nullable])</code>	
<code>User.drop_table([safe, drop_sequences])</code>	
<code>User.ensure_password(plain_text)</code>	
<code>User.filter(*dq_nodes, **filters)</code>	
<code>User.get(*query, **filters)</code>	
<code>User.get_auth_token()</code>	Constructs the user's authentication token.
<code>User.get_by_id(pk)</code>	
<code>User.get_fields([exclude, include, sort_order])</code>	
<code>User.get_id()</code>	Returns the user identification attribute.
<code>User.get_or_create(**kwargs)</code>	
<code>User.get_or_none(*query, **filters)</code>	
<code>User.get_redirect_qparams([existing])</code>	Return user info that will be added to redirect query params.
<code>User.get_reset_token()</code>	
<code>User.get_security_payload()</code>	Serialize user object as response payload.
<code>User.has_permission(permission)</code>	Returns <i>True</i> if user has this permission (via a role it has).
<code>User.has_role(role)</code>	Returns <i>True</i> if the user identifies with the specified role.
<code>User.index(*fields, **kwargs)</code>	
<code>User.insert([_Model__data])</code>	
<code>User.insert_from(query, fields)</code>	
<code>User.insert_many(rows[, fields])</code>	
<code>User.is_alias()</code>	
<code>User.is_dirty()</code>	
<code>User.noop()</code>	

continues on next page

Table 81 – continued from previous page

<code>User.raw(sql, *params)</code>	
<code>User.replace([_Model__data])</code>	
<code>User.replace_many(rows[, fields])</code>	
<code>User.save(*args, **kwargs)</code>	
<code>User.select(*fields)</code>	
<code>User.set_by_id(key, value)</code>	
<code>User.table_exists()</code>	
<code>User.tf_send_security_token(method, **kwargs)</code>	Generate and send the security code for two-factor.
<code>User.truncate_table(**options)</code>	
<code>User.unwrap()</code>	
<code>User.update([_Model__data])</code>	
<code>User.us_send_security_token(method, **kwargs)</code>	Generate and send the security code for unified sign in.
<code>User.validate_model()</code>	
<code>User.verify_and_update_password(password)</code>	Returns True if the password is valid for the specified user.
<code>User.verify_auth_token(data)</code>	Perform additional verification of contents of auth token.
<code>User.verify_reset_token(token)</code>	

app.models.user.User.__init__

`User.__init__(*args, **kwargs)`

Initialize self. See help(type(self)) for accurate signature.

app.models.user.User.add_index

classmethod `User.add_index(*fields, **kwargs)`

app.models.user.User.alias

classmethod `User.alias(alias=None)`

app.models.user.User.bind

classmethod `User.bind(database, bind_refs=True, bind_backrefs=True)`

app.models.user.User.bind_ctx

classmethod User.**bind_ctx**(*database, bind_refs=True, bind_backrefs=True*)

app.models.user.User.bulk_create

classmethod User.**bulk_create**(*model_list, batch_size=None*)

app.models.user.User.bulk_update

classmethod User.**bulk_update**(*model_list, fields, batch_size=None*)

app.models.user.User.calc_username

User.**calc_username**()

Come up with the best 'username' based on how the app is configured (via SECURITY_USER_IDENTITY_ATTRIBUTES). Returns the first non-null match (and converts to string). In theory this should NEVER be the empty string unless the user record isn't actually valid.

New in version 3.4.0.

app.models.user.User.clone

User.**clone**()

app.models.user.User.coerce

User.**coerce**(*_coerce=True*)

app.models.user.User.copy

static User.**copy**(*method*)

app.models.user.User.create

classmethod User.**create**(***query*)

app.models.user.User.create_table

```
classmethod User.create_table (safe=True, **options)
```

app.models.user.User.delete

```
classmethod User.delete ()
```

app.models.user.User.delete_by_id

```
classmethod User.delete_by_id (pk)
```

app.models.user.User.delete_instance

```
User.delete_instance (recursive=False, delete_nullable=False)
```

app.models.user.User.dependencies

```
User.dependencies (search_nullable=False)
```

app.models.user.User.drop_table

```
classmethod User.drop_table (safe=True, drop_sequences=True, **options)
```

app.models.user.User.ensure_password

```
static User.ensure_password (plain_text: str) → str
```

app.models.user.User.filter

```
classmethod User.filter (*dq_nodes, **filters)
```

app.models.user.User.get

```
classmethod User.get (*query, **filters)
```

app.models.user.User.get_auth_token

`User.get_auth_token()`

Constructs the user's authentication token.

This data **MUST** be securely signed using the `remember_token_serializer`

app.models.user.User.get_by_id

classmethod `User.get_by_id(pk)`

app.models.user.User.get_fields

classmethod `User.get_fields(exclude: list = None, include: list = None, sort_order: list = None) → set`

app.models.user.User.get_id

`User.get_id()`

Returns the user identification attribute.

This will be *fs_uniquifier* if that is available, else base class id (which is via Flask-Login and is `user.id`).

New in version 3.4.0.

app.models.user.User.get_or_create

classmethod `User.get_or_create(**kwargs)`

app.models.user.User.get_or_none

classmethod `User.get_or_none(*query, **filters)`

app.models.user.User.get_redirect_qparams

`User.get_redirect_qparams(existing=None)`

Return user info that will be added to redirect query params.

Parameters **existing** – A dict that will be updated.

Returns A dict whose keys will be query params and values will be query values.

New in version 3.2.0.

app.models.user.User.get_reset_token

`User.get_reset_token()` → str

app.models.user.User.get_security_payload

`User.get_security_payload()`
Serialize user object as response payload.

app.models.user.User.has_permission

`User.has_permission(permission)`
Returns *True* if user has this permission (via a role it has).
Parameters **permission** – permission string name
New in version 3.3.0.

app.models.user.User.has_role

`User.has_role(role)`
Returns *True* if the user identifies with the specified role.
Parameters **role** – A role name or *Role* instance

app.models.user.User.index

classmethod `User.index(*fields, **kwargs)`

app.models.user.User.insert

classmethod `User.insert(_Model__data=None, **insert)`

app.models.user.User.insert_from

classmethod `User.insert_from(query, fields)`

app.models.user.User.insert_many

classmethod `User.insert_many(rows, fields=None)`

app.models.user.User.is_alias

`User.is_alias()`

app.models.user.User.is_dirty

`User.is_dirty()`

app.models.user.User.noop

`classmethod User.noop()`

app.models.user.User.raw

`classmethod User.raw(sql, *params)`

app.models.user.User.replace

`classmethod User.replace(_Model__data=None, **insert)`

app.models.user.User.replace_many

`classmethod User.replace_many(rows, fields=None)`

app.models.user.User.save

`User.save(*args: list, **kwargs: dict) → int`

app.models.user.User.select

`classmethod User.select(*fields)`

app.models.user.User.set_by_id

`classmethod User.set_by_id(key, value)`

app.models.user.User.table_exists

```
classmethod User.table_exists()
```

app.models.user.User.tf_send_security_token

```
User.tf_send_security_token(method, **kwargs)
```

Generate and send the security code for two-factor.

Parameters

- **method** – The method in which the code will be sent
- **kwargs** – Opaque parameters that are subject to change at any time

Returns None if successful, error message if not.

This is a wrapper around `tf_send_security_token()` that can be overridden to manage any errors.

New in version 3.4.0.

app.models.user.User.truncate_table

```
classmethod User.truncate_table(**options)
```

app.models.user.User.unwrap

```
User.unwrap()
```

app.models.user.User.update

```
classmethod User.update(_Model__data=None, **update)
```

app.models.user.User.us_send_security_token

```
User.us_send_security_token(method, **kwargs)
```

Generate and send the security code for unified sign in.

Parameters

- **method** – The method in which the code will be sent
- **kwargs** – Opaque parameters that are subject to change at any time

Returns None if successful, error message if not.

This is a wrapper around `us_send_security_token()` that can be overridden to manage any errors.

New in version 3.4.0.

app.models.user.User.validate_model

```
classmethod User.validate_model()
```

app.models.user.User.verify_and_update_password

```
User.verify_and_update_password(password)
```

Returns True if the password is valid for the specified user.

Additionally, the hashed password in the database is updated if the hashing algorithm happens to have changed.

N.B. you MUST call DB commit if you are using a session-based datastore (such as SQLAlchemy) since the user instance might have been altered (i.e. `app.security.datastore.commit()`). This is usually handled in the view.

Parameters password – A plaintext password to verify
New in version 3.2.0.

app.models.user.User.verify_auth_token

```
User.verify_auth_token(data)
```

Perform additional verification of contents of auth token. Prior to this being called the token has been validated (via signing) and has not expired.

Parameters data – the data as formulated by `get_auth_token()`
New in version 3.3.0.

app.models.user.User.verify_reset_token

```
static User.verify_reset_token(token: str) → any
```

```
class app.models.user.User(*args, **kwargs)
```

```
DoesNotExist
```

```
alias of UserDoesNotExist
```

```
_coerce = True
```

```
_meta = <peewee.Metadata object>
```

```
classmethod _normalize_data(data, kwargs)
```

```
property _pk
```

```
_pk_expr()
```

```
_populate_unsaved_relations(field_dict)
```

```
_prune_fields(field_dict, only)
```

```
_schema = <peewee.SchemaManager object>
```

```
active = <BooleanField: User.active>
```

```
classmethod add_index(*fields, **kwargs)
```

```
classmethod alias(alias=None)
```

```
classmethod bind(database, bind_refs=True, bind_backrefs=True)
```

```

classmethod bind_ctx (database, bind_refs=True, bind_backrefs=True)
birth_date = <DateField: User.birth_date>
classmethod bulk_create (model_list, batch_size=None)
classmethod bulk_update (model_list, fields, batch_size=None)
calc_username ()
    Come up with the best 'username' based on how the app is configured (via
    SECURITY_USER_IDENTITY_ATTRIBUTES). Returns the first non-null match (and converts to
    string). In theory this should NEVER be the empty string unless the user record isn't actually valid.

    New in version 3.4.0.
children
clone ()
coerce (_coerce=True)
static copy (method)
classmethod create (**query)
classmethod create_table (safe=True, **options)
created_at = <TimestampField: User.created_at>
created_by = <ForeignKeyField: User.created_by>
created_by_id = <ForeignKeyField: User.created_by>
classmethod delete ()
classmethod delete_by_id (pk)
delete_instance (recursive=False, delete_nullable=False)
deleted_at = <TimestampField: User.deleted_at>
dependencies (search_nullable=False)
property dirty_fields
document_set
classmethod drop_table (safe=True, drop_sequences=True, **options)
email = <CharField: User.email>
static ensure_password (plain_text: str) → str
classmethod filter (*dq_nodes, **filters)
genre = <FixedCharField: User.genre>
classmethod get (*query, **filters)
get_auth_token ()
    Constructs the user's authentication token.

    This data MUST be securely signed using the remember_token_serializer
classmethod get_by_id (pk)
classmethod get_fields (exclude: list = None, include: list = None, sort_order: list = None) →
    set

```

get_id()

Returns the user identification attribute.

This will be *fs_uniquifier* if that is available, else base class id (which is via Flask-Login and is *user.id*).

New in version 3.4.0.

classmethod get_or_create (**kwargs)

classmethod get_or_none (*query, **filters)

get_redirect_qparams (existing=None)

Return user info that will be added to redirect query params.

Parameters existing – A dict that will be updated.

Returns A dict whose keys will be query params and values will be query values.

New in version 3.2.0.

get_reset_token() → str

get_security_payload()

Serialize user object as response payload.

has_permission (permission)

Returns *True* if user has this permission (via a role it has).

Parameters permission – permission string name

New in version 3.3.0.

has_role (role)

Returns *True* if the user identifies with the specified role.

Parameters role – A role name or *Role* instance

id = <AutoField: User.id>

classmethod index (*fields, **kwargs)

classmethod insert (_Model__data=None, **insert)

classmethod insert_from (query, fields)

classmethod insert_many (rows, fields=None)

property is_active

Returns *True* if the user is active.

is_alias()

property is_anonymous

property is_authenticated

is_dirty()

last_name = <CharField: User.last_name>

name = <CharField: User.name>

classmethod noop()

password = <CharField: User.password>

classmethod raw (sql, *params)

classmethod replace (_Model__data=None, **insert)


```

classmethod replace_many (rows, fields=None)
roles = <ManyToManyField: User.roles>
save (*args: list, **kwargs: dict) → int
classmethod select (*fields)
classmethod set_by_id (key, value)
classmethod table_exists ()
tf_send_security_token (method, **kwargs)
    Generate and send the security code for two-factor.

```

Parameters

- **method** – The method in which the code will be sent
- **kwargs** – Opaque parameters that are subject to change at any time

Returns None if successful, error message if not.

This is a wrapper around `tf_send_security_token()` that can be overridden to manage any errors.

New in version 3.4.0.

```

classmethod truncate_table (**options)
unwrap ()
classmethod update (_Model__data=None, **update)
updated_at = <TimestampField: User.updated_at>
us_send_security_token (method, **kwargs)
    Generate and send the security code for unified sign in.

```

Parameters

- **method** – The method in which the code will be sent
- **kwargs** – Opaque parameters that are subject to change at any time

Returns None if successful, error message if not.

This is a wrapper around `us_send_security_token()` that can be overridden to manage any errors.

New in version 3.4.0.

```

userrolethrough_set
classmethod validate_model ()
verify_and_update_password (password)

```

Returns True if the password is valid for the specified user.

Additionally, the hashed password in the database is updated if the hashing algorithm happens to have changed.

N.B. you MUST call DB commit if you are using a session-based datastore (such as SQLAlchemy) since the user instance might have been altered (i.e. `app.security.datastore.commit()`). This is usually handled in the view.

Parameters **password** – A plaintext password to verify

New in version 3.2.0.

verify_auth_token (*data*)

Perform additional verification of contents of auth token. Prior to this being called the token has been validated (via signing) and has not expired.

Parameters **data** – the data as formulated by *get_auth_token()*

New in version 3.3.0.

static verify_reset_token (*token: str*) → any

Functions

get_models()

app.models.get_models

`app.models.get_models()` → list

`app.models.get_models()` → list

2.1.6 app.utils

Description

Collection of functions and classes which make common patterns shorter and easier.

Modules

app.utils.decorators

app.utils.file_storage

app.utils.libreoffice

app.utils.marshmallow_schema

app.utils.swagger_models

app.utils.decorators

Description

Functions

token_required(fnc)

app.utils.decorators.token_required

`app.utils.decorators.token_required(fnc)`

`app.utils.decorators.token_required(fnc)`

app.utils.file_storage**Description****Classes**

FileStorage()

app.utils.file_storage.FileStorage

class `app.utils.file_storage.FileStorage`

Bases: `object`

Methods

<i>FileStorage.__init__()</i>	Initialize self.
-------------------------------	------------------

<i>FileStorage.copy_file(src, dst)</i>
--

<i>FileStorage.get_basename(filename[, ...])</i>
--

<i>FileStorage.get_filesize(filename)</i>

<i>FileStorage.rename(src, dst)</i>

<i>FileStorage.save_bytes(file_content, filename)</i>

app.utils.file_storage.FileStorage.__init__

`FileStorage.__init__()`

Initialize self. See `help(type(self))` for accurate signature.

app.utils.file_storage.FileStorage.copy_file

```
static FileStorage.copy_file (src: str, dst: str) → None
```

app.utils.file_storage.FileStorage.get_basename

```
static FileStorage.get_basename (filename: str, include_path: bool = False) →  
    str
```

app.utils.file_storage.FileStorage.get_filesize

```
static FileStorage.get_filesize (filename: str) → int
```

app.utils.file_storage.FileStorage.rename

```
static FileStorage.rename (src: str, dst: str) → None
```

app.utils.file_storage.FileStorage.save_bytes

```
FileStorage.save_bytes (file_content: bytes, filename: str, override: bool = False)
```

```
class app.utils.file_storage.FileStorage
```

```
    static copy_file (src: str, dst: str) → None
```

```
    static get_basename (filename: str, include_path: bool = False) → str
```

```
    static get_filesize (filename: str) → int
```

```
    static rename (src: str, dst: str) → None
```

```
    save_bytes (file_content: bytes, filename: str, override: bool = False)
```

app.utils.libreoffice**Description****Functions**

```
convert_to(folder, source)
```

```
libreoffice_exec()
```

app.utils.libreoffice.convert_to

`app.utils.libreoffice.convert_to(folder: str, source: str) → str`

app.utils.libreoffice.libreoffice_exec

`app.utils.libreoffice.libreoffice_exec() → str`

Exceptions

LibreOfficeError(output)

app.utils.libreoffice.LibreOfficeError

exception `app.utils.libreoffice.LibreOfficeError(output)`

exception `app.utils.libreoffice.LibreOfficeError(output)`

args**with_traceback()**

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

`app.utils.libreoffice.convert_to(folder: str, source: str) → str`

`app.utils.libreoffice.libreoffice_exec() → str`

app.utils.marshmallow_schema**Description****Classes**

DocumentSchema(*, only, Set[str] = None, ...)

ExportWordInputSchema(*, only, ...)

GetDocumentDataInputSchema(*, only, ...)

RoleSchema(*, only, Set[str] = None, ...)

SearchSchema(*, only, Set[str] = None, ...)

Timestamp(*, default, missing, data_key, ...)

Field that serializes to timestamp integer and deserializes to a datetime.datetime class.

UserSchema(*, only, Set[str] = None, ...)

app.utils.marshmallow_schema.DocumentSchema

```
class app.utils.marshmallow_schema.DocumentSchema(*, only: Union[Sequence[str],  
Set[str]] = None, exclude:  
Union[Sequence[str], Set[str]]  
= (), many: bool = False, con-  
text: Dict = None, load_only:  
Union[Sequence[str], Set[str]] = (),  
dump_only: Union[Sequence[str],  
Set[str]] = (), partial: Union[bool,  
Sequence[str], Set[str]] = False,  
unknown: str = None)
```

Bases: flask_marshmallow.schema.Schema

Attributes

DocumentSchema.TYPE_MAPPING

DocumentSchema.dict_class

DocumentSchema.error_messages

DocumentSchema.opts

DocumentSchema.set_class

app.utils.marshmallow_schema.DocumentSchema.TYPE_MAPPING

DocumentSchema.**TYPE_MAPPING** = {<class 'str'>: <class 'marshmallow.fields.String'>,

app.utils.marshmallow_schema.DocumentSchema.dict_class

property DocumentSchema.**dict_class**

app.utils.marshmallow_schema.DocumentSchema.error_messages

DocumentSchema.**error_messages** = {}

app.utils.marshmallow_schema.DocumentSchema.opts

DocumentSchema.**opts** = <marshmallow.schema.SchemaOpts object>

app.utils.marshmallow_schema.DocumentSchema.set_class**property** DocumentSchema.set_class**Methods**

<i>DocumentSchema.__init__</i> (*[, only, exclude, ...])	Initialize self.
<i>DocumentSchema.dump</i> (obj, *[, many])	Serialize an object to native Python data types according to this Schema's fields.
<i>DocumentSchema.dumps</i> (obj, *args[, many])	Same as <i>dump()</i> , except return a JSON-encoded string.
<i>DocumentSchema.from_dict</i> (fields, *[, name])	Generate a <i>Schema</i> class given a dictionary of fields.
<i>DocumentSchema.get_attribute</i> (obj, attr, default)	Defines how to pull values from an object to serialize.
<i>DocumentSchema.handle_error</i> (error, data, *, ...)	Custom error handler function for the schema.
<i>DocumentSchema.jsonify</i> (obj[, many])	Return a JSON response containing the serialized data.
<i>DocumentSchema.load</i> (data, *[, many, ...])	Deserialize a data structure to an object defined by this Schema's fields.
<i>DocumentSchema.loads</i> (json_data, *[, many, ...])	Same as <i>load()</i> , except it takes a JSON string as input.
<i>DocumentSchema.make_url</i> (data, **kwargs)	
<i>DocumentSchema.on_bind_field</i> (field_name, ...)	Hook to modify a field when it is bound to the <i>Schema</i> .
<i>DocumentSchema.valid_request_file</i> (data)	
<i>DocumentSchema.validate</i> (data, *[, many, partial])	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

app.utils.marshmallow_schema.DocumentSchema.__init__

DocumentSchema.__init__(*, only: Union[Sequence[str], Set[str]] = None, exclude: Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str], Set[str]] = False, unknown: str = None)

Initialize self. See help(type(self)) for accurate signature.

app.utils.marshmallow_schema.DocumentSchema.dump

DocumentSchema.**dump** (*obj*: Any, *, *many*: bool = None)

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.DocumentSchema.dumps

DocumentSchema.**dumps** (*obj*: Any, **args*, *many*: bool = None, ***kwargs*)

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

app.utils.marshmallow_schema.DocumentSchema.from_dict

classmethod DocumentSchema.**from_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], *, *name*: str = 'Generated-Schema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

app.utils.marshmallow_schema.DocumentSchema.get_attribute

`DocumentSchema.get_attribute` (*obj: Any, attr: str, default: Any*)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

app.utils.marshmallow_schema.DocumentSchema.handle_error

`DocumentSchema.handle_error` (*error: marshmallow.exceptions.ValidationError, data: Any, *, many: bool, **kwargs*)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.DocumentSchema.jsonify

`DocumentSchema.jsonify` (*obj, many=<object object>, *args, **kwargs*)

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to *False*, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.DocumentSchema.load

`DocumentSchema.load` (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None*)

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing

fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (data, errors) tuple. A `ValidationError` is raised if invalid data are passed.

app.utils.marshmallow_schema.DocumentSchema.loads

`DocumentSchema.loads` (*json_data*: str, *, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, **kwargs)

Same as `load()`, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (data, errors) tuple. A `ValidationError` is raised if invalid data are passed.

app.utils.marshmallow_schema.DocumentSchema.make_url

`DocumentSchema.make_url` (*data*, **kwargs)

app.utils.marshmallow_schema.DocumentSchema.on_bind_field

`DocumentSchema.on_bind_field` (*field_name*: str, *field_obj*: `marshmallow.fields.Field`) → None

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

app.utils.marshmallow_schema.DocumentSchema.valid_request_file

```
static DocumentSchema.valid_request_file(data)
```

app.utils.marshmallow_schema.DocumentSchema.validate

```
DocumentSchema.validate(data: Mapping, *, many: bool = None, partial:  
                        Union[bool, Sequence[str], Set[str]] = None) → Dict[str,  
                        List[str]]
```

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

app.utils.marshmallow_schema.ExportWordInputSchema

```
class app.utils.marshmallow_schema.ExportWordInputSchema(*, only:  
                                                            Union[Sequence[str],  
                                                            Set[str]] =  
                                                            None, exclude:  
                                                            Union[Sequence[str],  
                                                            Set[str]] = (), many:  
                                                            bool = False, context:  
                                                            Dict = None, load_only:  
                                                            Union[Sequence[str],  
                                                            Set[str]] = (),  
                                                            dump_only:  
                                                            Union[Sequence[str],  
                                                            Set[str]] = (), partial:  
                                                            Union[bool, Sequence[str], Set[str]] =  
                                                            False, unknown: str =  
                                                            None)
```

Bases: flask_marshmallow.schema.Schema

Attributes

<code>ExportWordInputSchema.</code>
<code>TYPE_MAPPING</code>
<code>ExportWordInputSchema.</code>
<code>dict_class</code>
<code>ExportWordInputSchema.</code>
<code>error_messages</code>
<code>ExportWordInputSchema.opts</code>
<code>ExportWordInputSchema.set_class</code>

`app.utils.marshmallow_schema.ExportWordInputSchema.TYPE_MAPPING`

`ExportWordInputSchema.TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.St`

`app.utils.marshmallow_schema.ExportWordInputSchema.dict_class`

property `ExportWordInputSchema.dict_class`

`app.utils.marshmallow_schema.ExportWordInputSchema.error_messages`

`ExportWordInputSchema.error_messages = {}`

`app.utils.marshmallow_schema.ExportWordInputSchema.opts`

`ExportWordInputSchema.opts = <marshmallow.schema.SchemaOpts object>`

`app.utils.marshmallow_schema.ExportWordInputSchema.set_class`

property `ExportWordInputSchema.set_class`

Methods

<code>ExportWordInputSchema.</code>	Initialize self.
<code>__init__(*[, only, ...])</code>	
<code>ExportWordInputSchema.</code>	
<code>convert_to_integer(...)</code>	
<code>ExportWordInputSchema.dump(obj, *[, many])</code>	Serialize an object to native Python data types according to this Schema's fields.
<code>ExportWordInputSchema.dumps(obj, *args[, many])</code>	Same as <code>dump()</code> , except return a JSON-encoded string.
<code>ExportWordInputSchema.from_dict(fields, *[, ...])</code>	Generate a <i>Schema</i> class given a dictionary of fields.
<code>ExportWordInputSchema.get_attribute(obj, ...)</code>	Defines how to pull values from an object to serialize.

continues on next page

Table 93 – continued from previous page

<code>ExportWordInputSchema.handle_error(error, ...)</code>	Custom error handler function for the schema.
<code>ExportWordInputSchema jsonify(obj[, many])</code>	Return a JSON response containing the serialized data.
<code>ExportWordInputSchema.load(data, *[, many, ...])</code>	Deserialize a data structure to an object defined by this Schema's fields.
<code>ExportWordInputSchema.loads(json_data, *[, ...])</code>	Same as <code>load()</code> , except it takes a JSON string as input.
<code>ExportWordInputSchema.on_bind_field(...)</code>	Hook to modify a field when it is bound to the <i>Schema</i> .
<code>ExportWordInputSchema.validate(data, *[, ...])</code>	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

app.utils.marshmallow_schema.ExportWordInputSchema.__init__

```
ExportWordInputSchema.__init__ (*, only: Union[Sequence[str], Set[str]] = None,
                                exclude: Union[Sequence[str], Set[str]] = (),
                                many: bool = False, context: Dict = None,
                                load_only: Union[Sequence[str], Set[str]] = (),
                                dump_only: Union[Sequence[str], Set[str]] = (),
                                partial: Union[bool, Sequence[str], Set[str]] =
                                False, unknown: str = None)
```

Initialize self. See `help(type(self))` for accurate signature.

app.utils.marshmallow_schema.ExportWordInputSchema.convert_to_integer

```
ExportWordInputSchema.convert_to_integer (value, many, **kwargs)
```

app.utils.marshmallow_schema.ExportWordInputSchema.dump

```
ExportWordInputSchema.dump (obj: Any, *, many: bool = None)
```

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.ExportWordInputSchema.dumps

`ExportWordInputSchema.dumps` (*obj*: Any, **args*, *many*: bool = None, ***kwargs*)

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (*data*, *errors*) tuple. A `ValidationError` is raised if *obj* is invalid.

app.utils.marshmallow_schema.ExportWordInputSchema.from_dict

classmethod `ExportWordInputSchema.from_dict` (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], *, *name*: str = 'GeneratedSchema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

app.utils.marshmallow_schema.ExportWordInputSchema.get_attribute

`ExportWordInputSchema.get_attribute` (*obj*: Any, *attr*: str, *default*: Any)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

app.utils.marshmallow_schema.ExportWordInputSchema.handle_error

`ExportWordInputSchema.handle_error` (*error*: *marshmallow.exceptions.ValidationError*, *data*: *Any*, *, *many*: *bool*, ***kwargs*)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of *many* on dump or load.
- **partial** – Value of *partial* on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.ExportWordInputSchema.jsonify

`ExportWordInputSchema.jsonify` (*obj*, *many*=<object object>, **args*, ***kwargs*)

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to *False*, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.ExportWordInputSchema.load

`ExportWordInputSchema.load` (*data*: *Union[Mapping[str, Any], Iterable[Mapping[str, Any]]]*, *, *many*: *bool* = *None*, *partial*: *Union[bool, Sequence[str], Set[str]]* = *None*, *unknown*: *str* = *None*)

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns

Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

`app.utils.marshmallow_schema.ExportWordInputSchema.loads`

`ExportWordInputSchema.loads` (*json_data: str, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, **kwargs*)

Same as `load()`, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use `EXCLUDE`, `INCLUDE` or `RAISE`. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

`app.utils.marshmallow_schema.ExportWordInputSchema.on_bind_field`

`ExportWordInputSchema.on_bind_field` (*field_name: str, field_obj: marshmallow.fields.Field*) \rightarrow *None*

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

`app.utils.marshmallow_schema.ExportWordInputSchema.validate`

`ExportWordInputSchema.validate` (*data: Mapping, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None*) \rightarrow *Dict[str, List[str]]*

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

app.utils.marshmallow_schema.GetDocumentDataInputSchema

```

class app.utils.marshmallow_schema.GetDocumentDataInputSchema (*,
                                                                only:
                                                                Union[Sequence[str],
                                                                Set[str]] =
                                                                None, exclude:
                                                                Union[Sequence[str],
                                                                Set[str]] = (),
                                                                many: bool
                                                                = False, con-
                                                                text: Dict =
                                                                None, load_only:
                                                                Union[Sequence[str],
                                                                Set[str]] =
                                                                (), dump_only:
                                                                Union[Sequence[str],
                                                                Set[str]] =
                                                                (), partial:
                                                                Union[bool,
                                                                Sequence[str],
                                                                Set[str]] = False,
                                                                unknown: str =
                                                                None)

```

Bases: flask_marshmallow.schema.Schema

Attributes

GetDocumentDataInputSchema.

TYPE_MAPPING

GetDocumentDataInputSchema.

dict_class

GetDocumentDataInputSchema.

error_messages

GetDocumentDataInputSchema.opts

GetDocumentDataInputSchema.

set_class

app.utils.marshmallow_schema.GetDocumentDataInputSchema.TYPE_MAPPING

GetDocumentDataInputSchema.**TYPE_MAPPING** = {<class 'str':>: <class 'marshmallow.fiel

app.utils.marshmallow_schema.GetDocumentDataInputSchema.dict_class

property GetDocumentDataInputSchema.**dict_class**

app.utils.marshmallow_schema.GetDocumentDataInputSchema.error_messages

GetDocumentDataInputSchema.**error_messages** = {}

app.utils.marshmallow_schema.GetDocumentDataInputSchema.opts

GetDocumentDataInputSchema.**opts** = <marshmallow.schema.SchemaOpts object>

app.utils.marshmallow_schema.GetDocumentDataInputSchema.set_class

property GetDocumentDataInputSchema.**set_class**

Methods

<i>GetDocumentDataInputSchema.__init__(*[, ...])</i>	Initialize self.
<i>GetDocumentDataInputSchema.convert_to_integer(...)</i>	
<i>GetDocumentDataInputSchema.dump(obj, *[, many])</i>	Serialize an object to native Python data types according to this Schema's fields.
<i>GetDocumentDataInputSchema.dumps(obj, *args)</i>	Same as <i>dump()</i> , except return a JSON-encoded string.
<i>GetDocumentDataInputSchema.from_dict(fields, *)</i>	Generate a <i>Schema</i> class given a dictionary of fields.
<i>GetDocumentDataInputSchema.get_attribute(...)</i>	Defines how to pull values from an object to serialize.
<i>GetDocumentDataInputSchema.handle_error(...)</i>	Custom error handler function for the schema.
<i>GetDocumentDataInputSchema jsonify(obj[, many])</i>	Return a JSON response containing the serialized data.
<i>GetDocumentDataInputSchema.load(data, *[, ...])</i>	Deserialize a data structure to an object defined by this Schema's fields.
<i>GetDocumentDataInputSchema.loads(json_data, *)</i>	Same as <i>load()</i> , except it takes a JSON string as input.
<i>GetDocumentDataInputSchema.on_bind_field(...)</i>	Hook to modify a field when it is bound to the <i>Schema</i> .
<i>GetDocumentDataInputSchema.validate(data, *)</i>	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.__init__

```
GetDocumentDataInputSchema.__init__(*, only: Union[Sequence[str], Set[str]]
                                     = None, exclude: Union[Sequence[str],
                                     Set[str]] = (), many: bool = False,
                                     context: Dict = None, load_only:
                                     Union[Sequence[str], Set[str]] = (),
                                     dump_only: Union[Sequence[str],
                                     Set[str]] = (), partial: Union[bool, Se-
                                     quence[str], Set[str]] = False, unknown:
                                     str = None)
```

Initialize self. See help(type(self)) for accurate signature.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.convert_to_integer

```
GetDocumentDataInputSchema.convert_to_integer(value, many, **kwargs)
```

app.utils.marshmallow_schema.GetDocumentDataInputSchema.dump

```
GetDocumentDataInputSchema.dump(obj: Any, *, many: bool = None)
```

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.dumps

```
GetDocumentDataInputSchema.dumps(obj: Any, *args, many: bool = None,
                                  **kwargs)
```

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.from_dict

classmethod `GetDocumentDataInputSchema.from_dict` (*fields: Dict[str, Union[marshmallow.fields.Field, type]], *, name: str = 'GeneratedSchema'*)
→ type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the repr for the class.

New in version 3.0.0.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.get_attribute

`GetDocumentDataInputSchema.get_attribute` (*obj: Any, attr: str, default: Any*)
Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.handle_error

`GetDocumentDataInputSchema.handle_error` (*error: marshmallow.exceptions.ValidationError, data: Any, *, many: bool, **kwargs*)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.jsonify

GetDocumentDataInputSchema.**jsonify**(*obj*, *many=<object object>*, **args*, ***kwargs*)

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.load

GetDocumentDataInputSchema.**load**(*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]]*, ***, *many: bool = None*, *partial: Union[bool, Sequence[str], Set[str]] = None*, *unknown: str = None*)

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns

Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.loads

GetDocumentDataInputSchema.**loads**(*json_data: str*, ***, *many: bool = None*, *partial: Union[bool, Sequence[str], Set[str]] = None*, *unknown: str = None*, ***kwargs*)

Same as *load()*, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing

fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a *(data, errors)* tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.on_bind_field

GetDocumentDataInputSchema.on_bind_field (*field_name: str, field_obj: marshmallow.fields.Field*) → *None*

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

app.utils.marshmallow_schema.GetDocumentDataInputSchema.validate

GetDocumentDataInputSchema.validate (*data: Mapping, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None*) → *Dict[str, List[str]]*

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

app.utils.marshmallow_schema.RoleSchema

```
class app.utils.marshmallow_schema.RoleSchema(*, only: Union[Sequence[str], Set[str]]
                                              = None, exclude: Union[Sequence[str],
                                              Set[str]] = (), many: bool = False,
                                              context: Dict = None, load_only:
                                              Union[Sequence[str], Set[str]] = (),
                                              dump_only: Union[Sequence[str],
                                              Set[str]] = (), partial: Union[bool,
                                              Sequence[str], Set[str]] = False, unknown:
                                              str = None)
```

Bases: *flask_marshmallow.schema.Schema*

Attributes

<code>RoleSchema.TYPE_MAPPING</code>
<code>RoleSchema.dict_class</code>
<code>RoleSchema.error_messages</code>
<code>RoleSchema.opts</code>
<code>RoleSchema.set_class</code>

`app.utils.marshmallow_schema.RoleSchema.TYPE_MAPPING`

`RoleSchema.TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <cl`

`app.utils.marshmallow_schema.RoleSchema.dict_class`

property `RoleSchema.dict_class`

`app.utils.marshmallow_schema.RoleSchema.error_messages`

`RoleSchema.error_messages = {}`

`app.utils.marshmallow_schema.RoleSchema.opts`

`RoleSchema.opts = <marshmallow.schema.SchemaOpts object>`

`app.utils.marshmallow_schema.RoleSchema.set_class`

property `RoleSchema.set_class`

Methods

<code>RoleSchema.__init__(*[, only, exclude, ...])</code>	Initialize self.
<code>RoleSchema.dump(obj, *[, many])</code>	Serialize an object to native Python data types according to this Schema's fields.
<code>RoleSchema.dumps(obj, *args[, many])</code>	Same as <code>dump()</code> , except return a JSON-encoded string.
<code>RoleSchema.from_dict(fields, *[, name])</code>	Generate a <i>Schema</i> class given a dictionary of fields.
<code>RoleSchema.get_attribute(obj, attr, default)</code>	Defines how to pull values from an object to serialize.
<code>RoleSchema.handle_error(error, data, *, ...)</code>	Custom error handler function for the schema.
<code>RoleSchema.jsonify(obj[, many])</code>	Return a JSON response containing the serialized data.

continues on next page

Table 97 – continued from previous page

<code>RoleSchema.load(data, *[, many, partial, ...])</code>	Deserialize a data structure to an object defined by this Schema's fields.
<code>RoleSchema.loads(json_data, *[, many, ...])</code>	Same as <code>load()</code> , except it takes a JSON string as input.
<code>RoleSchema.on_bind_field(field_name, field_obj)</code>	Hook to modify a field when it is bound to the Schema.
<code>RoleSchema.valid_request_role(data)</code>	
<code>RoleSchema.validate(data, *[, many, partial])</code>	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

app.utils.marshmallow_schema.RoleSchema.__init__

`RoleSchema.__init__(*, only: Union[Sequence[str], Set[str]] = None, exclude: Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str], Set[str]] = False, unknown: str = None)`

Initialize self. See `help(type(self))` for accurate signature.

app.utils.marshmallow_schema.RoleSchema.dump

`RoleSchema.dump(obj: Any, *, many: bool = None)`

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.RoleSchema.dumps

`RoleSchema.dumps(obj: Any, *args, many: bool = None, **kwargs)`

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

`app.utils.marshmallow_schema.RoleSchema.from_dict`

classmethod `RoleSchema.from_dict` (*fields:* *Dict[str, Union[mashmallow.fields.Field, type]],*
**, name: str = 'GeneratedSchema'*) → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the repr for the class.

New in version 3.0.0.

`app.utils.marshmallow_schema.RoleSchema.get_attribute`

`RoleSchema.get_attribute` (*obj: Any, attr: str, default: Any*)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

`app.utils.marshmallow_schema.RoleSchema.handle_error`

`RoleSchema.handle_error` (*error: marshmallow.exceptions.ValidationError, data: Any,*
**, many: bool, **kwargs*)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.RoleSchema.jsonify

`RoleSchema.jsonify(obj, many=<object object>, *args, **kwargs)`

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to *False*, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.RoleSchema.load

`RoleSchema.load(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None)`

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns

Deserialized data

New in version 1.0.0.
Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.RoleSchema.loads

`RoleSchema.loads(json_data: str, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, **kwargs)`

Same as *load()*, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

`app.utils.marshmallow_schema.RoleSchema.on_bind_field`

`RoleSchema.on_bind_field(field_name: str, field_obj: marshmallow.fields.Field) →`

`None`

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

`app.utils.marshmallow_schema.RoleSchema.valid_request_role`

static `RoleSchema.valid_request_role(data: dict) → dict`

`app.utils.marshmallow_schema.RoleSchema.validate`

`RoleSchema.validate(data: Mapping, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]`

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

`app.utils.marshmallow_schema.SearchSchema`

```
class app.utils.marshmallow_schema.SearchSchema(*, only: Union[Sequence[str],
Set[str]] = None, exclude: Union[Sequence[str], Set[str]]
= (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (),
dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str], Set[str]] = False,
unknown: str = None)
```

Bases: `flask_marshmallow.schema.Schema`

Attributes

<i>SearchSchema.TYPE_MAPPING</i>
<i>SearchSchema.dict_class</i>
<i>SearchSchema.error_messages</i>
<i>SearchSchema.opts</i>
<i>SearchSchema.set_class</i>

app.utils.marshmallow_schema.SearchSchema.TYPE_MAPPING

```
SearchSchema.TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <
```

app.utils.marshmallow_schema.SearchSchema.dict_class

```
property SearchSchema.dict_class
```

app.utils.marshmallow_schema.SearchSchema.error_messages

```
SearchSchema.error_messages = {}
```

app.utils.marshmallow_schema.SearchSchema.opts

```
SearchSchema.opts = <marshmallow.schema.SchemaOpts object>
```

app.utils.marshmallow_schema.SearchSchema.set_class

```
property SearchSchema.set_class
```

Methods

<i>SearchSchema.__init__</i> (*[, only, exclude, ...])	Initialize self.
<i>SearchSchema.dump</i> (obj, *[, many])	Serialize an object to native Python data types according to this Schema's fields.
<i>SearchSchema.dumps</i> (obj, *args[, many])	Same as <i>dump()</i> , except return a JSON-encoded string.
<i>SearchSchema.from_dict</i> (fields, *[, name])	Generate a <i>Schema</i> class given a dictionary of fields.
<i>SearchSchema.get_attribute</i> (obj, attr, default)	Defines how to pull values from an object to serialize.
<i>SearchSchema.handle_error</i> (error, data, *, ...)	Custom error handler function for the schema.
<i>SearchSchema.jsonify</i> (obj[, many])	Return a JSON response containing the serialized data.

continues on next page

Table 99 – continued from previous page

<code>SearchSchema.load(data, *[, many, partial, ...])</code>	Deserialize a data structure to an object defined by this Schema's fields.
<code>SearchSchema.loads(json_data, *[, many, ...])</code>	Same as <code>load()</code> , except it takes a JSON string as input.
<code>SearchSchema.on_bind_field(field_name, field_obj)</code>	Hook to modify a field when it is bound to the Schema.
<code>SearchSchema.validate(data, *[, many, partial])</code>	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

app.utils.marshmallow_schema.SearchSchema.__init__

`SearchSchema.__init__(*, only: Union[Sequence[str], Set[str]] = None, exclude: Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str], Set[str]] = False, unknown: str = None)`

Initialize self. See `help(type(self))` for accurate signature.

app.utils.marshmallow_schema.SearchSchema.dump

`SearchSchema.dump(obj: Any, *, many: bool = None)`

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.SearchSchema.dumps

`SearchSchema.dumps(obj: Any, *args, many: bool = None, **kwargs)`

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

app.utils.marshmallow_schema.SearchSchema.from_dict

classmethod SearchSchema.**from_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], *, *name*: str = 'GeneratedSchema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the repr for the class.

New in version 3.0.0.

app.utils.marshmallow_schema.SearchSchema.get_attribute

SearchSchema.**get_attribute** (*obj*: Any, *attr*: str, *default*: Any)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

app.utils.marshmallow_schema.SearchSchema.handle_error

SearchSchema.**handle_error** (*error*: marshmallow.exceptions.ValidationError, *data*: Any, *, *many*: bool, **kwargs)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of *many* on dump or load.
- **partial** – Value of *partial* on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.SearchSchema.jsonify

`SearchSchema.jsonify(obj, many=<object object>, *args, **kwargs)`

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to *False*, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.SearchSchema.load

`SearchSchema.load(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None)`

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.SearchSchema.loads

`SearchSchema.loads(json_data: str, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, **kwargs)`

Same as *load()*, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a *(data, errors)* tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.SearchSchema.on_bind_field

`SearchSchema.on_bind_field` (*field_name: str, field_obj: marshmallow.fields.Field*)
→ *None*

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

app.utils.marshmallow_schema.SearchSchema.validate

`SearchSchema.validate` (*data: Mapping, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None*) → *Dict[str, List[str]]*

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

app.utils.marshmallow_schema.Timestamp

```
class app.utils.marshmallow_schema.Timestamp(*, default: Any = <marshmallow.missing>,
                                              missing: Any = <marshmallow.missing>,
                                              data_key: str = None, attribute: str = None,
                                              validate: Union[Callable[[Any], Any], Iterable[Callable[[Any], Any]]] = None,
                                              required: bool = False, allow_none: bool = None, load_only: bool = False,
                                              dump_only: bool = False, error_messages: Dict[str, str] = None,
                                              **metadata)
```

Bases: *marshmallow.fields.Field*

Field that serializes to timestamp integer and deserializes to a *datetime.datetime* class.

Attributes

<code>Timestamp.context</code>	The context dictionary for the parent Schema.
<code>Timestamp.default_error_messages</code>	
<code>Timestamp.name</code>	
<code>Timestamp.parent</code>	
<code>Timestamp.root</code>	Reference to the <i>Schema</i> that this field belongs to even if it is buried in a container field (e.g.

`app.utils.marshmallow_schema.Timestamp.context`

property `Timestamp.context`
The context dictionary for the parent Schema.

`app.utils.marshmallow_schema.Timestamp.default_error_messages`

`Timestamp.default_error_messages = {'null': 'Field may not be null.', 'required':`

`app.utils.marshmallow_schema.Timestamp.name`

`Timestamp.name = None`

`app.utils.marshmallow_schema.Timestamp.parent`

`Timestamp.parent = None`

`app.utils.marshmallow_schema.Timestamp.root`

property `Timestamp.root`
Reference to the *Schema* that this field belongs to even if it is buried in a container field (e.g. *List*). Return *None* for unbound fields.

Methods

<code>Timestamp.__init__(*[, default, missing, ...])</code>	Initialize self.
<code>Timestamp.deserialize(value[, attr, data])</code>	Deserialize value.
<code>Timestamp.fail(key, **kwargs)</code>	Helper method that raises a <i>ValidationError</i> with an error message from <code>self.error_messages</code> .
<code>Timestamp.get_value(obj, attr[, accessor, ...])</code>	Return the value for a given key from an object.

continues on next page

Table 101 – continued from previous page

<code>Timestamp.make_error(key, **kwargs)</code>	Helper method to make a <i>ValidationError</i> with an error message from <code>self.error_messages</code> .
<code>Timestamp.serialize(attr, obj[, accessor])</code>	Pulls the value for the given key from the object, applies the field's formatting and returns the result.

app.utils.marshmallow_schema.Timestamp.__init__

`Timestamp.__init__(*, default: Any = <marshmallow.missing>, missing: Any = <marshmallow.missing>, data_key: str = None, attribute: str = None, validate: Union[Callable[[Any], Any], Iterable[Callable[[Any], Any]]] = None, required: bool = False, allow_none: bool = None, load_only: bool = False, dump_only: bool = False, error_messages: Dict[str, str] = None, **metadata) → None`

Initialize self. See `help(type(self))` for accurate signature.

app.utils.marshmallow_schema.Timestamp.deserialize

`Timestamp.deserialize(value: Any, attr: str = None, data: Mapping[str, Any] = None, **kwargs)`

Deserialize value.

Parameters

- **value** – The value to deserialize.
- **attr** – The attribute/key in `data` to deserialize.
- **data** – The raw input data passed to `Schema.load`.
- **kwargs** – Field-specific keyword arguments.

Raises `ValidationError` – If an invalid value is passed or if a required value is missing.

app.utils.marshmallow_schema.Timestamp.fail

`Timestamp.fail(key: str, **kwargs)`

Helper method that raises a *ValidationError* with an error message from `self.error_messages`.

Deprecated since version 3.0.0: Use `make_error <marshmallow.fields.Field.make_error>` instead.

app.utils.marshmallow_schema.Timestamp.get_value

`Timestamp.get_value(obj, attr, accessor=None, default=<marshmallow.missing>)`

Return the value for a given key from an object.

Parameters

- **obj** (*object*) – The object to get the value from.
- **attr** (*str*) – The attribute/key in `obj` to get the value from.
- **accessor** (*callable*) – A callable used to retrieve the value of `attr` from the object `obj`. Defaults to `marshmallow.utils.get_value`.

app.utils.marshmallow_schema.Timestamp.make_error

`Timestamp.make_error(key: str, **kwargs) → marshmallow.exceptions.ValidationError`
 Helper method to make a *ValidationError* with an error message from `self.error_messages`.

app.utils.marshmallow_schema.Timestamp.serialize

`Timestamp.serialize(attr: str, obj: Any, accessor: Callable[[Any, str, Any], Any] = None, **kwargs)`
 Pulls the value for the given key from the object, applies the field's formatting and returns the result.

Parameters

- **attr** – The attribute/key to get from the object.
- **obj** – The object to access the attribute/key from.
- **accessor** – Function used to access values from `obj`.
- **kwargs** – Field-specific keyword arguments.

app.utils.marshmallow_schema.UserSchema

```
class app.utils.marshmallow_schema.UserSchema(* , only: Union[Sequence[str], Set[str]]
                                         = None, exclude: Union[Sequence[str],
                                         Set[str]] = (), many: bool = False,
                                         context: Dict = None, load_only:
                                         Union[Sequence[str], Set[str]] = (),
                                         dump_only: Union[Sequence[str],
                                         Set[str]] = (), partial: Union[bool, Se-
                                         quence[str], Set[str]] = False, unknown:
                                         str = None)
```

Bases: `flask_marshmallow.schema.Schema`

Attributes

`UserSchema.TYPE_MAPPING`

`UserSchema.dict_class`

`UserSchema.error_messages`

`UserSchema.opts`

`UserSchema.set_class`

app.utils.marshmallow_schema.UserSchema.TYPE_MAPPING

```
UserSchema.TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <cl
```

app.utils.marshmallow_schema.UserSchema.dict_class

```
property UserSchema.dict_class
```

app.utils.marshmallow_schema.UserSchema.error_messages

```
UserSchema.error_messages = {}
```

app.utils.marshmallow_schema.UserSchema.opts

```
UserSchema.opts = <marshmallow.schema.SchemaOpts object>
```

app.utils.marshmallow_schema.UserSchema.set_class

```
property UserSchema.set_class
```

Methods

<code>UserSchema.__init__(*[, only, exclude, ...])</code>	Initialize self.
<code>UserSchema.dump(obj, *[, many])</code>	Serialize an object to native Python data types according to this Schema's fields.
<code>UserSchema.dumps(obj, *args[, many])</code>	Same as <code>dump()</code> , except return a JSON-encoded string.
<code>UserSchema.from_dict(fields, *[, name])</code>	Generate a <i>Schema</i> class given a dictionary of fields.
<code>UserSchema.get_attribute(obj, attr, default)</code>	Defines how to pull values from an object to serialize.
<code>UserSchema.handle_error(error, data, *, ...)</code>	Custom error handler function for the schema.
<code>UserSchema.jsonify(obj[, many])</code>	Return a JSON response containing the serialized data.
<code>UserSchema.load(data, *[, many, partial, ...])</code>	Deserialize a data structure to an object defined by this Schema's fields.
<code>UserSchema.loads(json_data, *[, many, ...])</code>	Same as <code>load()</code> , except it takes a JSON string as input.
<code>UserSchema.on_bind_field(field_name, field_obj)</code>	Hook to modify a field when it is bound to the <i>Schema</i> .
<code>UserSchema.valid_request_user(data)</code>	
<code>UserSchema.validate(data, *[, many, partial])</code>	Validate <i>data</i> against the schema, returning a dictionary of validation errors.

continues on next page

Table 103 – continued from previous page

<code>UserSchema.</code>
<code>validate_credentials(data)</code>
<code>UserSchema.validate_email(data)</code>
<code>UserSchema.</code>
<code>validate_password(password)</code>

app.utils.marshmallow_schema.UserSchema.__init__

`UserSchema.__init__`(**only*: Union[Sequence[str], Set[str]] = None, *exclude*: Union[Sequence[str], Set[str]] = (), *many*: bool = False, *context*: Dict = None, *load_only*: Union[Sequence[str], Set[str]] = (), *dump_only*: Union[Sequence[str], Set[str]] = (), *partial*: Union[bool, Sequence[str], Set[str]] = False, *unknown*: str = None)

Initialize self. See help(type(self)) for accurate signature.

app.utils.marshmallow_schema.UserSchema.dump

`UserSchema.dump`(*obj*: Any, *, *many*: bool = None)

Serialize an object to native Python data types according to this Schema's fields.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A dict of serialized data

Return type dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

app.utils.marshmallow_schema.UserSchema.dumps

`UserSchema.dumps`(*obj*: Any, **args*, *many*: bool = None, ***kwargs*)

Same as `dump()`, except return a JSON-encoded string.

Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

Returns A json string

Return type str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

app.utils.marshmallow_schema.UserSchema.from_dict

classmethod UserSchema.**from_dict** (*fields:* Dict[str, Union[mashmallow.fields.Field, type]], *, *name:* str = 'GeneratedSchema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the repr for the class.

New in version 3.0.0.

app.utils.marshmallow_schema.UserSchema.get_attribute

UserSchema.**get_attribute** (*obj:* Any, *attr:* str, *default:* Any)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

app.utils.marshmallow_schema.UserSchema.handle_error

UserSchema.**handle_error** (*error:* marshmallow.exceptions.ValidationError, *data:* Any, *, *many:* bool, ***kwargs*)

Custom error handler function for the schema.

Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of *many* on dump or load.
- **partial** – Value of *partial* on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

app.utils.marshmallow_schema.UserSchema.jsonify

`UserSchema.jsonify(obj, many=<object object>, *args, **kwargs)`

Return a JSON response containing the serialized data.

Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to *False*, regardless of the value of *Schema.many*.

app.utils.marshmallow_schema.UserSchema.load

`UserSchema.load(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None)`

Deserialize a data structure to an object defined by this Schema's fields.

Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

app.utils.marshmallow_schema.UserSchema.loads

`UserSchema.loads(json_data: str, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, **kwargs)`

Same as *load()*, except it takes a JSON string as input.

Parameters

- **json_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

Returns Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

app.utils.marshmallow_schema.UserSchema.on_bind_field

`UserSchema.on_bind_field` (*field_name*: str, *field_obj*: *marshmallow.fields.Field*) → None

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

app.utils.marshmallow_schema.UserSchema.valid_request_user

static `UserSchema.valid_request_user` (*data*: dict) → dict

app.utils.marshmallow_schema.UserSchema.validate

`UserSchema.validate` (*data*: Mapping, *, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]

Validate *data* against the schema, returning a dictionary of validation errors.

Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

Returns A dictionary of validation errors.

New in version 1.1.0.

app.utils.marshmallow_schema.UserSchema.validate_credentials

`UserSchema.validate_credentials` (*data*: dict) → dict

app.utils.marshmallow_schema.UserSchema.validate_email

static `UserSchema.validate_email` (*data*: dict) → dict


```
class Meta
```

OPTIONS_CLASS

```
TYPE_MAPPING = {<class 'str': <class 'marshmallow.fields.String'>, <clas
```

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field `load_only` and `dump_only` values if `field_name` was specified in class `Meta`.

Call `getter func` with `data` as its argument, and store any *ValidationErrors*.

Parameters

- **getter func** (*callable*) – Function for getting the serialized/deserialized value

- ```
declared fields = {'created at': <fields.Timestamp(default=<marshmallow.
```

```
_deserialize (data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, error_store: marsh-
```

```
Deserialize data.
```

## Parameters

- **partial** (*bool/tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**`_do_load`** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool = True*)

Deserialize *data*, returning the deserialized result. This method is private API.

**Parameters**

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

**`_has_processors`** (*tag*) → *bool*

**`_hooks`** = {('post\_dump', False): ['make\_url']}

**`_init_fields`** () → *None*

Update *self.fields*, *self.load\_fields*, and *self.dump\_fields* based on schema options. This method is private API.

**`_invoke_dump_processors`** (*tag: str, data, \*, many: bool, original\_data=None*)

**`_invoke_field_validators`** (*\*, error\_store: marshmallow.error\_store.ErrorStore, data, many: bool*)

**`_invoke_load_processors`** (*tag: str, data, \*, many: bool, original\_data, partial: Union[bool, Sequence[str], Set[str]]*)

**`_invoke_processors`** (*tag: str, \*, pass\_many: bool, data, many: bool, original\_data=None, \*\*kwargs*)

**`_invoke_schema_validators`** (*\*, error\_store: marshmallow.error\_store.ErrorStore, pass\_many: bool, data, original\_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field\_errors: bool = False*)

**`_normalize_nested_options`** () → *None*

Apply then flatten nested schema options. This method is private API.

**`_run_validator`** (*validator\_func, output, \*, original\_data, error\_store, many, partial, pass\_original, index=None*)

**\_serialize** (*obj: Union[\_T, Iterable[\_T]], \*, many: bool = False*)  
Serialize *obj*.

**Parameters**

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from `marshal`.

**property dict\_class**

**dump** (*obj: Any, \*, many: bool = None*)  
Serialize an object to native Python data types according to this Schema's fields.

**Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dumps** (*obj: Any, \*args, many: bool = None, \*\*kwargs*)  
Same as `dump()`, except return a JSON-encoded string.

**Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

**error\_messages = {}**

**classmethod from\_dict** (*fields: Dict[str, Union[marshmallow.fields.Field, type]], \*, name: str = 'GeneratedSchema'*) → type  
Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

**Parameters**

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj: Any, attr: str, default: Any*)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

**handle\_error** (*error: marshmallow.exceptions.ValidationError, data: Any, \*, many: bool, \*\*kwargs*)

Custom error handler function for the schema.

#### Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj, many=<object object>, \*args, \*\*kwargs*)

Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to `False`, regardless of the value of *Schema.many*.

**load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial:*

*Union[bool, Sequence[str], Set[str]] = None, unknown: str = None*)

Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to Nested fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as `load()`, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use `EXCLUDE`, `INCLUDE` or `RAISE`. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**make\_url** (*data*, \*\*kwargs)

**on\_bind\_field** (*field\_name*: str, *field\_obj*: `marshmallow.fields.Field`) → None

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = `<marshmallow.schema.SchemaOpts object>`

**property** `set_class`

**static** `valid_request_file` (*data*)

**validate** (*data*: Mapping, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]

Validate *data* against the schema, returning a dictionary of validation errors.

#### Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema.ExportWordInputSchema(*, only:
 Union[Sequence[str],
 Set[str]] =
 None, exclude:
 Union[Sequence[str],
 Set[str]] = (), many:
 bool = False, context:
 Dict = None, load_only:
 Union[Sequence[str],
 Set[str]] = (),
 dump_only:
 Union[Sequence[str],
 Set[str]] = (), partial:
 Union[bool, Sequence[str], Set[str]] =
 False, unknown: str =
 None)
```

#### **class Meta**

Options object for a Schema.

Example usage:

```
class Meta:
 fields = ("id", "email", "date_created")
 exclude = ("password", "secret_attribute")
```

Available options:

- **fields:** Tuple or list of fields to include in the serialized result.
- **additional:** Tuple or list of fields to include *in addition* to the explicitly declared fields. **additional** and **fields** are mutually-exclusive options.
- **include:** Dictionary of additional fields to include in the schema. It is usually better to define fields as class variables, but you may need to use this option, e.g., if your fields are Python keywords. May be an *OrderedDict*.
- **exclude:** Tuple or list of fields to exclude in the serialized result. Nested fields can be represented with dot delimiters.
- **dateformat:** Default format for *Date* <fields.Date> fields.
- **datetimeformat:** Default format for *DateTime* <fields.DateTime> fields.
- **render\_module:** Module to use for *loads* <Schema.loads> and *dumps* <Schema.dumps>. Defaults to *json* from the standard library.
- **ordered:** If *True*, order serialization output according to the order in which fields were declared. Output of *Schema.dump* will be a *collections.OrderedDict*.
- **index\_errors:** If *True*, errors dictionaries will include the **index** of invalid items in a collection.
- **load\_only:** Tuple or list of fields to exclude from serialized results.
- **dump\_only:** Tuple or list of fields to exclude from deserialization
- **unknown:** Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.

- **register:** Whether to register the *Schema* with marshmallow's internal class registry. Must be *True* if you intend to refer to this *Schema* by class name in *Nested* fields. Only set this to *False* when memory usage is critical. Defaults to *True*.

**OPTIONS\_CLASS**

alias of `marshmallow.schema.SchemaOpts`

**TYPE\_MAPPING** = {<class 'str':>: <class 'marshmallow.fields.String'>, <class 'bytes'>:

**\_bind\_field** (*field\_name*: str, *field\_obj*: marshmallow.fields.Field) → None

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field *load\_only* and *dump\_only* values if *field\_name* was specified in class *Meta*.

**static \_call\_and\_store** (*getter\_func*, *data*, \*, *field\_name*, *error\_store*, *index=None*)

Call *getter\_func* with *data* as its argument, and store any *ValidationErrors*.

**Parameters**

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from *data*.
- **data** – The data passed to *getter\_func*.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

**\_declared\_fields** = {'to\_pdf': <fields.Integer(default=<marshmallow.missing>, attribut

**\_default\_error\_messages** = {'type': 'Invalid input type.', 'unknown': 'Unknown field.

**\_deserialize** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *error\_store*: marshmallow.error\_store.ErrorStore, *many*: bool = False, *partial*=False, *unknown*='raise', *index=None*) → Union[\_T, List[\_T]]

Deserialize *data*.

**Parameters**

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if *data* should be deserialized as a collection.
- **partial** (*bool/tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**\_do\_load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, *postprocess*: bool = True)

Deserialize *data*, returning the deserialized result. This method is private API.

**Parameters**

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

```
_has_processors (tag) → bool
_hooks = {('pre_load', False): ['convert_to_integer']}
_init_fields () → None
 Update self.fields, self.load_fields, and self.dump_fields based on schema options. This method is private
 API.
_invoke_dump_processors (tag: str, data, *, many: bool, original_data=None)
_invoke_field_validators (*, error_store: marshmallow.error_store.ErrorStore, data, many:
 bool)
_invoke_load_processors (tag: str, data, *, many: bool, original_data, partial: Union[bool,
 Sequence[str], Set[str]])
_invoke_processors (tag: str, *, pass_many: bool, data, many: bool, original_data=None,
 **kwargs)
_invoke_schema_validators (*, error_store: marshmallow.error_store.ErrorStore, pass_many:
 bool, data, original_data, many: bool, partial: Union[bool, Se-
 quence[str], Set[str]], field_errors: bool = False)
_normalize_nested_options () → None
 Apply then flatten nested schema options. This method is private API.
_run_validator (validator_func, output, *, original_data, error_store, many, partial, pass_original,
 index=None)
_serialize (obj: Union[_T, Iterable[_T]], *, many: bool = False)
 Serialize obj.
```

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from *marshal*.

```
convert_to_integer (value, many, **kwargs)
```

```
property dict_class
```

```
dump (obj: Any, *, many: bool = None)
```

Serialize an object to native Python data types according to this Schema's fields.

#### Parameters

- **obj** – The object to serialize.



- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dumps** (*obj*: Any, \*args, many: bool = None, \*\*kwargs)

Same as `dump()`, except return a JSON-encoded string.

**Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], \*, name: str = 'GeneratedSchema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

**Parameters**

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj*: Any, attr: str, default: Any)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

**handle\_error** (*error*: marshmallow.exceptions.ValidationError, data: Any, \*, many: bool, \*\*kwargs)

Custom error handler function for the schema.

**Parameters**

- **error** – The `ValidationError` raised during (de)serialization.
- **data** – The original input data.

- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj*, *many*=<object object>, \*args, \*\*kwargs)  
Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None)  
Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as *load()*, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (data, errors) tuple. A `ValidationError` is raised if invalid data are passed.

**on\_bind\_field** (field\_name: str, field\_obj: marshmallow.fields.Field) → None  
Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**validate** (data: Mapping, \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]

Validate *data* against the schema, returning a dictionary of validation errors.

**Parameters**

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema.GetDocumentDataInputSchema (*,
 only:
 Union[Sequence[str],
 Set[str]] =
 None, exclude:
 Union[Sequence[str],
 Set[str]] = (),
 many: bool
 = False, con-
 text: Dict =
 None, load_only:
 Union[Sequence[str],
 Set[str]] =
 (), dump_only:
 Union[Sequence[str],
 Set[str]] =
 (), partial:
 Union[bool,
 Sequence[str],
 Set[str]] = False,
 unknown: str =
 None)
```

**class Meta**

Options object for a Schema.

Example usage:

```
class Meta:
 fields = ("id", "email", "date_created")
 exclude = ("password", "secret_attribute")
```

Available options:

- **fields:** Tuple or list of fields to include in the serialized result.
- **additional:** Tuple or list of fields to include *in addition to the* explicitly declared fields. **additional** and **fields** are mutually-exclusive options.
- **include:** Dictionary of additional fields to include in the schema. It is usually better to define fields as class variables, but you may need to use this option, e.g., if your fields are Python keywords. May be an *OrderedDict*.
- **exclude:** Tuple or list of fields to exclude in the serialized result. Nested fields can be represented with dot delimiters.
- **dateformat:** Default format for *Date* <fields.Date> fields.
- **datetimeformat:** Default format for *DateTime* <fields.DateTime> fields.
- **render\_module:** Module to use for *loads* <Schema.loads> and *dumps* <Schema.dumps>. Defaults to *json* from the standard library.
- **ordered:** If *True*, order serialization output according to the order in which fields were declared. Output of *Schema.dump* will be a *collections.OrderedDict*.
- **index\_errors:** If *True*, errors dictionaries will include the **index** of invalid items in a collection.
- **load\_only:** Tuple or list of fields to exclude from serialized results.
- **dump\_only:** Tuple or list of fields to exclude from deserialization
- **unknown:** Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **register:** Whether to register the *Schema* with marshmallow's internal class registry. Must be *True* if you intend to refer to this *Schema* by class name in *Nested* fields. Only set this to *False* when memory usage is critical. Defaults to *True*.

#### OPTIONS\_CLASS

alias of `marshmallow.schema.SchemaOpts`

**TYPE\_MAPPING** = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:

**\_bind\_field** (*field\_name*: str, *field\_obj*: marshmallow.fields.Field) → None

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field *load\_only* and *dump\_only* values if *field\_name* was specified in `class Meta`.

**static \_call\_and\_store** (*getter\_func*, *data*, \*, *field\_name*, *error\_store*, *index=None*)

Call *getter\_func* with *data* as its argument, and store any *ValidationErrors*.

#### Parameters

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from *data*.
- **data** – The data passed to *getter\_func*.
- **field\_name** (*str*) – Field name.

- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

**\_declared\_fields** = {'as\_attachment': <fields.Integer(default=<marshmallow.missing>, a

**\_default\_error\_messages** = {'type': 'Invalid input type.', 'unknown': 'Unknown field.

**\_deserialize** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, error\_store: marshmallow.error\_store.ErrorStore, many: bool = False, partial=False, unknown='raise', index=None*) → Union[\_T, List[\_T]]

Deserialize data.

#### Parameters

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if data should be deserialized as a collection.
- **partial** (*bool|tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**\_do\_load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool = True*)

Deserialize *data*, returning the deserialized result. This method is private API.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

**\_has\_processors** (*tag*) → bool

**\_hooks** = {('pre\_load', False): ['convert\_to\_integer']}

**\_init\_fields** () → None

Update *self.fields*, *self.load\_fields*, and *self.dump\_fields* based on schema options. This method is private API.

**\_invoke\_dump\_processors** (*tag: str, data, \*, many: bool, original\_data=None*)

```
_invoke_field_validators (*, error_store: marshmallow.error_store.ErrorStore, data, many: bool)
_invoke_load_processors (tag: str, data, *, many: bool, original_data, partial: Union[bool, Sequence[str], Set[str]])
_invoke_processors (tag: str, *, pass_many: bool, data, many: bool, original_data=None, **kwargs)
_invoke_schema_validators (*, error_store: marshmallow.error_store.ErrorStore, pass_many: bool, data, original_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field_errors: bool = False)
_normalize_nested_options () → None
 Apply then flatten nested schema options. This method is private API.
_run_validator (validator_func, output, *, original_data, error_store, many, partial, pass_original, index=None)
_serialize (obj: Union[_T, Iterable[_T]], *, many: bool = False)
 Serialize obj.
```

**Parameters**

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from `marshal`.

```
convert_to_integer (value, many, **kwargs)
```

**property dict\_class**

```
dump (obj: Any, *, many: bool = None)
```

Serialize an object to native Python data types according to this Schema's fields.

**Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

```
dumps (obj: Any, *args, many: bool = None, **kwargs)
```

Same as `dump()`, except return a JSON-encoded string.

**Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields: Dict[str, Union[marshmallow.fields.Field, type]], \*, name: str = 'GeneratedSchema'*) → type  
Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj: Any, attr: str, default: Any*)  
Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

**handle\_error** (*error: marshmallow.exceptions.ValidationError, data: Any, \*, many: bool, \*\*kwargs*)  
Custom error handler function for the schema.

#### Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj, many=<object object>, \*args, \*\*kwargs*)  
Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None)  
Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (data, errors) tuple. A *ValidationError* is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as *load()*, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (data, errors) tuple. A *ValidationError* is raised if invalid data are passed.

**on\_bind\_field** (*field\_name*: str, *field\_obj*: *marshmallow.fields.Field*) → None  
Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**validate** (*data*: Mapping, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]  
Validate *data* against the schema, returning a dictionary of validation errors.



**Parameters**

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema.RoleSchema(* , only: Union[Sequence[str], Set[str]]
 = None, exclude: Union[Sequence[str],
 Set[str]] = (), many: bool = False,
 context: Dict = None, load_only:
 Union[Sequence[str], Set[str]] = (),
 dump_only: Union[Sequence[str],
 Set[str]] = (), partial: Union[bool, Se-
 quence[str], Set[str]] = False, unknown:
 str = None)
```

```
class Meta
```

```
 fields = ('id', 'name', 'description', 'label', 'created_at', 'updated_at', 'delete
 ordered = True
```

```
OPTIONS_CLASS
```

```
 alias of marshmallow.schema.SchemaOpts
```

```
TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:
```

```
 _bind_field(field_name: str, field_obj: marshmallow.fields.Field) → None
```

```
 Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).
```

```
 Also set field load_only and dump_only values if field_name was specified in class Meta.
```

```
static _call_and_store(getter_func, data, *, field_name, error_store, index=None)
```

```
 Call getter_func with data as its argument, and store any ValidationErrors.
```

**Parameters**

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from data.
- **data** – The data passed to *getter\_func*.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

```
 _declared_fields = {'created_at': <fields.Timestamp(default=<marshmallow.missing>, at
```

```
 _default_error_messages = {'type': 'Invalid input type.', 'unknown': 'Unknown field.
```

```
 _deserialize(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, error_store: marsh-
 mallow.error_store.ErrorStore, many: bool = False, partial=False, unknown='raise',
 index=None) → Union[_T, List[_T]]
```

```
 Deserialize data.
```

**Parameters**

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if data should be deserialized as a collection.
- **partial** (*bool/tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**`_do_load`** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool = True*)

Deserialize *data*, returning the deserialized result. This method is private API.

**Parameters**

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

**`_has_processors`** (*tag*) → *bool*

**`_hooks`** = {}

**`_init_fields`** () → *None*

Update *self.fields*, *self.load\_fields*, and *self.dump\_fields* based on schema options. This method is private API.

**`_invoke_dump_processors`** (*tag: str, data, \*, many: bool, original\_data=None*)

**`_invoke_field_validators`** (*\*, error\_store: marshmallow.error\_store.ErrorStore, data, many: bool*)

**`_invoke_load_processors`** (*tag: str, data, \*, many: bool, original\_data, partial: Union[bool, Sequence[str], Set[str]]*)

**`_invoke_processors`** (*tag: str, \*, pass\_many: bool, data, many: bool, original\_data=None, \*\*kwargs*)

**`_invoke_schema_validators`** (*\*, error\_store: marshmallow.error\_store.ErrorStore, pass\_many: bool, data, original\_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field\_errors: bool = False*)

**\_normalize\_nested\_options** () → None

Apply then flatten nested schema options. This method is private API.

**\_run\_validator** (*validator\_func*, *output*, \*, *original\_data*, *error\_store*, *many*, *partial*, *pass\_original*, *index=None*)

**\_serialize** (*obj*: Union[\_T, Iterable[\_T]], \*, *many*: bool = False)

Serialize obj.

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from `marshal`.

#### **property dict\_class**

**dump** (*obj*: Any, \*, *many*: bool = None)

Serialize an object to native Python data types according to this Schema's fields.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dumps** (*obj*: Any, \**args*, *many*: bool = None, \*\**kwargs*)

Same as `dump()`, except return a JSON-encoded string.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], \*, *name*: str = 'GeneratedSchema') → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj: Any, attr: str, default: Any*)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

**handle\_error** (*error: marshmallow.exceptions.ValidationError, data: Any, \*, many: bool, \*\*kwargs*)

Custom error handler function for the schema.

#### Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj, many=<object object>, \*args, \*\*kwargs*)

Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to `False`, regardless of the value of *Schema.many*.

**load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial:*

*Union[bool, Sequence[str], Set[str]] = None, unknown: str = None*)

Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.

- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**loads** (*json\_data*: *str*, \*, *many*: *bool* = *None*, *partial*: *Union[bool, Sequence[str], Set[str]]* = *None*, *unknown*: *str* = *None*, \*\**kwargs*)  
Same as `load()`, except it takes a JSON string as input.

**Parameters**

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**on\_bind\_field** (*field\_name*: *str*, *field\_obj*: *marshmallow.fields.Field*) → *None*  
Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = `<marshmallow.schema.SchemaOpts object>`

**property set\_class**

**static valid\_request\_role** (*data*: *dict*) → *dict*

**validate** (*data*: *Mapping*, \*, *many*: *bool* = *None*, *partial*: *Union[bool, Sequence[str], Set[str]]* = *None*) → *Dict[str, List[str]]*  
Validate *data* against the schema, returning a dictionary of validation errors.

**Parameters**

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema.SearchSchema(*, only: Union[Sequence[str],
Set[str]] = None, exclude: Union[Sequence[str], Set[str]]
= (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (),
dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool,
Sequence[str], Set[str]] = False,
unknown: str = None)
```

#### class Meta

Options object for a Schema.

Example usage:

```
class Meta:
 fields = ("id", "email", "date_created")
 exclude = ("password", "secret_attribute")
```

Available options:

- **fields:** Tuple or list of fields to include in the serialized result.
- **additional:** Tuple or list of fields to include *in addition* to the explicitly declared fields. **additional** and **fields** are mutually-exclusive options.
- **include:** Dictionary of additional fields to include in the schema. It is usually better to define fields as class variables, but you may need to use this option, e.g., if your fields are Python keywords. May be an *OrderedDict*.
- **exclude:** Tuple or list of fields to exclude in the serialized result. Nested fields can be represented with dot delimiters.
- **dateformat:** Default format for *Date* <fields.Date> fields.
- **datetimeformat:** Default format for *DateTime* <fields.DateTime> fields.
- **render\_module:** Module to use for *loads* <Schema.loads> and *dumps* <Schema.dumps>. Defaults to *json* from the standard library.
- **ordered:** If *True*, order serialization output according to the order in which fields were declared. Output of *Schema.dump* will be a *collections.OrderedDict*.
- **index\_errors:** If *True*, errors dictionaries will include the **index** of invalid items in a collection.
- **load\_only:** Tuple or list of fields to exclude from serialized results.
- **dump\_only:** Tuple or list of fields to exclude from deserialization
- **unknown:** Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **register:** Whether to register the *Schema* with marshmallow's internal class registry. Must be *True* if you intend to refer to this *Schema* by class name in *Nested* fields. Only set this to *False* when memory usage is critical. Defaults to *True*.

#### OPTIONS\_CLASS

alias of `marshmallow.schema.SchemaOpts`

```
TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:
```

```
_bind_field(field_name: str, field_obj: marshmallow.fields.Field) → None
```

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field `load_only` and `dump_only` values if `field_name` was specified in `class Meta`.

```
static _call_and_store(getter_func, data, *, field_name, error_store, index=None)
```

Call `getter_func` with `data` as its argument, and store any *ValidationErrors*.

#### Parameters

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from `data`.
- **data** – The data passed to `getter_func`.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

```
_declared_fields = {'items_per_page': <fields.Integer(default=<marshmallow.missing>,
```

```
_default_error_messages = {'type': 'Invalid input type.', 'unknown': 'Unknown field.
```

```
_deserialize(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, error_store: marsh-
mallow.error_store.ErrorStore, many: bool = False, partial=False, unknown='raise',
index=None) → Union[_T, List[_T]]
```

Deserialize `data`.

#### Parameters

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if `data` should be deserialized as a collection.
- **partial** (*bool* | *tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

```
_do_load(data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], *, many: bool = None, par-
tial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool
= True)
```

Deserialize `data`, returning the deserialized result. This method is private API.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize `data` as a collection. If *None*, the value for `self.many` is used.

- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

```
_has_processors (tag) → bool
_hooks = {}
_init_fields () → None
 Update self.fields, self.load_fields, and self.dump_fields based on schema options. This method is private API.
_invoke_dump_processors (tag: str, data, *, many: bool, original_data=None)
_invoke_field_validators (*, error_store: marshmallow.error_store.ErrorStore, data, many: bool)
_invoke_load_processors (tag: str, data, *, many: bool, original_data, partial: Union[bool, Sequence[str], Set[str]])
_invoke_processors (tag: str, *, pass_many: bool, data, many: bool, original_data=None, **kwargs)
_invoke_schema_validators (*, error_store: marshmallow.error_store.ErrorStore, pass_many: bool, data, original_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field_errors: bool = False)
_normalize_nested_options () → None
 Apply then flatten nested schema options. This method is private API.
_run_validator (validator_func, output, *, original_data, error_store, many, partial, pass_original, index=None)
_serialize (obj: Union[_T, Iterable[_T]], *, many: bool = False)
 Serialize obj.
```

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from *marshal*.

#### property dict\_class

**dump** (obj: Any, \*, many: bool = None)  
Serialize an object to native Python data types according to this Schema's fields.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict



New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dumps** (*obj*: Any, \*args, many: bool = None, \*\*kwargs)  
Same as `dump()`, except return a JSON-encoded string.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], \*, *name*: str = 'GeneratedSchema') → type  
Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj*: Any, *attr*: str, *default*: Any)  
Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

**handle\_error** (*error*: marshmallow.exceptions.ValidationError, *data*: Any, \*, *many*: bool, \*\*kwargs)  
Custom error handler function for the schema.

#### Parameters

- **error** – The `ValidationError` raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj*, *many*=<object object>, \*args, \*\*kwargs)  
Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None)  
Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as *load()*, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**on\_bind\_field** (*field\_name*: str, *field\_obj*: *marshmallow.fields.Field*) → None

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**validate** (*data*: Mapping, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]

Validate *data* against the schema, returning a dictionary of validation errors.

#### Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to Nested fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema.Timestamp(*, default: Any = <marshmallow.missing>,
 missing: Any = <marshmallow.missing>,
 data_key: str = None, attribute: str = None,
 validate: Union[Callable[[Any], Any], Iterable[Callable[[Any], Any]]] = None,
 required: bool = False, allow_none: bool = None, load_only: bool = False,
 dump_only: bool = False, error_messages: Dict[str, str] = None,
 **metadata)
```

Field that serializes to timestamp integer and deserializes to a `datetime.datetime` class.

**\_CHECK\_ATTRIBUTE** = True

**\_bind\_to\_schema** (*field\_name*, *schema*)

Update field with values from its parent schema. Called by `Schema._bind_field`.

#### Parameters

- **field\_name** (*str*) – Field name set in schema.
- **schema** (*Schema*) – Parent schema.

**\_creation\_index** = 73

**\_deserialize** (*value*, *attr*, *data*, \*\*kwargs)

Deserialize value. Concrete `Field` classes should implement this method.

#### Parameters

- **value** – The value to be deserialized.
- **attr** – The attribute/key in *data* to be deserialized.
- **data** – The raw input data passed to the *Schema.load*.

- **kwargs** – Field-specific keyword arguments.

**Raises** **ValidationError** – In case of formatting or validation failure.

**Returns** The deserialized value.

Changed in version 2.0.0: Added `attr` and `data` parameters.

Changed in version 3.0.0: Added `**kwargs` to signature.

**\_serialize** (*value*, *attr*, *obj*, *\*\*kwargs*)

Serializes *value* to a basic Python datatype. Noop by default. Concrete `Field` classes should implement this method.

Example:

```
class TitleCase(Field):
 def _serialize(self, value, attr, obj, **kwargs):
 if not value:
 return ''
 return str(value).title()
```

#### Parameters

- **value** – The value to be serialized.
- **attr** (*str*) – The attribute or key on the object to be serialized.
- **obj** (*object*) – The object the value was pulled from.
- **kwargs** (*dict*) – Field-specific keyword arguments.

**Returns** The serialized value

**\_validate** (*value*)

Perform validation on *value*. Raise a `ValidationError` if validation does not succeed.

**\_validate\_missing** (*value*)

Validate missing values. Raise a `ValidationError` if *value* should be considered missing.

#### property context

The context dictionary for the parent `Schema`.

**default\_error\_messages** = {'null': 'Field may not be null.', 'required': 'Missing data for required field.'

**deserialize** (*value*: Any, *attr*: str = None, *data*: Mapping[str, Any] = None, *\*\*kwargs*)

Deserialize *value*.

#### Parameters

- **value** – The value to deserialize.
- **attr** – The attribute/key in *data* to deserialize.
- **data** – The raw input data passed to `Schema.load`.
- **kwargs** – Field-specific keyword arguments.

**Raises** **ValidationError** – If an invalid value is passed or if a required value is missing.

**fail** (*key*: str, *\*\*kwargs*)

Helper method that raises a `ValidationError` with an error message from `self.error_messages`.

Deprecated since version 3.0.0: Use `make_error` <marshmallow.fields.Field.make\_error> instead.

**get\_value** (*obj*, *attr*, *accessor*=None, *default*=<marshmallow.missing>)

Return the value for a given key from an object.

#### Parameters

- **obj** (*object*) – The object to get the value from.
- **attr** (*str*) – The attribute/key in *obj* to get the value from.
- **accessor** (*callable*) – A callable used to retrieve the value of *attr* from the object *obj*. Defaults to *marshmallow.utils.get\_value*.

**make\_error** (*key*: *str*, *\*\*kwargs*) → *marshmallow.exceptions.ValidationError*

Helper method to make a *ValidationError* with an error message from *self.error\_messages*.

**name** = None

**parent** = None

**property root**

Reference to the *Schema* that this field belongs to even if it is buried in a container field (e.g. *List*). Return *None* for unbound fields.

**serialize** (*attr*: *str*, *obj*: *Any*, *accessor*: *Callable*[[*Any*, *str*, *Any*], *Any*] = None, *\*\*kwargs*)

Pulls the value for the given key from the object, applies the field's formatting and returns the result.

#### Parameters

- **attr** – The attribute/key to get from the object.
- **obj** – The object to access the attribute/key from.
- **accessor** – Function used to access values from *obj*.
- **kwargs** – Field-specific keyword arguments.

```
class app.utils.marshmallow_schema.UserSchema(*, only: Union[Sequence[str], Set[str]]
 = None, exclude: Union[Sequence[str],
 Set[str]] = (), many: bool = False,
 context: Dict = None, load_only:
 Union[Sequence[str], Set[str]] = (),
 dump_only: Union[Sequence[str],
 Set[str]] = (), partial: Union[bool, Se-
 quence[str], Set[str]] = False, unknown:
 str = None)
```

**class Meta**

```
fields = ('id', 'name', 'last_name', 'email', 'genre', 'birth_date', 'active', 'cre
ordered = True
```

**OPTIONS\_CLASS**

alias of *marshmallow.schema.SchemaOpts*

```
TYPE_MAPPING = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:
```

**\_bind\_field** (*field\_name*: *str*, *field\_obj*: *marshmallow.fields.Field*) → None

Bind field to the schema, setting any necessary attributes on the field (e.g. *parent* and *name*).

Also set field *load\_only* and *dump\_only* values if *field\_name* was specified in *class Meta*.

**static \_call\_and\_store** (*getter\_func*, *data*, \*, *field\_name*, *error\_store*, *index*=None)

Call *getter\_func* with *data* as its argument, and store any *ValidationErrors*.

**Parameters**

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from data.
- **data** – The data passed to `getter_func`.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

**\_declared\_fields** = {'active': <fields.Boolean(default=<marshmallow.missing>, attribut

**\_default\_error\_messages** = {'type': 'Invalid input type.', 'unknown': 'Unknown field.

**\_deserialize** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, error\_store: marshmallow.error\_store.ErrorStore, many: bool = False, partial=False, unknown='raise', index=None*) → Union[\_T, List[\_T]]

Deserialize data.

**Parameters**

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if data should be deserialized as a collection.
- **partial** (*bool | tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**\_do\_load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool = True*)

Deserialize *data*, returning the deserialized result. This method is private API.

**Parameters**

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

**\_has\_processors** (*tag*) → bool

```

_hooks = {}

_init_fields() → None
 Update self.fields, self.load_fields, and self.dump_fields based on schema options. This method is private API.

_invoke_dump_processors(tag: str, data, *, many: bool, original_data=None)

_invoke_field_validators(*, error_store: marshmallow.error_store.ErrorStore, data, many: bool)

_invoke_load_processors(tag: str, data, *, many: bool, original_data, partial: Union[bool, Sequence[str], Set[str]])

_invoke_processors(tag: str, *, pass_many: bool, data, many: bool, original_data=None, **kwargs)

_invoke_schema_validators(*, error_store: marshmallow.error_store.ErrorStore, pass_many: bool, data, original_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field_errors: bool = False)

_normalize_nested_options() → None
 Apply then flatten nested schema options. This method is private API.

_run_validator(validator_func, output, *, original_data, error_store, many, partial, pass_original, index=None)

_serialize(obj: Union[_T, Iterable[_T]], *, many: bool = False)
 Serialize obj.

```

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from `marshal`.

#### property dict\_class

```
dump(obj: Any, *, many: bool = None)
 Serialize an object to native Python data types according to this Schema's fields.
```

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

```

dumps(obj: Any, *args, many: bool = None, **kwargs)
 Same as dump(), except return a JSON-encoded string.

```

#### Parameters

- **obj** – The object to serialize.

- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a (data, errors) tuple. A `ValidationError` is raised if *obj* is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields: Dict[str, Union[marshmallow.fields.Field, type]], \*, name: str = 'GeneratedSchema'*) → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj: Any, attr: str, default: Any*)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

**handle\_error** (*error: marshmallow.exceptions.ValidationError, data: Any, \*, many: bool, \*\*kwargs*)

Custom error handler function for the schema.

#### Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of *many* on dump or load.
- **partial** – Value of *partial* on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj, many=<object object>, \*args, \*\*kwargs*)

Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.



Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None*)  
 Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

**loads** (*json\_data: str, \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, \*\*kwargs*)  
 Same as *load()*, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

**on\_bind\_field** (*field\_name: str, field\_obj: marshmallow.fields.Field*) → None  
 Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**static valid\_request\_user** (*data: dict*) → dict

```
validate (data: Mapping, *, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]
```

Validate *data* against the schema, returning a dictionary of validation errors.

#### Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
validate_credentials (data: dict) → dict
```

```
static validate_email (data: dict) → dict
```

```
static validate_password (password: str) → None
```

```
class app.utils.marshmallow_schema._SearchOrderSchema (*, only: Union[Sequence[str], Set[str]] = None, exclude: Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only: Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str], Set[str]] = False, unknown: str = None)
```

#### **class Meta**

Options object for a Schema.

Example usage:

```
class Meta:
 fields = ("id", "email", "date_created")
 exclude = ("password", "secret_attribute")
```

Available options:

- **fields**: Tuple or list of fields to include in the serialized result.
- **additional**: Tuple or list of fields to include *in addition to the* explicitly declared fields. *additional* and *fields* are mutually-exclusive options.
- **include**: Dictionary of additional fields to include in the schema. It is usually better to define fields as class variables, but you may need to use this option, e.g., if your fields are Python keywords. May be an *OrderedDict*.
- **exclude**: Tuple or list of fields to exclude in the serialized result. Nested fields can be represented with dot delimiters.
- **dateformat**: Default format for *Date* <fields.Date> fields.

- **datetimeformat**: Default format for *DateTime* <*fields.DateTime*> fields.
- **render\_module**: Module to use for *loads* <*Schema.loads*> and *dumps* <*Schema.dumps*>. Defaults to *json* from the standard library.
- **ordered**: If *True*, order serialization output according to the order in which fields were declared. Output of *Schema.dump* will be a *collections.OrderedDict*.
- **index\_errors**: If *True*, errors dictionaries will include the *index* of invalid items in a collection.
- **load\_only**: Tuple or list of fields to exclude from serialized results.
- **dump\_only**: Tuple or list of fields to exclude from deserialization
- **unknown**: Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **register**: Whether to register the *Schema* with marshmallow's internal class registry. Must be *True* if you intend to refer to this *Schema* by class name in *Nested* fields. Only set this to *False* when memory usage is critical. Defaults to *True*.

**OPTIONS\_CLASS**

alias of `marshmallow.schema.SchemaOpts`

**TYPE\_MAPPING** = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:

**\_bind\_field** (*field\_name*: str, *field\_obj*: marshmallow.fields.Field) → None

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field *load\_only* and *dump\_only* values if *field\_name* was specified in class *Meta*.

**static \_call\_and\_store** (*getter\_func*, *data*, \*, *field\_name*, *error\_store*, *index*=None)

Call *getter\_func* with *data* as its argument, and store any *ValidationErrors*.

**Parameters**

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from *data*.
- **data** – The data passed to *getter\_func*.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise *None*.

**\_declared\_fields** = {'field\_name': <fields.String(default=<marshmallow.missing>, attri

**\_default\_error\_messages** = {'type': 'Invalid input type.', 'unknown': 'Unknown field.

**\_deserialize** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *error\_store*: marshmallow.error\_store.ErrorStore, *many*: bool = False, *partial*=False, *unknown*='raise', *index*=None) → Union[\_T, List[\_T]]

Deserialize data.

**Parameters**

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – *True* if *data* should be deserialized as a collection.
- **partial** (*bool* | *tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only

missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise *None*.

**Returns** A dictionary of the deserialized data.

**\_\_do\_load** (*data: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, many: bool = None, partial: Union[bool, Sequence[str], Set[str]] = None, unknown: str = None, postprocess: bool = True*)

Deserialize *data*, returning the deserialized result. This method is private API.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

**\_\_has\_processors** (*tag*) → bool

**\_\_hooks** = {}

**\_\_init\_fields** () → None

Update *self.fields*, *self.load\_fields*, and *self.dump\_fields* based on schema options. This method is private API.

**\_\_invoke\_dump\_processors** (*tag: str, data, \*, many: bool, original\_data=None*)

**\_\_invoke\_field\_validators** (*\*, error\_store: marshmallow.error\_store.ErrorStore, data, many: bool*)

**\_\_invoke\_load\_processors** (*tag: str, data, \*, many: bool, original\_data, partial: Union[bool, Sequence[str], Set[str]]*)

**\_\_invoke\_processors** (*tag: str, \*, pass\_many: bool, data, many: bool, original\_data=None, \*\*kwargs*)

**\_\_invoke\_schema\_validators** (*\*, error\_store: marshmallow.error\_store.ErrorStore, pass\_many: bool, data, original\_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field\_errors: bool = False*)

**\_\_normalize\_nested\_options** () → None

Apply then flatten nested schema options. This method is private API.

**\_\_run\_validator** (*validator\_func, output, \*, original\_data, error\_store, many, partial, pass\_original, index=None*)

**\_\_serialize** (*obj: Union[\_T, Iterable[\_T]], \*, many: bool = False*)

Serialize *obj*.

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from `marshal`.

#### **property dict\_class**

**dump** (*obj: Any, \*, many: bool = None*)

Serialize an object to native Python data types according to this Schema's fields.

#### **Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dump** (*obj: Any, \*args, many: bool = None, \*\*kwargs*)

Same as `dump()`, except return a JSON-encoded string.

#### **Parameters**

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if *obj* is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields: Dict[str, Union[marshmallow.fields.Field, type]], \*, name: str = 'GeneratedSchema'*) → type

Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### **Parameters**

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj*: Any, *attr*: str, *default*: Any)

Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of *obj* and *attr*.

**handle\_error** (*error*: *marshmallow.exceptions.ValidationError*, *data*: Any, \*, *many*: bool, *\*\*kwargs*)

Custom error handler function for the schema.

#### Parameters

- **error** – The *ValidationError* raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of *many* on dump or load.
- **partial** – Value of *partial* on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj*, *many*=<object object>, *\*args*, *\*\*kwargs*)

Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask.jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask.jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None)

Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to Nested fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as `load()`, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use `EXCLUDE`, `INCLUDE` or `RAISE`. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**on\_bind\_field** (*field\_name*: str, *field\_obj*: `marshmallow.fields.Field`) → None  
Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**validate** (*data*: Mapping, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]  
Validate *data* against the schema, returning a dictionary of validation errors.

#### Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

```
class app.utils.marshmallow_schema._SearchValueSchema(*, only: Union[Sequence[str],
 Set[str]] = None, exclude:
 Union[Sequence[str],
 Set[str]] = (), many:
 bool = False, context:
 Dict = None, load_only:
 Union[Sequence[str],
 Set[str]] = (), dump_only:
 Union[Sequence[str],
 Set[str]] = (), partial:
 Union[bool, Sequence[str],
 Set[str]] = False, unknown:
 str = None)
```

#### class Meta

Options object for a Schema.

Example usage:

```
class Meta:
 fields = ("id", "email", "date_created")
 exclude = ("password", "secret_attribute")
```

Available options:

- **fields:** Tuple or list of fields to include in the serialized result.
- **additional:** Tuple or list of fields to include *in addition* to the explicitly declared fields. **additional** and **fields** are mutually-exclusive options.
- **include:** Dictionary of additional fields to include in the schema. It is usually better to define fields as class variables, but you may need to use this option, e.g., if your fields are Python keywords. May be an *OrderedDict*.
- **exclude:** Tuple or list of fields to exclude in the serialized result. Nested fields can be represented with dot delimiters.
- **dateformat:** Default format for *Date* <fields.Date> fields.
- **datetimeformat:** Default format for *DateTime* <fields.DateTime> fields.
- **render\_module:** Module to use for *loads* <Schema.loads> and *dumps* <Schema.dumps>. Defaults to *json* from the standard library.
- **ordered:** If *True*, order serialization output according to the order in which fields were declared. Output of *Schema.dump* will be a *collections.OrderedDict*.
- **index\_errors:** If *True*, errors dictionaries will include the **index** of invalid items in a collection.
- **load\_only:** Tuple or list of fields to exclude from serialized results.
- **dump\_only:** Tuple or list of fields to exclude from deserialization
- **unknown:** Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*.
- **register:** Whether to register the *Schema* with marshmallow's internal class registry. Must be *True* if you intend to refer to this *Schema* by class name in *Nested* fields. Only set this to *False* when memory usage is critical. Defaults to *True*.



**OPTIONS\_CLASS**

alias of `marshmallow.schema.SchemaOpts`

**TYPE\_MAPPING** = {<class 'str'>: <class 'marshmallow.fields.String'>, <class 'bytes'>:

**\_bind\_field** (*field\_name*: str, *field\_obj*: marshmallow.fields.Field) → None

Bind field to the schema, setting any necessary attributes on the field (e.g. parent and name).

Also set field `load_only` and `dump_only` values if `field_name` was specified in class `Meta`.

**static \_call\_and\_store** (*getter\_func*, *data*, \*, *field\_name*, *error\_store*, *index*=None)

Call `getter_func` with `data` as its argument, and store any `ValidationErrors`.

**Parameters**

- **getter\_func** (*callable*) – Function for getting the serialized/deserialized value from `data`.
- **data** – The data passed to `getter_func`.
- **field\_name** (*str*) – Field name.
- **index** (*int*) – Index of the item being validated, if validating a collection, otherwise `None`.

**\_declared\_fields** = {'field\_name': <fields.String(default=<marshmallow.missing>, attri

**\_default\_error\_messages** = {'type': 'Invalid input type.', 'unknown': 'Unknown field.

**\_deserialize** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *error\_store*: marshmallow.error\_store.ErrorStore, *many*: bool = False, *partial*=False, *unknown*='raise', *index*=None) → Union[\_T, List[\_T]]

Deserialize `data`.

**Parameters**

- **data** (*dict*) – The data to deserialize.
- **error\_store** (*ErrorStore*) – Structure to store errors.
- **many** (*bool*) – `True` if `data` should be deserialized as a collection.
- **partial** (*bool*/*tuple*) – Whether to ignore missing fields and not require any fields declared. Propagates down to `Nested` fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use `EXCLUDE`, `INCLUDE` or `RAISE`.
- **index** (*int*) – Index of the item being serialized (for storing errors) if serializing a collection, otherwise `None`.

**Returns** A dictionary of the deserialized data.

**\_do\_load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, *postprocess*: bool = True)

Deserialize `data`, returning the deserialized result. This method is private API.

**Parameters**

- **data** – The data to deserialize.
- **many** – Whether to deserialize `data` as a collection. If `None`, the value for `self.many` is used.

- **partial** – Whether to validate required fields. If its value is an iterable, only fields listed in that iterable will be ignored will be allowed missing. If *True*, all fields will be allowed missing. If *None*, the value for *self.partial* is used.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.
- **postprocess** – Whether to run *post\_load* methods..

**Returns** Deserialized data

```
_has_processors (tag) → bool
_hooks = {}
_init_fields () → None
 Update self.fields, self.load_fields, and self.dump_fields based on schema options. This method is private API.
_invoke_dump_processors (tag: str, data, *, many: bool, original_data=None)
_invoke_field_validators (*, error_store: marshmallow.error_store.ErrorStore, data, many: bool)
_invoke_load_processors (tag: str, data, *, many: bool, original_data, partial: Union[bool, Sequence[str], Set[str]])
_invoke_processors (tag: str, *, pass_many: bool, data, many: bool, original_data=None, **kwargs)
_invoke_schema_validators (*, error_store: marshmallow.error_store.ErrorStore, pass_many: bool, data, original_data, many: bool, partial: Union[bool, Sequence[str], Set[str]], field_errors: bool = False)
_normalize_nested_options () → None
 Apply then flatten nested schema options. This method is private API.
_run_validator (validator_func, output, *, original_data, error_store, many, partial, pass_original, index=None)
_serialize (obj: Union[_T, Iterable[_T]], *, many: bool = False)
 Serialize obj.
```

#### Parameters

- **obj** – The object(s) to serialize.
- **many** (*bool*) – *True* if data should be serialized as a collection.

**Returns** A dictionary of the serialized data

Changed in version 1.0.0: Renamed from *marshal*.

#### property dict\_class

**dump** (obj: Any, \*, many: bool = None)  
Serialize an object to native Python data types according to this Schema's fields.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A dict of serialized data

**Return type** dict

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

Changed in version 3.0.0rc9: Validation no longer occurs upon serialization.

**dumps** (*obj*: Any, \*args, many: bool = None, \*\*kwargs)  
Same as `dump()`, except return a JSON-encoded string.

#### Parameters

- **obj** – The object to serialize.
- **many** – Whether to serialize *obj* as a collection. If *None*, the value for *self.many* is used.

**Returns** A json string

**Return type** str

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the serialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if `obj` is invalid.

**error\_messages** = {}

**classmethod from\_dict** (*fields*: Dict[str, Union[marshmallow.fields.Field, type]], \*, *name*: str = 'GeneratedSchema') → type  
Generate a *Schema* class given a dictionary of fields.

```
from marshmallow import Schema, fields

PersonSchema = Schema.from_dict({"name": fields.Str()})
print(PersonSchema().load({"name": "David"})) # => {'name': 'David'}
```

Generated schemas are not added to the class registry and therefore cannot be referred to by name in *Nested* fields.

#### Parameters

- **fields** (*dict*) – Dictionary mapping field names to field instances.
- **name** (*str*) – Optional name for the class, which will appear in the `repr` for the class.

New in version 3.0.0.

**get\_attribute** (*obj*: Any, *attr*: str, *default*: Any)  
Defines how to pull values from an object to serialize.

New in version 2.0.0.

Changed in version 3.0.0a1: Changed position of `obj` and `attr`.

**handle\_error** (*error*: marshmallow.exceptions.ValidationError, *data*: Any, \*, *many*: bool, \*\*kwargs)  
Custom error handler function for the schema.

#### Parameters

- **error** – The `ValidationError` raised during (de)serialization.
- **data** – The original input data.
- **many** – Value of `many` on dump or load.
- **partial** – Value of `partial` on load.

New in version 2.0.0.

Changed in version 3.0.0rc9: Receives *many* and *partial* (on deserialization) as keyword arguments.

**jsonify** (*obj*, *many*=<object object>, \*args, \*\*kwargs)  
Return a JSON response containing the serialized data.

#### Parameters

- **obj** – Object to serialize.
- **many** (*bool*) – Whether *obj* should be serialized as an instance or as a collection. If unset, defaults to the value of the *many* attribute on this Schema.
- **kwargs** – Additional keyword arguments passed to *flask jsonify*.

Changed in version 0.6.0: Takes the same arguments as *marshmallow.Schema.dump*. Additional keyword arguments are passed to *flask jsonify*.

Changed in version 0.6.3: The *many* argument for this method defaults to the value of the *many* attribute on the Schema. Previously, the *many* argument of this method defaulted to False, regardless of the value of *Schema.many*.

**load** (*data*: Union[Mapping[str, Any], Iterable[Mapping[str, Any]]], \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None)  
Deserialize a data structure to an object defined by this Schema's fields.

#### Parameters

- **data** – The data to deserialize.
- **many** – Whether to deserialize *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a (*data*, *errors*) tuple. A *ValidationError* is raised if invalid data are passed.

**loads** (*json\_data*: str, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None, *unknown*: str = None, \*\*kwargs)  
Same as *load()*, except it takes a JSON string as input.

#### Parameters

- **json\_data** – A JSON string of the data to deserialize.
- **many** – Whether to deserialize *obj* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to *Nested* fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.
- **unknown** – Whether to exclude, include, or raise an error for unknown fields in the data. Use *EXCLUDE*, *INCLUDE* or *RAISE*. If *None*, the value for *self.unknown* is used.

**Returns** Deserialized data

New in version 1.0.0.

Changed in version 3.0.0b7: This method returns the deserialized data rather than a `(data, errors)` tuple. A `ValidationError` is raised if invalid data are passed.

**on\_bind\_field** (*field\_name*: str, *field\_obj*: *marshmallow.fields.Field*) → None

Hook to modify a field when it is bound to the *Schema*.

No-op by default.

**opts** = <marshmallow.schema.SchemaOpts object>

**property set\_class**

**validate** (*data*: Mapping, \*, *many*: bool = None, *partial*: Union[bool, Sequence[str], Set[str]] = None) → Dict[str, List[str]]

Validate *data* against the schema, returning a dictionary of validation errors.

#### Parameters

- **data** – The data to validate.
- **many** – Whether to validate *data* as a collection. If *None*, the value for *self.many* is used.
- **partial** – Whether to ignore missing fields and not require any fields declared. Propagates down to Nested fields as well. If its value is an iterable, only missing fields listed in that iterable will be ignored. Use dot delimiters to specify nested fields.

**Returns** A dictionary of validation errors.

New in version 1.1.0.

## app.utils.swagger\_models

### Description

### Modules

---

[\*app.utils.swagger\\_models.auth\*](#)

---

[\*app.utils.swagger\\_models.document\*](#)

---

[\*app.utils.swagger\\_models.role\*](#)

---

[\*app.utils.swagger\\_models.user\*](#)

---

## app.utils.swagger\_models.auth

### Description

## app.utils.swagger\_models.document

### Description

## app.utils.swagger\_models.role

### Description

**app.utils.swagger\_models.user****Description****Functions**

---

`class_for_name(module_name, class_name)`

---

`create_search_query(db_model, query[, data])`

---

`find_longest_word(word_list)`

---

`get_request_query_fields(db_model[, ...])`

---

`ignore_keys(data, exclude)`

---

`pos_to_char(pos)`

---

`to_readable(obj)`

---

**app.utils.class\_for\_name**

`app.utils.class_for_name` (*module\_name: str, class\_name: str*) → any

**app.utils.create\_search\_query**

`app.utils.create_search_query` (*db\_model: Type[peewee.Model], query: peewee.ModelSelect, data: dict = None*) → peewee.ModelSelect

**app.utils.find\_longest\_word**

`app.utils.find_longest_word` (*word\_list: list*) → str

**app.utils.get\_request\_query\_fields**

`app.utils.get_request_query_fields` (*db\_model: Type[peewee.Model], request\_data=None*) → tuple

**app.utils.ignore\_keys**

`app.utils.ignore_keys` (*data: dict, exclude: list*) → dict

**app.utils.pos\_to\_char**

`app.utils.pos_to_char` (*pos: int*) → str

`app.utils.to_readable(obj: object) → object`

## Attributes

|                               |                                                                                                                                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>STRING_QUERY_OPERATORS</i> | REQUEST_QUERY_DELIMITER is used for converting requests field values to a list, for example: Request send these values: field_operator: contains field_values: valueA;valueB;valueC |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## app.utils.STRING\_QUERY\_OPERATORS

`app.utils.STRING_QUERY_OPERATORS` = ['eq', 'ne', 'contains', 'ncontains', 'startswith', 'endwith', 'in', 'notin', 'isnull', 'isnotnull', 'isgreater', 'isless', 'isgreaterorequal', 'islessoorequal']

`REQUEST_QUERY_DELIMITER` is used for converting requests field values to a list, for example:

**Request send these values:** field\_operator: contains field\_values: valueA;valueB;valueC

**The delimiter operator splits values to a list of values:** field\_values: [valueA, valueB, valueC]

## Exceptions

---

*FileEmptyError*

## app.utils.FileEmptyError

```
exception app.utils.FileEmptyError
```

```
exception app.utils.FileEmptyError
```

**args**

```
characters_written
```

errno

POSIX exception code

**filename**

exception filename

**filename2**

second exception filename

strerror

exception strerror

```
with traceback():
```

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

```
app.utils.STRING_QUERY_OPERATORS = ['eq', 'ne', 'contains', 'ncontains', 'startswith', 'endwith']
REQUEST_QUERY_DELIMITER is used for converting requests field values to a list, for example:
```

**Request send these values:** field\_operator: contains field\_values: valueA;valueB;valueC

**The delimiter operator splits values to a list of values:** `field_values: [valueA, valueB, valueC]`

`app.utils._build_clause_operators (field: peewee.Field, field_operator: str, field_value) → tuple`

`app.utils._build_order_by (db_model: Type[peewee.Model], request_data: dict) → list`

`app.utils._build_query_clause (field: peewee.Field, field_operator: str, field_value)`

`app.utils._build_string_clause (field: peewee.Field, field_operator: str, field_value) → tuple`

`app.utils.class_for_name (module_name: str, class_name: str) → any`

`app.utils.create_search_query (db_model: Type[peewee.Model], query: peewee.ModelSelect, data: dict = None) → peewee.ModelSelect`

`app.utils.find_longest_word (word_list: list) → str`

`app.utils.get_request_query_fields (db_model: Type[peewee.Model], request_data=None) → tuple`

`app.utils.ignore_keys (data: dict, exclude: list) → dict`

`app.utils.pos_to_char (pos: int) → str`

`app.utils.to_readable (obj: object) → object`

## Functions

---

|                                     |                                                           |
|-------------------------------------|-----------------------------------------------------------|
| <code>create_app(env_config)</code> | Builds an application based on environment configuration. |
|-------------------------------------|-----------------------------------------------------------|

---

### 2.1.7 app.create\_app

`app.create_app (env_config: str) → flask.app.Flask`  
Builds an application based on environment configuration.

**Parameters** `env_config` – Environment configuration.

**Returns** A `flask.flask` instance.

**Return type** Flask

#### Notes

Environment configuration values could be:

`config.ProdConfig  
config.DevConfig  
config.TestConfig`

`app._init_app (app: flask.app.Flask) → None`  
Call the method ‘`init_app`’ to register the extensions in the Flask object passed as parameter.

`app._init_logging (app: flask.app.Flask) → None`

`app._register_blueprints (app: flask.app.Flask) → None`

`app.create_app (env_config: str) → flask.app.Flask`  
Builds an application based on environment configuration.



**Parameters** `env_config` – Environment configuration.

**Returns** A *flask.flask* instance.

**Return type** Flask

### Notes

Environment configuration values could be:

```
config.ProdConfig
config.DevConfig
config.TestConfig
```

## 2.2 database

### Description

Package for managing the database.

The database package can creates and migrates tables and it can fills them with fake data.

### Modules

|                                  |                                     |
|----------------------------------|-------------------------------------|
| <code>database.factories</code>  | Package contains factories modules. |
| <code>database.migrations</code> |                                     |
| <code>database.seeds</code>      |                                     |

### 2.2.1 database.factories

#### Description

Package contains factories modules.

A factory is a database model filled with fake data. The factory purposes is creating records in a simple way.

The module is used in testing and seeds.

## References

The factory concept is based on [Laravel factories](#)

## Classes

---

|                                           |                                                        |
|-------------------------------------------|--------------------------------------------------------|
| <code>Factory(model_name, records)</code> | Class for managing factories based on database models. |
|-------------------------------------------|--------------------------------------------------------|

---

### database.factories.Factory

**class** database.factories.**Factory** (*model\_name: str, records: int = 1*)

Bases: object

Class for managing factories based on database models.

Create and save instances of database models or dicts based on database models registered in the application.

**make** (*self, params: dict = None, to\_dict: bool = False, exclude: list = None*)

Create instances of database models with fake data.

**save** (*self, params: dict = None*)

Save instances of database models in the database.

## Examples

How to create a fake user without save in database from command line:

```
source venv/bin/activate
flask shell
>>> user_factory = Factory('User')
>>> user = user_factory.make() # An instance of database model
<User: None>
>>> user.__data__ # You can see user data on this way
```

Oh, Wait!

```
>>> from pprint import pprint
>>> pprint(user.__data__) # Even better!
```

You can save the user in the database.

```
>>> user.save()
1
```

Factory can create a dictionary instead of an instance of database model.

```
>>> user = user_factory.make(to_dict=True)
>>> pprint(user)
```

Also can set params too.

```
>>> user_factory = Factory('User')
>>> user = user_factory.make({'name': 'Ruben', 'last_name': 'Rodriguez'})
>>> user.name
```

(continues on next page)

(continued from previous page)

```
'Ruben'
>>> user.last_name
'Rodriguez'
```

Factory allow to make many users in once time.

```
>>> user_factory = Factory('User', 3)
>>> users = user_factory.make()
[<User: None>, <User: None>, <User: None>]
```

If you want to fill some params later then you can pass a fieldnames list to the factory of thats fields that you don't want to fill yet.

```
>>> user_factory = Factory('User')
>>> user = user_factory.make(exclude=['name', 'birth_date'])
>>> user.name
None
>>> user.birth_date
None
```

If you only need to save data you can do it.

```
>>> Factory('User', 3).save()
[<User: 1>, <User: 2>, <User: 3>]
```

And you can set params for all users.

```
>>> Factory('User', 3).save({'name': 'Ruben'})
[<User: 4>, <User: 5>, <User: 6>]
```

## Methods

|                                                       |                                                    |
|-------------------------------------------------------|----------------------------------------------------|
| <code>Factory.__init__(model_name[, records])</code>  | Register as many factories as given records.       |
| <code>Factory.make([params, to_dict, exclude])</code> | Create instances of database model with fake data. |
| <code>Factory.save([params])</code>                   | Save instances of database model in the database.  |

### database.factories.Factory.\_\_init\_\_

`Factory.__init__(model_name: str, records: int = 1)`  
 Register as many factories as given records.

**database.factories.Factory.make**

`Factory.make` (*params: dict = None, to\_dict: bool = False, exclude: list = None*) → any  
 Create instances of database model with fake data.

**Parameters**

- **params** (*dict*) – Params to set when an instance of database model is created.
- **to\_dict** (*bool*) – If is True returns a dict otherwise is an instance of database model. By default is False.
- **exclude** (*list*) – Params are not going to be filled. These fields are equals to None.

**Returns** Could be a dict, a list or an instance of database model.

**Return type** any

**database.factories.Factory.save**

`Factory.save` (*params: dict = None*) → any  
 Save instances of database model in the database.

**Parameters** **params** (*dict*) – Params to set when an instance of database model is created.

**Returns** Could be a list or an instance of database model.

**Return type** any

**class** `database.factories.Factory` (*model\_name: str, records: int = 1*)

Class for managing factories based on database models.

Create and save instances of database models or dicts based on database models registered in the application.

**make** (*self, params: dict = None, to\_dict: bool = False, exclude: list = None*)

Create instances of database models with fake data.

**save** (*self, params: dict = None*)

Save instances of database models in the database.

**Examples**

How to create a fake user without save in database from command line:

```
source venv/bin/activate
flask shell
>>> user_factory = Factory('User')
>>> user = user_factory.make() # An instance of database model
<User: None>
>>> user.__data__ # You can see user data on this way
```

Oh, Wait!

```
>>> from pprint import pprint
>>> pprint(user.__data__) # Even better!
```

You can save the user in the database.

```
>>> user.save()
1
```

Factory can create a dictionary instead of an instance of database model.

```
>>> user = user_factory.make(to_dict=True)
>>> pprint(user)
```

Also can set params too.

```
>>> user_factory = Factory('User')
>>> user = user_factory.make({'name': 'Ruben', 'last_name': 'Rodriguez'})
>>> user.name
'Ruben'
>>> user.last_name
'Rodriguez'
```

Factory allow to make many users in once time.

```
>>> user_factory = Factory('User', 3)
>>> users = user_factory.make()
[<User: None>, <User: None>, <User: None>]
```

If you want to fill some params later then you can pass a fieldnames list to the factory of thats fields that you don't want to fill yet.

```
>>> user_factory = Factory('User')
>>> user = user_factory.make(exclude=['name', 'birth_date'])
>>> user.name
None
>>> user.birth_date
None
```

If you only need to save data you can do it.

```
>>> Factory('User', 3).save()
[<User: 1>, <User: 2>, <User: 3>]
```

And you can set params for all users.

```
>>> Factory('User', 3).save({'name': 'Ruben'})
[<User: 4>, <User: 5>, <User: 6>]
```

**make** (*params: dict = None, to\_dict: bool = False, exclude: list = None*) → any  
Create instances of database model with fake data.

#### Parameters

- **params** (*dict*) – Params to set when an instance of database model is created.
- **to\_dict** (*bool*) – If is True returns a dict otherwise is an instance of database model. By default is False.
- **exclude** (*list*) – Params are not going to be filled. These fields are equals to None.

**Returns** Could be a dict, a list or an instance of database model.

**Return type** any

**save** (*params: dict = None*) → any  
Save instances of database model in the database.

**Parameters** **params** (*dict*) – Params to set when an instance of database model is created.

**Returns** Could be a list or an instance of database model.

**Return type** any

## 2.2.2 database.migrations

### Description

### Modules

---

```
database.migrations.
aaa_add_genre_column_on_user_table
database.migrations.
aab_add_created_by_column_on_user_table
database.migrations.
aac_create_documents_table
database.migrations.
aad_create_user_roles_table
database.migrations.
aaf_remove_role_slug_column
```

---

### database.migrations.aaa\_add\_genre\_column\_on\_user\_table

### Description

### Classes

---

```
AddGenreColumnOnUserTable()
```

---

### database.migrations.aaa\_add\_genre\_column\_on\_user\_table.AddGenreColumnOnUserTable

**class** database.migrations.aaa\_add\_genre\_column\_on\_user\_table.**AddGenreColumnOnUserTable**  
Bases: object

### Methods

---

|                                               |                  |
|-----------------------------------------------|------------------|
| <code>AddGenreColumnOnUserTable.</code>       | Initialize self. |
| <code>__init__()</code>                       |                  |
| <code>AddGenreColumnOnUserTable.down()</code> |                  |
| <code>AddGenreColumnOnUserTable.up()</code>   |                  |

---

**database.migrations.aaa\_add\_genre\_column\_on\_user\_table.AddGenreColumnOnUserTable.\_\_init\_\_**

**AddGenreColumnOnUserTable.\_\_init\_\_()**  
Initialize self. See help(type(self)) for accurate signature.

**database.migrations.aaa\_add\_genre\_column\_on\_user\_table.AddGenreColumnOnUserTable.down**

**AddGenreColumnOnUserTable.down()**

**database.migrations.aaa\_add\_genre\_column\_on\_user\_table.AddGenreColumnOnUserTable.up**

**AddGenreColumnOnUserTable.up()**

**class** database.migrations.aaa\_add\_genre\_column\_on\_user\_table.**AddGenreColumnOnUserTable**

**\_exists\_column()** → bool

**down()**

**up()**

**database.migrations.aab\_add\_created\_by\_column\_on\_user\_table**

## Description

## Classes

---

*AddCreatedByColumnOnUserTable()*

---

**database.migrations.aab\_add\_created\_by\_column\_on\_user\_table.AddCreatedByColumnOnUserTable**

**class** database.migrations.aab\_add\_created\_by\_column\_on\_user\_table.**AddCreatedByColumnOnUserTable**  
Bases: object

## Methods

---

*AddCreatedByColumnOnUserTable.\_\_init\_\_()* Initialize self.

---

*AddCreatedByColumnOnUserTable.down()*

---

*AddCreatedByColumnOnUserTable.up()*

---

`database.migrations.aab_add_created_by_column_on_user_table.AddCreatedByColumnOnUserTable.`

`AddCreatedByColumnOnUserTable.__init__()`  
Initialize self. See `help(type(self))` for accurate signature.

`database.migrations.aab_add_created_by_column_on_user_table.AddCreatedByColumnOnUserTable.d`

`AddCreatedByColumnOnUserTable.down()`

`database.migrations.aab_add_created_by_column_on_user_table.AddCreatedByColumnOnUserTable.up`

`AddCreatedByColumnOnUserTable.up()`

**class** `database.migrations.aab_add_created_by_column_on_user_table.AddCreatedByColumnOnUserT`

`_exists_column()` → bool

`down()`

`up()`

`database.migrations.aac_create_documents_table`

## Description

## Classes

---

`CreateDocumentsTable()`

---

`database.migrations.aac_create_documents_table.CreateDocumentsTable`

**class** `database.migrations.aac_create_documents_table.CreateDocumentsTable`  
Bases: object

## Methods

---

`CreateDocumentsTable.__init__()` Initialize self.

---

`CreateDocumentsTable.down()`

---

`CreateDocumentsTable.up()`

---



**database.migrations.aac\_create\_documents\_table.CreateDocumentsTable.\_\_init\_\_**

CreateDocumentsTable.\_\_init\_\_()

Initialize self. See help(type(self)) for accurate signature.

**database.migrations.aac\_create\_documents\_table.CreateDocumentsTable.down**

CreateDocumentsTable.down()

**database.migrations.aac\_create\_documents\_table.CreateDocumentsTable.up**

CreateDocumentsTable.up()

**class** database.migrations.aac\_create\_documents\_table.CreateDocumentsTable

**\_exists\_table**() → bool

**down**()

**up**()

**database.migrations.aad\_create\_user\_roles\_table****Description****Classes**


---

*CreateUserRolesTable()*

---

**database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable**

**class** database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable

Bases: object

**Methods**


---

*CreateUserRolesTable.\_\_init\_\_()*      Initialize self.

---

*CreateUserRolesTable.down()*

---

*CreateUserRolesTable.up()*

---

**database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable.\_\_init\_\_**

CreateUserRolesTable.\_\_init\_\_()   
 Initialize self. See help(type(self)) for accurate signature.

**database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable.down**

CreateUserRolesTable.down()

**database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable.up**

CreateUserRolesTable.up()

**class** database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable

**static** \_add\_foreign\_key\_constraint\_users\_table() → None   
 [https://www.sqlite.org/lang\\_altertable.html](https://www.sqlite.org/lang_altertable.html)

**static** \_drop\_foreign\_key\_constraint\_users\_table() → list   
 [https://www.sqlite.org/lang\\_altertable.html](https://www.sqlite.org/lang_altertable.html)

\_exists\_table() → bool

down()

up()

**class** database.migrations.aad\_create\_user\_roles\_table.\_OldUser(\*args,   
 \*\*kwargs)

**DoesNotExist**   
 alias of \_OldUserDoesNotExist

\_coerce = True

\_meta = <peewee.Metadata object>

**classmethod** \_normalize\_data(data, kwargs)

**property** \_pk

\_pk\_expr()

\_populate\_unsaved\_relations(field\_dict)

\_prune\_fields(field\_dict, only)

\_schema = <peewee.SchemaManager object>

**active** = <BooleanField: \_OldUser.active>

**classmethod** add\_index(\*fields, \*\*kwargs)

**classmethod** alias(alias=None)

**classmethod** bind(database, bind\_refs=True, bind\_backrefs=True)

**classmethod** bind\_ctx(database, bind\_refs=True, bind\_backrefs=True)

**birth\_date** = <DateField: \_OldUser.birth\_date>

**classmethod** bulk\_create(model\_list, batch\_size=None)

```

classmethod bulk_update(model_list, fields, batch_size=None)
children
clone()
coerce(_coerce=True)
static copy(method)
classmethod create(**query)
classmethod create_table(safe=True, **options)
created_at = <TimestampField: _OldUser.created_at>
created_by = <ForeignKeyField: _OldUser.created_by>
created_by_id = <ForeignKeyField: _OldUser.created_by>
classmethod delete()
classmethod delete_by_id(pk)
delete_instance(recursive=False, delete_nullable=False)
deleted_at = <TimestampField: _OldUser.deleted_at>
dependencies(search_nullable=False)
property dirty_fields
classmethod drop_table(safe=True, drop_sequences=True, **options)
email = <CharField: _OldUser.email>
classmethod filter(*dq_nodes, **filters)
genre = <FixedCharField: _OldUser.genre>
classmethod get(*query, **filters)
classmethod get_by_id(pk)
classmethod get_fields(exclude: list = None, include: list = None, sort_order: list = None) →
 set
get_id()
classmethod get_or_create(**kwargs)
classmethod get_or_none(*query, **filters)
id = <AutoField: _OldUser.id>
classmethod index(*fields, **kwargs)
classmethod insert(_Model__data=None, **insert)
classmethod insert_from(query, fields)
classmethod insert_many(rows, fields=None)
property is_active
is_alias()
property is_anonymous
property is_authenticated
is_dirty()

```

```
last_name = <CharField: _OldUser.last_name>
name = <CharField: _OldUser.name>
classmethod noop()
password = <CharField: _OldUser.password>
classmethod raw(sql, *params)
classmethod replace(_Model__data=None, **insert)
classmethod replace_many(rows, fields=None)
role = <ForeignKeyField: _OldUser.role>
role_id = <ForeignKeyField: _OldUser.role>
abstract save(*args: list, **kwargs: dict) → int
classmethod select(*fields)
classmethod set_by_id(key, value)
classmethod table_exists()
classmethod truncate_table(**options)
unwrap()
classmethod update(_Model__data=None, **update)
updated_at = <TimestampField: _OldUser.updated_at>
classmethod validate_model()
```

## database.migrations.aaf\_remove\_role\_slug\_column

### Description

### Classes

---

*RemoveRoleSlugColumn()*

---

## database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlugColumn

```
class database.migrations.aaf_remove_role_slug_column.RemoveRoleSlugColumn
 Bases: object
```

### Methods

---

*RemoveRoleSlugColumn.\_\_init\_\_()*    Initialize self.

---

*RemoveRoleSlugColumn.down()*

---

*RemoveRoleSlugColumn.up()*

---

**database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlugColumn.\_\_init\_\_**

**RemoveRoleSlugColumn.\_\_init\_\_()**  
Initialize self. See help(type(self)) for accurate signature.

**database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlugColumn.down**

**RemoveRoleSlugColumn.down()**

**database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlugColumn.up**

**RemoveRoleSlugColumn.up()**

**class** database.migrations.aaf\_remove\_role\_slug\_column.**RemoveRoleSlugColumn**

**static** **\_add\_unique\_constraint\_roles\_table()** → None  
[https://www.sqlite.org/lang\\_altertable.html](https://www.sqlite.org/lang_altertable.html)

**static** **\_drop\_unique\_constraint\_roles\_table()** → None  
[https://www.sqlite.org/lang\\_altertable.html](https://www.sqlite.org/lang_altertable.html)

**\_exists\_column()** → bool

**down()**

**up()**

**class** database.migrations.aaf\_remove\_role\_slug\_column.**\_OldRole**(\*args,  
\*\*kwargs)

**DoesNotExist**  
alias of **\_OldRoleDoesNotExist**

**\_coerce** = **True**

**\_meta** = <peewee.Metadata object>

**classmethod** **\_normalize\_data**(data, kwargs)

**property** **\_pk**

**\_pk\_expr()**

**\_populate\_unsaved\_relations**(field\_dict)

**\_prune\_fields**(field\_dict, only)

**\_schema** = <peewee.SchemaManager object>

**classmethod** **add\_index**(\*fields, \*\*kwargs)

**add\_permissions**(permissions)  
Add one or more permissions to role.

**Parameters** **permissions** – a set, list, or single string.

Caller must commit to DB.

New in version 3.3.0.

Deprecated since version 3.4.4: Use **UserDatastore.remove\_permissions\_from\_role()**

```
classmethod alias (alias=None)
classmethod bind (database, bind_refs=True, bind_backrefs=True)
classmethod bind_ctx (database, bind_refs=True, bind_backrefs=True)
classmethod bulk_create (model_list, batch_size=None)
classmethod bulk_update (model_list, fields, batch_size=None)
clone ()
coerce (_coerce=True)
static copy (method)
classmethod create (**query)
classmethod create_table (safe=True, **options)
created_at = <TimestampField: _OldRole.created_at>
classmethod delete ()
classmethod delete_by_id (pk)
delete_instance (recursive=False, delete_nullable=False)
deleted_at = <TimestampField: _OldRole.deleted_at>
dependencies (search_nullable=False)
description = <TextField: _OldRole.description>
property dirty_fields
classmethod drop_table (safe=True, drop_sequences=True, **options)
classmethod filter (*dq_nodes, **filters)
classmethod get (*query, **filters)
classmethod get_by_id (pk)
classmethod get_fields (exclude: list = None, include: list = None, sort_order: list = None) →
 set
get_id ()
classmethod get_or_create (**kwargs)
classmethod get_or_none (*query, **filters)
get_permissions ()
 Return set of permissions associated with role.

 Either takes a comma separated string of permissions or an interable of strings if permissions are in their
 own table.

 New in version 3.3.0.
id = <AutoField: _OldRole.id>
classmethod index (*fields, **kwargs)
classmethod insert (_Model__data=None, **insert)
classmethod insert_from (query, fields)
classmethod insert_many (rows, fields=None)
```

```

is_alias()
is_dirty()
name = <CharField: _OldRole.name>
classmethod noop()
classmethod raw(sql, *params)
remove_permissions(permissions)
 Remove one or more permissions from role.

 Parameters permissions – a set, list, or single string.

 Caller must commit to DB.

 New in version 3.3.0.

 Deprecated since version 3.4.4: Use UserDatastore.remove_permissions_from_role()
classmethod replace(_Model__data=None, **insert)
classmethod replace_many(rows, fields=None)
abstract save(*args: list, **kwargs: dict) → int
classmethod select(*fields)
classmethod set_by_id(key, value)
slug = <CharField: _OldRole.slug>
classmethod table_exists()
classmethod truncate_table(**options)
unwrap()
classmethod update(_Model__data=None, **update)
updated_at = <TimestampField: _OldRole.updated_at>
classmethod validate_model()

```

## Classes

---

*Migration*(\*args, \*\*kwargs)

---

### database.migrations.Migration

```

class database.migrations.Migration(*args, **kwargs)
 Bases: playhouse.flask_utils.FlaskDB.get_model_class.<locals>.BaseModel

```

## Attributes

---

*Migration.dirty\_fields*

---

*Migration.id*

---

*Migration.name*

---

**database.migrations.Migration.dirty\_fields**

**property** `Migration.dirty_fields`

**database.migrations.Migration.id**

`Migration.id = <AutoField: BaseModel.id>`

**database.migrations.Migration.name**

`Migration.name = <CharField: Migration.name>`

## Methods

---

*Migration.\_\_init\_\_*(\*args, \*\*kwargs) Initialize self.

---

*Migration.add\_index*(\*fields, \*\*kwargs)

---

*Migration.alias*([alias])

---

*Migration.bind*(database[, bind\_refs, ...])

---

*Migration.bind\_ctx*(database[, bind\_refs, ...])

---

*Migration.bulk\_create*(model\_list[, batch\_size])

---

*Migration.bulk\_update*(model\_list, fields[, ...])

---

*Migration.clone*()

---

*Migration.coerce*([\_coerce])

---

*Migration.copy*(method)

---

*Migration.create*(\*\*query)

---

*Migration.create\_table*([safe])

---

*Migration.delete*()

---

*Migration.delete\_by\_id*(pk)

---

*Migration.delete\_instance*([recursive, ...])

---

*Migration.dependencies*([search\_nullable])

---

*Migration.drop\_table*([safe, drop\_sequences])

---

*Migration.filter*(\*dq\_nodes, \*\*filters)

---

*Migration.get*(\*query, \*\*filters)

---

continues on next page



Table 125 – continued from previous page

|                                                 |
|-------------------------------------------------|
| <i>Migration.get_by_id(pk)</i>                  |
| <i>Migration.get_id()</i>                       |
| <i>Migration.get_or_create(**kwargs)</i>        |
| <i>Migration.get_or_none(*query, **filters)</i> |
| <i>Migration.index(*fields, **kwargs)</i>       |
| <i>Migration.insert([_Model__data])</i>         |
| <i>Migration.insert_from(query, fields)</i>     |
| <i>Migration.insert_many(rows[, fields])</i>    |
| <i>Migration.is_alias()</i>                     |
| <i>Migration.is_dirty()</i>                     |
| <i>Migration.noop()</i>                         |
| <i>Migration.raw(sql, *params)</i>              |
| <i>Migration.replace([_Model__data])</i>        |
| <i>Migration.replace_many(rows[, fields])</i>   |
| <i>Migration.save([force_insert, only])</i>     |
| <i>Migration.select(*fields)</i>                |
| <i>Migration.set_by_id(key, value)</i>          |
| <i>Migration.table_exists()</i>                 |
| <i>Migration.truncate_table(**options)</i>      |
| <i>Migration.unwrap()</i>                       |
| <i>Migration.update([_Model__data])</i>         |
| <i>Migration.validate_model()</i>               |

**database.migrations.Migration.\_\_init\_\_**

**Migration.\_\_init\_\_** (\*args, \*\*kwargs)

Initialize self. See help(type(self)) for accurate signature.

**database.migrations.Migration.add\_index**

**classmethod** **Migration.add\_index** (\*fields, \*\*kwargs)

**database.migrations.Migration.alias****classmethod** Migration.**alias** (*alias=None*)**database.migrations.Migration.bind****classmethod** Migration.**bind** (*database, bind\_refs=True, bind\_backrefs=True*)**database.migrations.Migration.bind\_ctx****classmethod** Migration.**bind\_ctx** (*database, bind\_refs=True, bind\_backrefs=True*)**database.migrations.Migration.bulk\_create****classmethod** Migration.**bulk\_create** (*model\_list, batch\_size=None*)**database.migrations.Migration.bulk\_update****classmethod** Migration.**bulk\_update** (*model\_list, fields, batch\_size=None*)**database.migrations.Migration.clone**Migration.**clone**()**database.migrations.Migration.coerce**Migration.**coerce** (*\_coerce=True*)**database.migrations.Migration.copy****static** Migration.**copy** (*method*)**database.migrations.Migration.create****classmethod** Migration.**create** (*\*\*query*)

**database.migrations.Migration.create\_table**

```
classmethod Migration.create_table (safe=True, **options)
```

**database.migrations.Migration.delete**

```
classmethod Migration.delete ()
```

**database.migrations.Migration.delete\_by\_id**

```
classmethod Migration.delete_by_id (pk)
```

**database.migrations.Migration.delete\_instance**

```
Migration.delete_instance (recursive=False, delete_nullable=False)
```

**database.migrations.Migration.dependencies**

```
Migration.dependencies (search_nullable=False)
```

**database.migrations.Migration.drop\_table**

```
classmethod Migration.drop_table (safe=True, drop_sequences=True, **options)
```

**database.migrations.Migration.filter**

```
classmethod Migration.filter (*dq_nodes, **filters)
```

**database.migrations.Migration.get**

```
classmethod Migration.get (*query, **filters)
```

**database.migrations.Migration.get\_by\_id**

```
classmethod Migration.get_by_id (pk)
```

**database.migrations.Migration.get\_id**

`Migration.get_id()`

**database.migrations.Migration.get\_or\_create**

**classmethod** `Migration.get_or_create(**kwargs)`

**database.migrations.Migration.get\_or\_none**

**classmethod** `Migration.get_or_none(*query, **filters)`

**database.migrations.Migration.index**

**classmethod** `Migration.index(*fields, **kwargs)`

**database.migrations.Migration.insert**

**classmethod** `Migration.insert(_Model__data=None, **insert)`

**database.migrations.Migration.insert\_from**

**classmethod** `Migration.insert_from(query, fields)`

**database.migrations.Migration.insert\_many**

**classmethod** `Migration.insert_many(rows, fields=None)`

**database.migrations.Migration.is\_alias**

`Migration.is_alias()`

**database.migrations.Migration.is\_dirty**

`Migration.is_dirty()`

**database.migrations.Migration.noop****classmethod** Migration.noop()**database.migrations.Migration.raw****classmethod** Migration.raw(sql, \*params)**database.migrations.Migration.replace****classmethod** Migration.replace(\_Model\_\_data=None, \*\*insert)**database.migrations.Migration.replace\_many****classmethod** Migration.replace\_many(rows, fields=None)**database.migrations.Migration.save**

Migration.save(force\_insert=False, only=None)

**database.migrations.Migration.select****classmethod** Migration.select(\*fields)**database.migrations.Migration.set\_by\_id****classmethod** Migration.set\_by\_id(key, value)**database.migrations.Migration.table\_exists****classmethod** Migration.table\_exists()**database.migrations.Migration.truncate\_table****classmethod** Migration.truncate\_table(\*\*options)

**database.migrations.Migration.unwrap**`Migration.unwrap()`**database.migrations.Migration.update**`classmethod Migration.update(_Model__data=None, **update)`**database.migrations.Migration.validate\_model**`classmethod Migration.validate_model()`**Functions**

---

`get_migration_names()`

---

`init_migrations([rollback])`

---

`migrate_actions(fnc)`

---

`rollback_actions(fnc)`

---

**database.migrations.get\_migration\_names**`database.migrations.get_migration_names() → list`**database.migrations.init\_migrations**`database.migrations.init_migrations(rollback: bool = False) → None`**database.migrations.migrate\_actions**`database.migrations.migrate_actions(fnc)`**database.migrations.rollback\_actions**`database.migrations.rollback_actions(fnc)``class database.migrations.Migration(*args, **kwargs)``DoesNotExist``alias of MigrationDoesNotExist``_coerce = True``_meta = <peewee.Metadata object>``classmethod _normalize_data(data, kwargs)``property _pk``_pk_expr()`

```

 _populate_unsaved_relations (field_dict)
 _prune_fields (field_dict, only)
 _schema = <peewee.SchemaManager object>
 classmethod add_index (*fields, **kwargs)
 classmethod alias (alias=None)
 classmethod bind (database, bind_refs=True, bind_backrefs=True)
 classmethod bind_ctx (database, bind_refs=True, bind_backrefs=True)
 classmethod bulk_create (model_list, batch_size=None)
 classmethod bulk_update (model_list, fields, batch_size=None)
 clone ()
 coerce (_coerce=True)
 static copy (method)
 classmethod create (**query)
 classmethod create_table (safe=True, **options)
 classmethod delete ()
 classmethod delete_by_id (pk)
 delete_instance (recursive=False, delete_nullable=False)
 dependencies (search_nullable=False)
 property dirty_fields
 classmethod drop_table (safe=True, drop_sequences=True, **options)
 classmethod filter (*dq_nodes, **filters)
 classmethod get (*query, **filters)
 classmethod get_by_id (pk)
 get_id ()
 classmethod get_or_create (**kwargs)
 classmethod get_or_none (*query, **filters)
 id = <AutoField: BaseModel.id>
 classmethod index (*fields, **kwargs)
 classmethod insert (_Model__data=None, **insert)
 classmethod insert_from (query, fields)
 classmethod insert_many (rows, fields=None)
 is_alias ()
 is_dirty ()
 name = <CharField: Migration.name>
 classmethod noop ()
 classmethod raw (sql, *params)

```

```
classmethod replace (_Model__data=None, **insert)
classmethod replace_many (rows, fields=None)
save (force_insert=False, only=None)
classmethod select (*fields)
classmethod set_by_id (key, value)
classmethod table_exists ()
classmethod truncate_table (**options)
unwrap ()
classmethod update (_Model__data=None, **update)
classmethod validate_model ()
```

database.migrations.**get\_migration\_names** () → list

database.migrations.**init\_migrations** (rollback: bool = False) → None

database.migrations.**migrate\_actions** (fnc)

database.migrations.**rollback\_actions** (fnc)

## 2.2.3 database.seeds

### Description

### Modules

---

*database.seeds.document\_seeder*

---

*database.seeds.role\_seeder*

---

*database.seeds.user\_seeder*

---

### database.seeds.document\_seeder

### Description

### Classes

---

*DocumentSeeder*(\*\*kwargs)

---



**database.seeds.document\_seeder.DocumentSeeder**

```
class database.seeds.document_seeder.DocumentSeeder (**kwargs)
 Bases: object
```

**Attributes**


---

*DocumentSeeder.name*

---

**database.seeds.document\_seeder.DocumentSeeder.name**

DocumentSeeder.name = 'DocumentSeeder'

**Methods**


---

*DocumentSeeder.\_\_init\_\_*([rows])      Initialize self.

---

**database.seeds.document\_seeder.DocumentSeeder.\_\_init\_\_**

DocumentSeeder.\_\_init\_\_(rows: int = 30)  
 Initialize self. See help(type(self)) for accurate signature.

```
class database.seeds.document_seeder.DocumentSeeder (**kwargs)

 name = 'DocumentSeeder'
```

**database.seeds.role\_seeder****Description****Classes**


---

*RoleSeeder*(\*\*kwargs)

---

**database.seeds.role\_seeder.RoleSeeder**

```
class database.seeds.role_seeder.RoleSeeder (**kwargs)
 Bases: object
```

## Attributes

---

*RoleSeeder.name*

---

**database.seeds.role\_seeder.RoleSeeder.name**

`RoleSeeder.name = 'RoleSeeder'`

## Methods

---

*RoleSeeder.\_\_init\_\_()* Initialize self.

---

**database.seeds.role\_seeder.RoleSeeder.\_\_init\_\_**

`RoleSeeder.__init__()`  
Initialize self. See `help(type(self))` for accurate signature.

**class** `database.seeds.role_seeder.RoleSeeder(**kwargs)`

**static** `_create_admin_role()` → None

**static** `_create_team_leader()` → None

**static** `_create_worker_role()` → None

**name** = `'RoleSeeder'`

**database.seeds.user\_seeder**

## Description

## Classes

---

*UserSeeder(\*\*kwargs)*

---

**database.seeds.user\_seeder.UserSeeder**

**class** `database.seeds.user_seeder.UserSeeder(**kwargs)`  
Bases: `object`

## Attributes

---

*UserSeeder.name*

---

### database.seeds.user\_seeder.UserSeeder.name

UserSeeder.name = 'UserSeeder'

## Methods

---

*UserSeeder.\_\_init\_\_([rows])*      Initialize self.

---

### database.seeds.user\_seeder.UserSeeder.\_\_init\_\_

UserSeeder.\_\_init\_\_(rows: int = 30)

Initialize self. See help(type(self)) for accurate signature.

**class** database.seeds.user\_seeder.UserSeeder(\*\*kwargs)

**static** \_create\_admin\_user()

name = 'UserSeeder'

## Functions

---

*get\_seeders()*

---

*init\_seed()*

---

### database.seeds.get\_seeders

database.seeds.get\_seeders() → list

### database.seeds.init\_seed

database.seeds.init\_seed() → None  
database.seeds.get\_seeders() → list  
database.seeds.init\_seed() → None

## Functions

---

*init\_database()*  
*seed\_actions(fnc)*

---

### 2.2.4 database.init\_database

database.init\_database() → None

### 2.2.5 database.seed\_actions

database.seed\_actions(fnc)  
database.init\_database() → None  
database.seed\_actions(fnc)

## 2.3 tests

### Description

Package for testing the application.

The tests package stores the application's tests that they are executed for ensuring the proper behaviour of the application.

You can get report coverage statistics with coverage package.

### Notes

There are three kinds of tests created:

1. **Unit testing** Unit testing means testing individual modules of an application in isolation (without any interaction with dependencies) to confirm that the code is doing things right.
2. **Integration testing** Integration testing means checking if different modules are working fine when combined together as a group.
3. **Functional testing** Functional testing means testing a slice of functionality in the system (may interact with dependencies) to confirm that the code is doing the right things.

Let us understand these three types of testing with an oversimplified example.

E.g. For a functional mobile phone, the main parts required are “battery” and “sim card”.

Unit testing Example – The battery is checked for its life, capacity and other parameters. Sim card is checked for its activation.

Integration Testing Example – Battery and sim card are integrated i.e. assembled in order to start the mobile phone.

Functional Testing Example – The functionality of a mobile phone is checked in terms of its features and battery usage as well as sim card facilities.

## References

[The Differences Between Unit Testing, Integration Testing and Functional Testing.](#)

## Examples

How to usage:

```
source venv/bin/activate
pytest
```

How to call a specific test:

```
source venv/bin/activate
pytest -k test_welcome_api
```

You can use coverage package for running tests as well:

```
source venv/bin/activate
coverage run -m pytest
```

And get a report coverage statistics on modules:

```
source venv/bin/activate
coverage report -m
```

For a nicer presentation, use coverage html to get annotated HTML listings detailing missed lines:

```
source venv/bin/activate
coverage html
```

## Modules

|                                |                                   |
|--------------------------------|-----------------------------------|
| <code>tests.blueprints</code>  | Package for testing blueprints.   |
| <code>tests.celery</code>      | Package for testing Celery tasks. |
| <code>tests.conftest</code>    | Module for configuring Pytest.    |
| <code>tests.test_config</code> | Module for testing Config module. |
| <code>tests.test_db</code>     | Module for testing database.      |
| <code>tests.test_mail</code>   | Module for testing mail.          |

## 2.3.1 tests.blueprints

### Description

Package for testing blueprints.

### Modules

|                                        |                                         |
|----------------------------------------|-----------------------------------------|
| <i>tests.blueprints.test_auth</i>      | Module for testing auth blueprint.      |
| <i>tests.blueprints.test_base</i>      | Module for testing base blueprint.      |
| <i>tests.blueprints.test_documents</i> | Module for testing documents blueprint. |
| <i>tests.blueprints.test_roles</i>     | Module for testing roles blueprint.     |
| <i>tests.blueprints.test_users</i>     | Module for testing users blueprint.     |

### tests.blueprints.test\_auth

### Description

Module for testing auth blueprint.

### Functions

|                                                  |
|--------------------------------------------------|
| <i>test_request_reset_password(client)</i>       |
| <i>test_reset_password(client)</i>               |
| <i>test_user_login(client)</i>                   |
| <i>test_user_logout(client, auth_header)</i>     |
| <i>test_validate_reset_password(client, app)</i> |

### tests.blueprints.test\_auth.test\_request\_reset\_password

tests.blueprints.test\_auth.**test\_request\_reset\_password**(*client*:  
*flask.testing.FlaskClient*)

**tests.blueprints.test\_auth.test\_reset\_password**

```
tests.blueprints.test_auth.test_reset_password(client: flask.testing.FlaskClient)
```

**tests.blueprints.test\_auth.test\_user\_login**

```
tests.blueprints.test_auth.test_user_login(client: flask.testing.FlaskClient)
```

**tests.blueprints.test\_auth.test\_user\_logout**

```
tests.blueprints.test_auth.test_user_logout(client: flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_auth.test\_validate\_reset\_password**

```
tests.blueprints.test_auth.test_validate_reset_password(client: flask.testing.FlaskClient,
 app: flask.app.Flask)
tests.blueprints.test_auth.test_request_reset_password(client: flask.testing.FlaskClient)
tests.blueprints.test_auth.test_reset_password(client: flask.testing.FlaskClient)
tests.blueprints.test_auth.test_user_login(client: flask.testing.FlaskClient)
tests.blueprints.test_auth.test_user_logout(client: flask.testing.FlaskClient,
 auth_header: any)
tests.blueprints.test_auth.test_validate_reset_password(client: flask.testing.FlaskClient,
 app: flask.app.Flask)
```

**tests.blueprints.test\_base****Description**

Module for testing base blueprint.

**Functions**


---

```
test_welcome_api(client)
```

---

### tests.blueprints.test\_base.test\_welcome\_api

tests.blueprints.test\_base.**test\_welcome\_api** (*client: flask.testing.FlaskClient*)

tests.blueprints.test\_base.**test\_welcome\_api** (*client: flask.testing.FlaskClient*)

### tests.blueprints.test\_documents

#### Description

Module for testing documents blueprint.

#### Functions

---

*test\_delete\_document*(client, auth\_header)

---

*test\_get\_document\_data*(client, auth\_header)

---

*test\_get\_document\_file*(client, auth\_header)

---

*test\_save\_document*(client, auth\_header)

---

*test\_search\_document*(client, auth\_header)

---

*test\_update\_document*(client, auth\_header)

---

### tests.blueprints.test\_documents.test\_delete\_document

tests.blueprints.test\_documents.**test\_delete\_document** (*client:*  
*flask.testing.FlaskClient,*  
*auth\_header: any*)

### tests.blueprints.test\_documents.test\_get\_document\_data

tests.blueprints.test\_documents.**test\_get\_document\_data** (*client:*  
*flask.testing.FlaskClient,*  
*auth\_header: any*)



### tests.blueprints.test\_documents.test\_get\_document\_file

```
tests.blueprints.test_documents.test_get_document_file (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

### tests.blueprints.test\_documents.test\_save\_document

```
tests.blueprints.test_documents.test_save_document (client: flask.testing.FlaskClient,
 auth_header: any)
```

### tests.blueprints.test\_documents.test\_search\_document

```
tests.blueprints.test_documents.test_search_document (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

### tests.blueprints.test\_documents.test\_update\_document

```
tests.blueprints.test_documents.test_update_document (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_delete_document (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_get_document_data (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_get_document_file (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_save_document (client: flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_search_document (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_documents.test_update_document (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_roles****Description**

Module for testing roles blueprint.

**Functions**

---

```
test_delete_role_endpoint(client,
auth_header)
test_get_role_endpoint(client, auth_header)
test_save_role_endpoint(client, auth_header,
...)
test_search_roles_endpoint(client,
auth_header)
test_update_role_endpoint(client, ...)
```

---

**tests.blueprints.test\_roles.test\_delete\_role\_endpoint**

```
tests.blueprints.test_roles.test_delete_role_endpoint (client:
flask.testing.FlaskClient,
auth_header: any)
```

**tests.blueprints.test\_roles.test\_get\_role\_endpoint**

```
tests.blueprints.test_roles.test_get_role_endpoint (client: flask.testing.FlaskClient,
auth_header: any)
```

**tests.blueprints.test\_roles.test\_save\_role\_endpoint**

```
tests.blueprints.test_roles.test_save_role_endpoint (client: flask.testing.FlaskClient,
auth_header: any, factory: any)
```

**tests.blueprints.test\_roles.test\_search\_roles\_endpoint**

```
tests.blueprints.test_roles.test_search_roles_endpoint (client:
flask.testing.FlaskClient,
auth_header: any)
```

**tests.blueprints.test\_roles.test\_update\_role\_endpoint**

```

tests.blueprints.test_roles.test_update_role_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any, factory:
 any)

tests.blueprints.test_roles.test_delete_role_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_roles.test_get_role_endpoint (client: flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_roles.test_save_role_endpoint (client: flask.testing.FlaskClient,
 auth_header: any, factory: any)

tests.blueprints.test_roles.test_search_roles_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)

tests.blueprints.test_roles.test_update_role_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any, factory:
 any)

```

**tests.blueprints.test\_users****Description**

Module for testing users blueprint.

**Functions**


---

```

test_delete_user_endpoint(client,
auth_header)

```

---

```

test_export_excel_endpoint(client,
auth_header)

```

---

```

test_export_word_endpoint(client,
auth_header)

```

---

```

test_get_user_endpoint(client, auth_header)

```

---

```

test_save_user_endpoint(client, auth_header,
...)

```

---

```

test_search_users_endpoint(client,
auth_header)

```

---

```

test_update_user_endpoint(client, ...)

```

---

**tests.blueprints.test\_users.test\_delete\_user\_endpoint**

```
tests.blueprints.test_users.test_delete_user_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_users.test\_export\_excel\_endpoint**

```
tests.blueprints.test_users.test_export_excel_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_users.test\_export\_word\_endpoint**

```
tests.blueprints.test_users.test_export_word_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_users.test\_get\_user\_endpoint**

```
tests.blueprints.test_users.test_get_user_endpoint (client: flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_users.test\_save\_user\_endpoint**

```
tests.blueprints.test_users.test_save_user_endpoint (client: flask.testing.FlaskClient,
 auth_header: any, factory: any)
```

**tests.blueprints.test\_users.test\_search\_users\_endpoint**

```
tests.blueprints.test_users.test_search_users_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

**tests.blueprints.test\_users.test\_update\_user\_endpoint**

```
tests.blueprints.test_users.test_update_user_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any, factory:
 any)
```

```
tests.blueprints.test_users.test_delete_user_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

```
tests.blueprints.test_users.test_export_excel_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
```

```

tests.blueprints.test_users.test_export_word_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
tests.blueprints.test_users.test_get_user_endpoint (client: flask.testing.FlaskClient,
 auth_header: any)
tests.blueprints.test_users.test_save_user_endpoint (client: flask.testing.FlaskClient,
 auth_header: any, factory: any)
tests.blueprints.test_users.test_search_users_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any)
tests.blueprints.test_users.test_update_user_endpoint (client:
 flask.testing.FlaskClient,
 auth_header: any, factory:
 any)

```

## 2.3.2 tests.celery

### Description

Package for testing Celery tasks.

### Modules

|                                      |                                  |
|--------------------------------------|----------------------------------|
| <code>tests.celery.test_excel</code> | Module for testing excel module. |
| <code>tests.celery.test_tasks</code> | Module for testing task module.  |
| <code>tests.celery.test_word</code>  | Module for testing word module.  |

### tests.celery.test\_excel

### Description

Module for testing excel module.

### Functions

|                                          |
|------------------------------------------|
| <code>test_export_excel_task(app)</code> |
|------------------------------------------|

**tests.celery.test\_excel.test\_export\_excel\_task**

tests.celery.test\_excel.**test\_export\_excel\_task** (*app: flask.app.Flask*)

tests.celery.test\_excel.**test\_export\_excel\_task** (*app: flask.app.Flask*)

**tests.celery.test\_tasks****Description**

Module for testing task module.

**Functions**

---

*test\_create\_user\_email\_task*(factory)

---

*test\_create\_word\_and\_excel\_documents*(app)

---

*test\_reset\_password\_email\_task*(app)

---

**tests.celery.test\_tasks.test\_create\_user\_email\_task**

tests.celery.test\_tasks.**test\_create\_user\_email\_task** (*factory: any*)

**tests.celery.test\_tasks.test\_create\_word\_and\_excel\_documents**

tests.celery.test\_tasks.**test\_create\_word\_and\_excel\_documents** (*app:*  
*flask.app.Flask*)

**tests.celery.test\_tasks.test\_reset\_password\_email\_task**

tests.celery.test\_tasks.**test\_reset\_password\_email\_task** (*app: flask.app.Flask*)

tests.celery.test\_tasks.**test\_create\_user\_email\_task** (*factory: any*)

tests.celery.test\_tasks.**test\_create\_word\_and\_excel\_documents** (*app:*  
*flask.app.Flask*)

tests.celery.test\_tasks.**test\_reset\_password\_email\_task** (*app: flask.app.Flask*)

**tests.celery.test\_word****Description**

Module for testing word module.

## Functions

---

`test_export_word_task(app)`


---

### tests.celery.test\_word.test\_export\_word\_task

`tests.celery.test_word.test_export_word_task (app: flask.app.Flask)`
`tests.celery.test_word.test_export_word_task (app: flask.app.Flask)`

## 2.3.3 tests.conftest

### Description

Module for configuring Pytest.

### Functions

|                                       |                                                                                |
|---------------------------------------|--------------------------------------------------------------------------------|
| <code>app()</code>                    | Create an app with testing environment.                                        |
| <code>auth_header(app, client)</code> | Create an auth header from a given user that can be added to an http requests. |
| <code>client(app)</code>              | Create a test client for making http requests.                                 |
| <code>factory(app)</code>             | Create a Factory from a database model.                                        |
| <code>runner(app)</code>              | Create a CLI runner for testing CLI commands.                                  |

### tests.conftest.app

`tests.conftest.app()`  
 Create an app with testing environment.

### tests.conftest.auth\_header

`tests.conftest.auth_header (app: flask.app.Flask, client: flask.testing.FlaskClient)`  
 Create an auth header from a given user that can be added to an http requests.

### tests.conftest.client

`tests.conftest.client (app: flask.app.Flask)`  
 Create a test client for making http requests.

### tests.conftest.factory

`tests.conftest.factory` (*app: flask.app.Flask*)  
Create a Factory from a database model.

### tests.conftest.runner

`tests.conftest.runner` (*app: flask.app.Flask*)  
Create a CLI runner for testing CLI commands.

`tests.conftest._remove_test_files` (*storage\_path: str*) → None  
Remove test files created in storage path.

`tests.conftest.app` ()  
Create an app with testing environment.

`tests.conftest.auth_header` (*app: flask.app.Flask, client: flask.testing.FlaskClient*)  
Create an auth header from a given user that can be added to an http requests.

`tests.conftest.client` (*app: flask.app.Flask*)  
Create a test client for making http requests.

`tests.conftest.factory` (*app: flask.app.Flask*)  
Create a Factory from a database model.

`tests.conftest.runner` (*app: flask.app.Flask*)  
Create a CLI runner for testing CLI commands.

## 2.3.4 tests.test\_config

### Description

Module for testing Config module.

### Functions

---

|                            |                                        |
|----------------------------|----------------------------------------|
| <code>test_config()</code> | Check if TESTING attribute is enabled. |
|----------------------------|----------------------------------------|

---

### tests.test\_config.test\_config

`tests.test_config.test_config` ()  
Check if TESTING attribute is enabled.

`tests.test_config.test_config` ()  
Check if TESTING attribute is enabled.



### 2.3.5 tests.test\_db

#### Description

Module for testing database.

#### Functions

---

|                                  |                                           |
|----------------------------------|-------------------------------------------|
| <code>test_get_close_db()</code> | Check if a database connection is closed. |
|----------------------------------|-------------------------------------------|

---

#### tests.test\_db.test\_get\_close\_db

`tests.test_db.test_get_close_db()`  
Check if a database connection is closed.

`tests.test_db.test_get_close_db()`  
Check if a database connection is closed.

### 2.3.6 tests.test\_mail

#### Description

Module for testing mail.

#### Functions

---

|                                             |                           |
|---------------------------------------------|---------------------------|
| <code>test_mail_record_messages(app)</code> | Check if a email is sent. |
|---------------------------------------------|---------------------------|

---

#### tests.test\_mail.test\_mail\_record\_messages

`tests.test_mail.test_mail_record_messages(app)`  
Check if a email is sent.

References

Unit tests and suppressing emails

`tests.test_mail.test_mail_record_messages (app)`  
Check if a email is sent.

References

Unit tests and suppressing emails

2.4 config

Description

Module loads the application’s configuration.  
The extension and custom configurations are defined here.

Classes

|                                |                                        |
|--------------------------------|----------------------------------------|
| <i>Config()</i>                | Default configuration options.         |
| <i>DevConfig()</i>             | Development configuration options.     |
| <i>Meta(name, bases, dict)</i> | Metaclass for updating Config options. |
| <i>ProdConfig()</i>            | Production configuration options.      |
| <i>TestConfig()</i>            | Testing configuration options.         |

2.4.1 config.Config

**class** `config.Config`  
Bases: object  
Default configuration options.

Attributes

|                                     |
|-------------------------------------|
| <i>Config.ALLOWED_CONTENT_TYPES</i> |
| <i>Config.ALLOWED_MIME_TYPES</i>    |
| <i>Config.DATABASE</i>              |
| <i>Config.DEBUG</i>                 |
| <i>Config.DEVELOPMENT</i>           |
| <i>Config.ERROR_404_HELP</i>        |
| <i>Config.FLASK_RESTFUL_PREFIX</i>  |
| <i>Config.HOME</i>                  |
| <i>Config.LOGIN_DISABLED</i>        |
| <i>Config.LOG_DIRECTORY</i>         |
| <i>Config.MAIL_PASSWORD</i>         |

continues on next page

Table 155 – continued from previous page

|                                                    |
|----------------------------------------------------|
| <i>Config.MAIL_PORT</i>                            |
| <i>Config.MAIL_SERVER</i>                          |
| <i>Config.MAIL_USERNAME</i>                        |
| <i>Config.MAIL_USE_SSL</i>                         |
| <i>Config.MAIL_USE_TLS</i>                         |
| <i>Config.RESET_TOKEN_EXPIRES</i>                  |
| <i>Config.RESTX_MASK_SWAGGER</i>                   |
| <i>Config.ROOT_DIRECTORY</i>                       |
| <i>Config.SECRET_KEY</i>                           |
| <i>Config.SECURITY_PASSWORD_HASH</i>               |
| <i>Config.SECURITY_PASSWORD_LENGTH_MIN</i>         |
| <i>Config.SECURITY_PASSWORD_SALT</i>               |
| <i>Config.SECURITY_TOKEN_AUTHENTICATION_HEADER</i> |
| <i>Config.SECURITY_TOKEN_MAX_AGE</i>               |
| <i>Config.SERVER_NAME</i>                          |
| <i>Config.STORAGE_DIRECTORY</i>                    |
| <i>Config.SWAGGER_API_URL</i>                      |
| <i>Config.SWAGGER_URL</i>                          |
| <i>Config.TESTING</i>                              |
| <i>Config.TEST_USER_EMAIL</i>                      |
| <i>Config.TEST_USER_PASSWORD</i>                   |
| <i>Config.accept_content</i>                       |
| <i>Config.broker_url</i>                           |
| <i>Config.enable_utc</i>                           |
| <i>Config.include</i>                              |
| <i>Config.result_backend</i>                       |
| <i>Config.result_expires</i>                       |
| <i>Config.result_extended</i>                      |
| <i>Config.result_serializer</i>                    |
| <i>Config.task_default_rate_limit</i>              |
| <i>Config.task_serializer</i>                      |
| <i>Config.task_track_started</i>                   |
| <i>Config.timezone</i>                             |
| <i>Config.worker_log_format</i>                    |
| <i>Config.worker_task_log_format</i>               |

**config.Config.ALLOWED\_CONTENT\_TYPES**

Config.**ALLOWED\_CONTENT\_TYPES** = {'application/json', 'application/octet-stream', 'mu

**config.Config.ALLOWED\_MIME\_TYPES**

```
Config.ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
```

**config.Config.DATABASE**

```
Config.DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas': {}}
```

**config.Config.DEBUG**

```
Config.DEBUG = False
```

**config.Config.DEVELOPMENT**

```
Config.DEVELOPMENT = False
```

**config.Config.ERROR\_404\_HELP**

```
Config.ERROR_404_HELP = False
```

**config.Config.FLASK\_RESTFUL\_PREFIX**

```
Config.FLASK_RESTFUL_PREFIX = '/api'
```

**config.Config.HOME**

```
Config.HOME = '/home/docs'
```

**config.Config.LOGIN\_DISABLED**

```
Config.LOGIN_DISABLED = False
```

**config.Config.LOG\_DIRECTORY**

```
Config.LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/
```

**config.Config.MAIL\_PASSWORD**

```
Config.MAIL_PASSWORD = None
```

**config.Config.MAIL\_PORT**

```
Config.MAIL_PORT = None
```

**config.Config.MAIL\_SERVER**

```
Config.MAIL_SERVER = None
```

**config.Config.MAIL\_USERNAME**

```
Config.MAIL_USERNAME = None
```

**config.Config.MAIL\_USE\_SSL**

```
Config.MAIL_USE_SSL = False
```

**config.Config.MAIL\_USE\_TLS**

```
Config.MAIL_USE_TLS = True
```

**config.Config.RESET\_TOKEN\_EXPIRES**

```
Config.RESET_TOKEN_EXPIRES = 86400
```

**config.Config.RESTX\_MASK\_SWAGGER**

```
Config.RESTX_MASK_SWAGGER = False
```

**config.Config.ROOT\_DIRECTORY**

```
Config.ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api'
```

**config.Config.SECRET\_KEY**

```
Config.SECRET_KEY = None
```

**config.Config.SECURITY\_PASSWORD\_HASH**

```
Config.SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
```

**config.Config.SECURITY\_PASSWORD\_LENGTH\_MIN**

```
Config.SECURITY_PASSWORD_LENGTH_MIN = 8
```

**config.Config.SECURITY\_PASSWORD\_SALT**

```
Config.SECURITY_PASSWORD_SALT = None
```

**config.Config.SECURITY\_TOKEN\_AUTHENTICATION\_HEADER**

```
Config.SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
```

**config.Config.SECURITY\_TOKEN\_MAX\_AGE**

```
Config.SECURITY_TOKEN_MAX_AGE = None
```

**config.Config.SERVER\_NAME**

```
Config.SERVER_NAME = None
```

**config.Config.STORAGE\_DIRECTORY**

```
Config.STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-
```

**config.Config.SWAGGER\_API\_URL**

```
Config.SWAGGER_API_URL = 'http://None/static/swagger.yaml'
```

**config.Config.SWAGGER\_URL**

```
Config.SWAGGER_URL = '/docs'
```

**config.Config.TESTING**

```
Config.TESTING = False
```

**config.Config.TEST\_USER\_EMAIL**

```
Config.TEST_USER_EMAIL = None
```

**config.Config.TEST\_USER\_PASSWORD**

```
Config.TEST_USER_PASSWORD = None
```

**config.Config.accept\_content**

```
Config.accept_content = ['json']
```

**config.Config.broker\_url**

```
Config.broker_url = 'pyamqp://'
```

**config.Config.enable\_utc**

```
Config.enable_utc = True
```

**config.Config.include**

```
Config.include = ['app.celery.tasks']
```

**config.Config.result\_backend**

```
Config.result_backend = 'amqp://'
```

**config.Config.result\_expires**

```
Config.result_expires = 3600
```

**config.Config.result\_extended**

```
Config.result_extended = True
```

**config.Config.result\_serializer**

```
Config.result_serializer = 'json'
```

**config.Config.task\_default\_rate\_limit**

```
Config.task_default_rate_limit = 3
```

**config.Config.task\_serializer**

```
Config.task_serializer = 'json'
```

**config.Config.task\_track\_started**

```
Config.task_track_started = True
```

**config.Config.timezone**

```
Config.timezone = 'UTC'
```

**config.Config.worker\_log\_format**

```
Config.worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**config.Config.worker\_task\_log\_format**

```
Config.worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**Methods**

---

|                          |                  |
|--------------------------|------------------|
| <i>Config.__init__()</i> | Initialize self. |
|--------------------------|------------------|

---

**config.Config.\_\_init\_\_**

```
Config.__init__()
```

Initialize self. See help(type(self)) for accurate signature.

## 2.4.2 config.DevConfig

**class** config.DevConfig

Bases: *config.Config*

Development configuration options.

**Attributes**

---

|                                        |
|----------------------------------------|
| <i>DevConfig.ALLOWED_CONTENT_TYPES</i> |
| <i>DevConfig.ALLOWED_MIME_TYPES</i>    |
| <i>DevConfig.DATABASE</i>              |
| <i>DevConfig.DEBUG</i>                 |
| <i>DevConfig.DEVELOPMENT</i>           |
| <i>DevConfig.ERROR_404_HELP</i>        |
| <i>DevConfig.FLASK_RESTFUL_PREFIX</i>  |
| <i>DevConfig.HOME</i>                  |

---

continues on next page



Table 157 – continued from previous page

|                                                                  |
|------------------------------------------------------------------|
| <i>DevConfig.LOGIN_DISABLED</i>                                  |
| <i>DevConfig.LOG_DIRECTORY</i>                                   |
| <i>DevConfig.MAIL_PASSWORD</i>                                   |
| <i>DevConfig.MAIL_PORT</i>                                       |
| <i>DevConfig.MAIL_SERVER</i>                                     |
| <i>DevConfig.MAIL_USERNAME</i>                                   |
| <i>DevConfig.MAIL_USE_SSL</i>                                    |
| <i>DevConfig.MAIL_USE_TLS</i>                                    |
| <i>DevConfig.RESET_TOKEN_EXPIRES</i>                             |
| <i>DevConfig.RESTX_MASK_SWAGGER</i>                              |
| <i>DevConfig.ROOT_DIRECTORY</i>                                  |
| <i>DevConfig.SECRET_KEY</i>                                      |
| <i>DevConfig.</i><br><i>SECURITY_PASSWORD_HASH</i>               |
| <i>DevConfig.</i><br><i>SECURITY_PASSWORD_LENGTH_MIN</i>         |
| <i>DevConfig.</i><br><i>SECURITY_PASSWORD_SALT</i>               |
| <i>DevConfig.</i><br><i>SECURITY_TOKEN_AUTHENTICATION_HEADER</i> |
| <i>DevConfig.</i><br><i>SECURITY_TOKEN_MAX_AGE</i>               |
| <i>DevConfig.SERVER_NAME</i>                                     |
| <i>DevConfig.STORAGE_DIRECTORY</i>                               |
| <i>DevConfig.SWAGGER_API_URL</i>                                 |
| <i>DevConfig.SWAGGER_URL</i>                                     |
| <i>DevConfig.TESTING</i>                                         |
| <i>DevConfig.TEST_USER_EMAIL</i>                                 |
| <i>DevConfig.TEST_USER_PASSWORD</i>                              |
| <i>DevConfig.accept_content</i>                                  |
| <i>DevConfig.broker_url</i>                                      |
| <i>DevConfig.enable_utc</i>                                      |
| <i>DevConfig.include</i>                                         |
| <i>DevConfig.result_backend</i>                                  |
| <i>DevConfig.result_expires</i>                                  |
| <i>DevConfig.result_extended</i>                                 |
| <i>DevConfig.result_serializer</i>                               |
| <i>DevConfig.</i><br><i>task_default_rate_limit</i>              |
| <i>DevConfig.task_serializer</i>                                 |
| <i>DevConfig.task_track_started</i>                              |
| <i>DevConfig.timezone</i>                                        |
| <i>DevConfig.worker_log_format</i>                               |
| <i>DevConfig.</i><br><i>worker_task_log_format</i>               |

**config.DevConfig.ALLOWED\_CONTENT\_TYPES**

```
DevConfig.ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream',
```

**config.DevConfig.ALLOWED\_MIME\_TYPES**

```
DevConfig.ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
```

**config.DevConfig.DATABASE**

```
DevConfig.DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas':
```

**config.DevConfig.DEBUG**

```
DevConfig.DEBUG = True
```

**config.DevConfig.DEVELOPMENT**

```
DevConfig.DEVELOPMENT = True
```

**config.DevConfig.ERROR\_404\_HELP**

```
DevConfig.ERROR_404_HELP = False
```

**config.DevConfig.FLASK\_RESTFUL\_PREFIX**

```
DevConfig.FLASK_RESTFUL_PREFIX = '/api'
```

**config.DevConfig.HOME**

```
DevConfig.HOME = '/home/docs'
```

**config.DevConfig.LOGIN\_DISABLED**

```
DevConfig.LOGIN_DISABLED = False
```

**config.DevConfig.LOG\_DIRECTORY**

```
DevConfig.LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-a
```

**config.DevConfig.MAIL\_PASSWORD**

DevConfig.MAIL\_PASSWORD = None

**config.DevConfig.MAIL\_PORT**

DevConfig.MAIL\_PORT = None

**config.DevConfig.MAIL\_SERVER**

DevConfig.MAIL\_SERVER = None

**config.DevConfig.MAIL\_USERNAME**

DevConfig.MAIL\_USERNAME = None

**config.DevConfig.MAIL\_USE\_SSL**

DevConfig.MAIL\_USE\_SSL = False

**config.DevConfig.MAIL\_USE\_TLS**

DevConfig.MAIL\_USE\_TLS = True

**config.DevConfig.RESET\_TOKEN\_EXPIRES**

DevConfig.RESET\_TOKEN\_EXPIRES = 86400

**config.DevConfig.RESTX\_MASK\_SWAGGER**

DevConfig.RESTX\_MASK\_SWAGGER = False

**config.DevConfig.ROOT\_DIRECTORY**

DevConfig.ROOT\_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user\_builds/flask-

**config.DevConfig.SECRET\_KEY**

DevConfig.SECRET\_KEY = None

**config.DevConfig.SECURITY\_PASSWORD\_HASH**

```
DevConfig.SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
```

**config.DevConfig.SECURITY\_PASSWORD\_LENGTH\_MIN**

```
DevConfig.SECURITY_PASSWORD_LENGTH_MIN = 8
```

**config.DevConfig.SECURITY\_PASSWORD\_SALT**

```
DevConfig.SECURITY_PASSWORD_SALT = None
```

**config.DevConfig.SECURITY\_TOKEN\_AUTHENTICATION\_HEADER**

```
DevConfig.SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
```

**config.DevConfig.SECURITY\_TOKEN\_MAX\_AGE**

```
DevConfig.SECURITY_TOKEN_MAX_AGE = None
```

**config.DevConfig.SERVER\_NAME**

```
DevConfig.SERVER_NAME = None
```

**config.DevConfig.STORAGE\_DIRECTORY**

```
DevConfig.STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/fla
```

**config.DevConfig.SWAGGER\_API\_URL**

```
DevConfig.SWAGGER_API_URL = 'http://None/static/swagger.yaml'
```

**config.DevConfig.SWAGGER\_URL**

```
DevConfig.SWAGGER_URL = '/docs'
```

**config.DevConfig.TESTING**

```
DevConfig.TESTING = False
```

**config.DevConfig.TEST\_USER\_EMAIL**

```
DevConfig.TEST_USER_EMAIL = None
```

**config.DevConfig.TEST\_USER\_PASSWORD**

```
DevConfig.TEST_USER_PASSWORD = None
```

**config.DevConfig.accept\_content**

```
DevConfig.accept_content = ['json']
```

**config.DevConfig.broker\_url**

```
DevConfig.broker_url = 'pyamqp://'
```

**config.DevConfig.enable\_utc**

```
DevConfig.enable_utc = True
```

**config.DevConfig.include**

```
DevConfig.include = ['app.celery.tasks']
```

**config.DevConfig.result\_backend**

```
DevConfig.result_backend = 'amqp://'
```

**config.DevConfig.result\_expires**

```
DevConfig.result_expires = 3600
```

**config.DevConfig.result\_extended**

```
DevConfig.result_extended = True
```

**config.DevConfig.result\_serializer**

```
DevConfig.result_serializer = 'json'
```

**config.DevConfig.task\_default\_rate\_limit**

```
DevConfig.task_default_rate_limit = 3
```

**config.DevConfig.task\_serializer**

```
DevConfig.task_serializer = 'json'
```

**config.DevConfig.task\_track\_started**

```
DevConfig.task_track_started = True
```

**config.DevConfig.timezone**

```
DevConfig.timezone = 'UTC'
```

**config.DevConfig.worker\_log\_format**

```
DevConfig.worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**config.DevConfig.worker\_task\_log\_format**

```
DevConfig.worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**Methods**

---

|                             |                  |
|-----------------------------|------------------|
| <i>DevConfig.__init__()</i> | Initialize self. |
|-----------------------------|------------------|

---

**config.DevConfig.\_\_init\_\_**

```
DevConfig.__init__()
```

```
Initialize self. See help(type(self)) for accurate signature.
```

### 2.4.3 config.Meta

**class** config.**Meta** (*name: str, bases: tuple, dict: dict*)

Bases: type

Metaclass for updating Config options.

#### Methods

|                                             |                                          |
|---------------------------------------------|------------------------------------------|
| <code>Meta.__init__(*args, **kwargs)</code> | Initialize self.                         |
| <code>Meta.mro()</code>                     | Return a type's method resolution order. |

#### config.Meta.\_\_init\_\_

`Meta.__init__(*args, **kwargs)`

Initialize self. See help(type(self)) for accurate signature.

#### config.Meta.mro

`Meta.mro()`

Return a type's method resolution order.

### 2.4.4 config.ProdConfig

**class** config.**ProdConfig**

Bases: `config.Config`

Production configuration options.

#### Attributes

|                                              |
|----------------------------------------------|
| <code>ProdConfig.</code>                     |
| <code>ALLOWED_CONTENT_TYPES</code>           |
| <code>ProdConfig.ALLOWED_MIME_TYPES</code>   |
| <code>ProdConfig.DATABASE</code>             |
| <code>ProdConfig.DEBUG</code>                |
| <code>ProdConfig.DEVELOPMENT</code>          |
| <code>ProdConfig.ERROR_404_HELP</code>       |
| <code>ProdConfig.FLASK_RESTFUL_PREFIX</code> |
| <code>ProdConfig.HOME</code>                 |
| <code>ProdConfig.LOGIN_DISABLED</code>       |
| <code>ProdConfig.LOG_DIRECTORY</code>        |
| <code>ProdConfig.MAIL_PASSWORD</code>        |
| <code>ProdConfig.MAIL_PORT</code>            |
| <code>ProdConfig.MAIL_SERVER</code>          |
| <code>ProdConfig.MAIL_USERNAME</code>        |
| <code>ProdConfig.MAIL_USE_SSL</code>         |
| <code>ProdConfig.MAIL_USE_TLS</code>         |

continues on next page

Table 160 – continued from previous page

|                                                                   |
|-------------------------------------------------------------------|
| <i>ProdConfig.RESET_TOKEN_EXPIRES</i>                             |
| <i>ProdConfig.RESTX_MASK_SWAGGER</i>                              |
| <i>ProdConfig.ROOT_DIRECTORY</i>                                  |
| <i>ProdConfig.SECRET_KEY</i>                                      |
| <i>ProdConfig.</i><br><i>SECURITY_PASSWORD_HASH</i>               |
| <i>ProdConfig.</i><br><i>SECURITY_PASSWORD_LENGTH_MIN</i>         |
| <i>ProdConfig.</i><br><i>SECURITY_PASSWORD_SALT</i>               |
| <i>ProdConfig.</i><br><i>SECURITY_TOKEN_AUTHENTICATION_HEADER</i> |
| <i>ProdConfig.</i><br><i>SECURITY_TOKEN_MAX_AGE</i>               |
| <i>ProdConfig.SERVER_NAME</i>                                     |
| <i>ProdConfig.STORAGE_DIRECTORY</i>                               |
| <i>ProdConfig.SWAGGER_API_URL</i>                                 |
| <i>ProdConfig.SWAGGER_URL</i>                                     |
| <i>ProdConfig.TESTING</i>                                         |
| <i>ProdConfig.TEST_USER_EMAIL</i>                                 |
| <i>ProdConfig.TEST_USER_PASSWORD</i>                              |
| <i>ProdConfig.accept_content</i>                                  |
| <i>ProdConfig.broker_url</i>                                      |
| <i>ProdConfig.enable_utc</i>                                      |
| <i>ProdConfig.include</i>                                         |
| <i>ProdConfig.result_backend</i>                                  |
| <i>ProdConfig.result_expires</i>                                  |
| <i>ProdConfig.result_extended</i>                                 |
| <i>ProdConfig.result_serializer</i>                               |
| <i>ProdConfig.</i><br><i>task_default_rate_limit</i>              |
| <i>ProdConfig.task_serializer</i>                                 |
| <i>ProdConfig.task_track_started</i>                              |
| <i>ProdConfig.timezone</i>                                        |
| <i>ProdConfig.worker_log_format</i>                               |
| <i>ProdConfig.</i><br><i>worker_task_log_format</i>               |

**config.ProdConfig.ALLOWED\_CONTENT\_TYPES**

```
ProdConfig.ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream',
```



**config.ProdConfig.ALLOWED\_MIME\_TYPES**

```
ProdConfig.ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
```

**config.ProdConfig.DATABASE**

```
ProdConfig.DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas': {}}
```

**config.ProdConfig.DEBUG**

```
ProdConfig.DEBUG = False
```

**config.ProdConfig.DEVELOPMENT**

```
ProdConfig.DEVELOPMENT = False
```

**config.ProdConfig.ERROR\_404\_HELP**

```
ProdConfig.ERROR_404_HELP = False
```

**config.ProdConfig.FLASK\_RESTFUL\_PREFIX**

```
ProdConfig.FLASK_RESTFUL_PREFIX = '/api'
```

**config.ProdConfig.HOME**

```
ProdConfig.HOME = '/home/docs'
```

**config.ProdConfig.LOGIN\_DISABLED**

```
ProdConfig.LOGIN_DISABLED = False
```

**config.ProdConfig.LOG\_DIRECTORY**

```
ProdConfig.LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/
```

**config.ProdConfig.MAIL\_PASSWORD**

```
ProdConfig.MAIL_PASSWORD = None
```

**config.ProdConfig.MAIL\_PORT**

```
ProdConfig.MAIL_PORT = None
```

**config.ProdConfig.MAIL\_SERVER**

```
ProdConfig.MAIL_SERVER = None
```

**config.ProdConfig.MAIL\_USERNAME**

```
ProdConfig.MAIL_USERNAME = None
```

**config.ProdConfig.MAIL\_USE\_SSL**

```
ProdConfig.MAIL_USE_SSL = False
```

**config.ProdConfig.MAIL\_USE\_TLS**

```
ProdConfig.MAIL_USE_TLS = True
```

**config.ProdConfig.RESET\_TOKEN\_EXPIRES**

```
ProdConfig.RESET_TOKEN_EXPIRES = 86400
```

**config.ProdConfig.RESTX\_MASK\_SWAGGER**

```
ProdConfig.RESTX_MASK_SWAGGER = False
```

**config.ProdConfig.ROOT\_DIRECTORY**

```
ProdConfig.ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask
```

**config.ProdConfig.SECRET\_KEY**

```
ProdConfig.SECRET_KEY = None
```

**config.ProdConfig.SECURITY\_PASSWORD\_HASH**

```
ProdConfig.SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
```

**config.ProdConfig.SECURITY\_PASSWORD\_LENGTH\_MIN**

```
ProdConfig.SECURITY_PASSWORD_LENGTH_MIN = 8
```

**config.ProdConfig.SECURITY\_PASSWORD\_SALT**

```
ProdConfig.SECURITY_PASSWORD_SALT = None
```

**config.ProdConfig.SECURITY\_TOKEN\_AUTHENTICATION\_HEADER**

```
ProdConfig.SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
```

**config.ProdConfig.SECURITY\_TOKEN\_MAX\_AGE**

```
ProdConfig.SECURITY_TOKEN_MAX_AGE = None
```

**config.ProdConfig.SERVER\_NAME**

```
ProdConfig.SERVER_NAME = None
```

**config.ProdConfig.STORAGE\_DIRECTORY**

```
ProdConfig.STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/builds/1234567890/docs'
```

**config.ProdConfig.SWAGGER\_API\_URL**

```
ProdConfig.SWAGGER_API_URL = 'http://None/static/swagger.yaml'
```

**config.ProdConfig.SWAGGER\_URL**

```
ProdConfig.SWAGGER_URL = '/docs'
```

**config.ProdConfig.TESTING**

```
ProdConfig.TESTING = False
```

**config.ProdConfig.TEST\_USER\_EMAIL**

```
ProdConfig.TEST_USER_EMAIL = None
```

**config.ProdConfig.TEST\_USER\_PASSWORD**

`ProdConfig.TEST_USER_PASSWORD = None`

**config.ProdConfig.accept\_content**

`ProdConfig.accept_content = ['json']`

**config.ProdConfig.broker\_url**

`ProdConfig.broker_url = 'pyamqp://'`

**config.ProdConfig.enable\_utc**

`ProdConfig.enable_utc = True`

**config.ProdConfig.include**

`ProdConfig.include = ['app.celery.tasks']`

**config.ProdConfig.result\_backend**

`ProdConfig.result_backend = 'amqp://'`

**config.ProdConfig.result\_expires**

`ProdConfig.result_expires = 3600`

**config.ProdConfig.result\_extended**

`ProdConfig.result_extended = True`

**config.ProdConfig.result\_serializer**

`ProdConfig.result_serializer = 'json'`

**config.ProdConfig.task\_default\_rate\_limit**

`ProdConfig.task_default_rate_limit = 3`

**config.ProdConfig.task\_serializer**

```
ProdConfig.task_serializer = 'json'
```

**config.ProdConfig.task\_track\_started**

```
ProdConfig.task_track_started = True
```

**config.ProdConfig.timezone**

```
ProdConfig.timezone = 'UTC'
```

**config.ProdConfig.worker\_log\_format**

```
ProdConfig.worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**config.ProdConfig.worker\_task\_log\_format**

```
ProdConfig.worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**Methods**


---

|                              |                  |
|------------------------------|------------------|
| <i>ProdConfig.__init__()</i> | Initialize self. |
|------------------------------|------------------|

---

**config.ProdConfig.\_\_init\_\_**

```
ProdConfig.__init__()
 Initialize self. See help(type(self)) for accurate signature.
```

**2.4.5 config.TestConfig**

```
class config.TestConfig
```

Bases: *config.Config*

Testing configuration options.

**Attributes**


---

|                                         |
|-----------------------------------------|
| <i>TestConfig.ALLOWED_CONTENT_TYPES</i> |
| <i>TestConfig.ALLOWED_MIME_TYPES</i>    |
| <i>TestConfig.DATABASE</i>              |
| <i>TestConfig.DEBUG</i>                 |
| <i>TestConfig.DEVELOPMENT</i>           |
| <i>TestConfig.ERROR_404_HELP</i>        |
| <i>TestConfig.FLASK_RESTFUL_PREFIX</i>  |

---

continues on next page

Table 162 – continued from previous page

|                                                                   |
|-------------------------------------------------------------------|
| <i>TestConfig.HOME</i>                                            |
| <i>TestConfig.LOGIN_DISABLED</i>                                  |
| <i>TestConfig.LOG_DIRECTORY</i>                                   |
| <i>TestConfig.MAIL_PASSWORD</i>                                   |
| <i>TestConfig.MAIL_PORT</i>                                       |
| <i>TestConfig.MAIL_SERVER</i>                                     |
| <i>TestConfig.MAIL_USERNAME</i>                                   |
| <i>TestConfig.MAIL_USE_SSL</i>                                    |
| <i>TestConfig.MAIL_USE_TLS</i>                                    |
| <i>TestConfig.RESET_TOKEN_EXPIRES</i>                             |
| <i>TestConfig.RESTX_MASK_SWAGGER</i>                              |
| <i>TestConfig.ROOT_DIRECTORY</i>                                  |
| <i>TestConfig.SECRET_KEY</i>                                      |
| <i>TestConfig.</i><br><i>SECURITY_PASSWORD_HASH</i>               |
| <i>TestConfig.</i><br><i>SECURITY_PASSWORD_LENGTH_MIN</i>         |
| <i>TestConfig.</i><br><i>SECURITY_PASSWORD_SALT</i>               |
| <i>TestConfig.</i><br><i>SECURITY_TOKEN_AUTHENTICATION_HEADER</i> |
| <i>TestConfig.</i><br><i>SECURITY_TOKEN_MAX_AGE</i>               |
| <i>TestConfig.SERVER_NAME</i>                                     |
| <i>TestConfig.STORAGE_DIRECTORY</i>                               |
| <i>TestConfig.SWAGGER_API_URL</i>                                 |
| <i>TestConfig.SWAGGER_URL</i>                                     |
| <i>TestConfig.TESTING</i>                                         |
| <i>TestConfig.TEST_USER_EMAIL</i>                                 |
| <i>TestConfig.TEST_USER_PASSWORD</i>                              |
| <i>TestConfig.accept_content</i>                                  |
| <i>TestConfig.broker_url</i>                                      |
| <i>TestConfig.enable_utc</i>                                      |
| <i>TestConfig.include</i>                                         |
| <i>TestConfig.result_backend</i>                                  |
| <i>TestConfig.result_expires</i>                                  |
| <i>TestConfig.result_extended</i>                                 |
| <i>TestConfig.result_serializer</i>                               |
| <i>TestConfig.</i><br><i>task_default_rate_limit</i>              |
| <i>TestConfig.task_serializer</i>                                 |
| <i>TestConfig.task_track_started</i>                              |
| <i>TestConfig.timezone</i>                                        |
| <i>TestConfig.worker_log_format</i>                               |
| <i>TestConfig.</i><br><i>worker_task_log_format</i>               |

**config.TestConfig.ALLOWED\_CONTENT\_TYPES**

```
TestConfig.ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream',
```

**config.TestConfig.ALLOWED\_MIME\_TYPES**

```
TestConfig.ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
```

**config.TestConfig.DATABASE**

```
TestConfig.DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': 'test.db', 'pra
```

**config.TestConfig.DEBUG**

```
TestConfig.DEBUG = True
```

**config.TestConfig.DEVELOPMENT**

```
TestConfig.DEVELOPMENT = True
```

**config.TestConfig.ERROR\_404\_HELP**

```
TestConfig.ERROR_404_HELP = False
```

**config.TestConfig.FLASK\_RESTFUL\_PREFIX**

```
TestConfig.FLASK_RESTFUL_PREFIX = '/api'
```

**config.TestConfig.HOME**

```
TestConfig.HOME = '/home/docs'
```

**config.TestConfig.LOGIN\_DISABLED**

```
TestConfig.LOGIN_DISABLED = False
```

**config.TestConfig.LOG\_DIRECTORY**

```
TestConfig.LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-
```

**config.TestConfig.MAIL\_PASSWORD**

```
TestConfig.MAIL_PASSWORD = None
```

**config.TestConfig.MAIL\_PORT**

```
TestConfig.MAIL_PORT = None
```

**config.TestConfig.MAIL\_SERVER**

```
TestConfig.MAIL_SERVER = None
```

**config.TestConfig.MAIL\_USERNAME**

```
TestConfig.MAIL_USERNAME = None
```

**config.TestConfig.MAIL\_USE\_SSL**

```
TestConfig.MAIL_USE_SSL = False
```

**config.TestConfig.MAIL\_USE\_TLS**

```
TestConfig.MAIL_USE_TLS = True
```

**config.TestConfig.RESET\_TOKEN\_EXPIRES**

```
TestConfig.RESET_TOKEN_EXPIRES = 86400
```

**config.TestConfig.RESTX\_MASK\_SWAGGER**

```
TestConfig.RESTX_MASK_SWAGGER = False
```

**config.TestConfig.ROOT\_DIRECTORY**

```
TestConfig.ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask'
```

**config.TestConfig.SECRET\_KEY**

```
TestConfig.SECRET_KEY = None
```



**config.TestConfig.SECURITY\_PASSWORD\_HASH**

```
TestConfig.SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
```

**config.TestConfig.SECURITY\_PASSWORD\_LENGTH\_MIN**

```
TestConfig.SECURITY_PASSWORD_LENGTH_MIN = 8
```

**config.TestConfig.SECURITY\_PASSWORD\_SALT**

```
TestConfig.SECURITY_PASSWORD_SALT = None
```

**config.TestConfig.SECURITY\_TOKEN\_AUTHENTICATION\_HEADER**

```
TestConfig.SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
```

**config.TestConfig.SECURITY\_TOKEN\_MAX\_AGE**

```
TestConfig.SECURITY_TOKEN_MAX_AGE = None
```

**config.TestConfig.SERVER\_NAME**

```
TestConfig.SERVER_NAME = None
```

**config.TestConfig.STORAGE\_DIRECTORY**

```
TestConfig.STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/builds/1234567890/docs'
```

**config.TestConfig.SWAGGER\_API\_URL**

```
TestConfig.SWAGGER_API_URL = 'http://None/static/swagger.yaml'
```

**config.TestConfig.SWAGGER\_URL**

```
TestConfig.SWAGGER_URL = '/docs'
```

**config.TestConfig.TESTING**

```
TestConfig.TESTING = True
```

**config.TestConfig.TEST\_USER\_EMAIL**

```
TestConfig.TEST_USER_EMAIL = None
```

**config.TestConfig.TEST\_USER\_PASSWORD**

```
TestConfig.TEST_USER_PASSWORD = None
```

**config.TestConfig.accept\_content**

```
TestConfig.accept_content = ['json']
```

**config.TestConfig.broker\_url**

```
TestConfig.broker_url = 'pyamqp://'
```

**config.TestConfig.enable\_utc**

```
TestConfig.enable_utc = True
```

**config.TestConfig.include**

```
TestConfig.include = ['app.celery.tasks']
```

**config.TestConfig.result\_backend**

```
TestConfig.result_backend = 'amqp://'
```

**config.TestConfig.result\_expires**

```
TestConfig.result_expires = 3600
```

**config.TestConfig.result\_extended**

```
TestConfig.result_extended = True
```

**config.TestConfig.result\_serializer**

```
TestConfig.result_serializer = 'json'
```

**config.TestConfig.task\_default\_rate\_limit**

```
TestConfig.task_default_rate_limit = 3
```

**config.TestConfig.task\_serializer**

```
TestConfig.task_serializer = 'json'
```

**config.TestConfig.task\_track\_started**

```
TestConfig.task_track_started = True
```

**config.TestConfig.timezone**

```
TestConfig.timezone = 'UTC'
```

**config.TestConfig.worker\_log\_format**

```
TestConfig.worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**config.TestConfig.worker\_task\_log\_format**

```
TestConfig.worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

**Methods**


---

|                              |                  |
|------------------------------|------------------|
| <i>TestConfig.__init__()</i> | Initialize self. |
|------------------------------|------------------|

---

**config.TestConfig.\_\_init\_\_**

```
TestConfig.__init__()
 Initialize self. See help(type(self)) for accurate signature.
```

**class config.Config**

Default configuration options.

```
ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream', 'multipart/form-data', 'text/html'}
```

```
ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
```

```
DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas': {'cache_size': 512, 'journal_mode': 'wal'}}
```

```
DEBUG = False
```

```
DEVELOPMENT = False
```

```
ERROR_404_HELP = False
```

```
FLASK_RESTFUL_PREFIX = '/api'
```

```
HOME = '/home/docs'
```

```
LOGIN_DISABLED = False
```

```
LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/'
MAIL_PASSWORD = None
MAIL_PORT = None
MAIL_SERVER = None
MAIL_USERNAME = None
MAIL_USE_SSL = False
MAIL_USE_TLS = True
RESET_TOKEN_EXPIRES = 86400
RESTX_MASK_SWAGGER = False
ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/'
SECRET_KEY = None
SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
SECURITY_PASSWORD_LENGTH_MIN = 8
SECURITY_PASSWORD_SALT = None
SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
SECURITY_TOKEN_MAX_AGE = None
SERVER_NAME = None
STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/'
SWAGGER_API_URL = 'http://None/static/swagger.yaml'
SWAGGER_URL = '/docs'
TESTING = False
TEST_USER_EMAIL = None
TEST_USER_PASSWORD = None
accept_content = ['json']
broker_url = 'pyamqp://'
enable_utc = True
include = ['app.celery.tasks']
result_backend = 'amqp://'
result_expires = 3600
result_extended = True
result_serializer = 'json'
task_default_rate_limit = 3
task_serializer = 'json'
task_track_started = True
timezone = 'UTC'
worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
```

```

 worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(task_name)s'
class config.DevConfig
 Development configuration options.

 ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream', 'multipart/form-data'}
 ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
 DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas': {'cache_size': 10000}}
 DEBUG = True
 DEVELOPMENT = True
 ERROR_404_HELP = False
 FLASK_RESTFUL_PREFIX = '/api'
 HOME = '/home/docs'
 LOGIN_DISABLED = False
 LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/default-logs'
 MAIL_PASSWORD = None
 MAIL_PORT = None
 MAIL_SERVER = None
 MAIL_USERNAME = None
 MAIL_USE_SSL = False
 MAIL_USE_TLS = True
 RESET_TOKEN_EXPIRES = 86400
 RESTX_MASK_SWAGGER = False
 ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/default'
 SECRET_KEY = None
 SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
 SECURITY_PASSWORD_LENGTH_MIN = 8
 SECURITY_PASSWORD_SALT = None
 SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
 SECURITY_TOKEN_MAX_AGE = None
 SERVER_NAME = None
 STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/default-storage'
 SWAGGER_API_URL = 'http://None/static/swagger.yaml'
 SWAGGER_URL = '/docs'
 TESTING = False
 TEST_USER_EMAIL = None
 TEST_USER_PASSWORD = None
 accept_content = ['json']

```

```
broker_url = 'pyamqp://'
enable_utc = True
include = ['app.celery.tasks']
result_backend = 'amqp://'
result_expires = 3600
result_extended = True
result_serializer = 'json'
task_default_rate_limit = 3
task_serializer = 'json'
task_track_started = True
timezone = 'UTC'
worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(task_name)s'

class config.Meta(name: str, bases: tuple, dict: dict)
 Metaclass for updating Config options.

 classmethod _rename_celery_settings(config: type) → None
 Rename old Celery setting names with new ones.
```

## References

<https://docs.celeryproject.org/en/latest/userguide/configuration.html#new-lowercase-settings>

```
mro()
 Return a type's method resolution order.
```

```
class config.ProdConfig
 Production configuration options.

 ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream', 'multipart/form-data'}
 ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
 DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': None, 'pragmas': {'cache_size': 512000}}
 DEBUG = False
 DEVELOPMENT = False
 ERROR_404_HELP = False
 FLASK_RESTFUL_PREFIX = '/api'
 HOME = '/home/docs'
 LOGIN_DISABLED = False
 LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/latest/logs'
 MAIL_PASSWORD = None
 MAIL_PORT = None
 MAIL_SERVER = None
```

---

```

MAIL_USERNAME = None
MAIL_USE_SSL = False
MAIL_USE_TLS = True
RESET_TOKEN_EXPIRES = 86400
RESTX_MASK_SWAGGER = False
ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts'
SECRET_KEY = None
SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
SECURITY_PASSWORD_LENGTH_MIN = 8
SECURITY_PASSWORD_SALT = None
SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
SECURITY_TOKEN_MAX_AGE = None
SERVER_NAME = None
STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts'
SWAGGER_API_URL = 'http://None/static/swagger.yaml'
SWAGGER_URL = '/docs'
TESTING = False
TEST_USER_EMAIL = None
TEST_USER_PASSWORD = None
accept_content = ['json']
broker_url = 'pyamqp://'
enable_utc = True
include = ['app.celery.tasks']
result_backend = 'amqp://'
result_expires = 3600
result_extended = True
result_serializer = 'json'
task_default_rate_limit = 3
task_serializer = 'json'
task_track_started = True
timezone = 'UTC'
worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(task_name)s'

class config.TestConfig
 Testing configuration options.

 ALLOWED_CONTENT_TYPES = {'application/json', 'application/octet-stream', 'multipart/form-data'}

```

```
ALLOWED_MIME_TYPES = {'application/pdf', 'application/vnd.ms-excel'}
DATABASE = {'engine': 'peewee.SqliteDatabase', 'name': 'test.db', 'pragmas': {'cach
DEBUG = True
DEVELOPMENT = True
ERROR_404_HELP = False
FLASK_RESTFUL_PREFIX = '/api'
HOME = '/home/docs'
LOGIN_DISABLED = False
LOG_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts/
MAIL_PASSWORD = None
MAIL_PORT = None
MAIL_SERVER = None
MAIL_USERNAME = None
MAIL_USE_SSL = False
MAIL_USE_TLS = True
RESET_TOKEN_EXPIRES = 86400
RESTX_MASK_SWAGGER = False
ROOT_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checkouts
SECRET_KEY = None
SECURITY_PASSWORD_HASH = 'pbkdf2_sha512'
SECURITY_PASSWORD_LENGTH_MIN = 8
SECURITY_PASSWORD_SALT = None
SECURITY_TOKEN_AUTHENTICATION_HEADER = 'Authorization'
SECURITY_TOKEN_MAX_AGE = None
SERVER_NAME = None
STORAGE_DIRECTORY = '/home/docs/checkouts/readthedocs.org/user_builds/flask-api/checko
SWAGGER_API_URL = 'http://None/static/swagger.yaml'
SWAGGER_URL = '/docs'
TESTING = True
TEST_USER_EMAIL = None
TEST_USER_PASSWORD = None
accept_content = ['json']
broker_url = 'pyamqp://'
enable_utc = True
include = ['app.celery.tasks']
result_backend = 'amqp://'
```



```
result_expires = 3600
result_extended = True
result_serializer = 'json'
task_default_rate_limit = 3
task_serializer = 'json'
task_track_started = True
timezone = 'UTC'
worker_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(message)s'
worker_task_log_format = '%(asctime)s - %(levelname)s - %(processName)s - %(task_name)s'
```



## FLASK COMMAND LINE

Flask command line allow run scripts for managing database, start up task queues, etc.

You don't need to start up the server for running these scripts but you must activate your virtual environment.



## CHANGELOG

All notable changes to this project will be documented in this file. See [standard-version](#) for commit guidelines.

### 4.1 2.0.0 (2020-10-27)

#### 4.1.1 BREAKING CHANGES

- **order** field in search requests is a list of dicts.

#### 4.1.2 Features

- **factories:** add prevent code for checking if a given model is registered as factory ([1605a01](#))
- **shell:** import Factory class to Flask interactive shell ([8cf9c8e](#))

#### 4.1.3 Code Refactoring

- replace cerberus to flask-mashmallow validation ([7552d5c](#))
- **celery:** replace old Celery setting names with new ones ([15e0c03](#))
- **celery:** update way to set FLASK\_CONFIG value on Flask command ([a865548](#))

#### 4.1.4 Build System

- add .versionrc that shows build/perf/refactor/revert ([0017d66](#))
- add sphinx-click configuration to Sphinx and create new file for showing Click documentation ([bb41b7b](#))
- add sphinx-click for showing Click documentation in Sphinx ([c0a8f55](#))
- **pip:** remove cerberus package ([4a4fe72](#))
- **pip:** split python packages in two requirements local and production ([f39a2a5](#))

#### 4.1.5 1.4.1 (2020-10-07)

#### 4.1.6 Bug Fixes

- **celery:** correct problem when start Celery ([52ad2fb](#)), closes #3

### 4.2 1.4.0 (2020-10-04)

#### 4.2.1 Features

- **celery:** add task for exporting several files ([ca8355f](#))
- **documentation:** add sphinx integration ([8c313fd](#))

### 4.3 1.3.0 (2020-09-20)

#### 4.3.1 Features

- **swagger:** add Swagger full integration ([1eaf8d8](#))

### 4.4 1.2.0 (2020-09-18)

#### 4.4.1 BREAKING CHANGES

- install/update Node.js and Python libraries

#### 4.4.2 build

- update Node.js and Python packages ([b7416cc](#))

### 4.5 1.1.0 (2020-05-31)

#### 4.5.1 Features

- **security:** add role-based authorization ([345b57e](#))
- add advanced search in documents, roles and users ([8fce3e3](#))
- add marshmallow package integration ([a8b647e](#))
- add Swagger integration ([dc6ace4](#))

### 4.5.2 Refactor

- replace HTTP exceptions to Werkzeug HTTP Exceptions (31e5606)
- move Word and Excel celery tasks to them own modules (00e42e5)

### 4.5.3 Docs

- docs: add installation project guide (b915d31)

## 4.6 1.0.0 (2020-05-17)

### 4.6.1 BREAKING CHANGES

- **pip:** update Python dependencies

### 4.6.2 Features

- **celery:** add basic installation (147dd2c)
- **db:** add pewee migrations (231696c)
- **documents:** add document logic (1eb7ec1)
- **emails:** add send emails after creation an user (7c2cfe0)
- **log:** add support for logrotate (09925e1)
- **users:** add created\_by column in user model (8a3d013)
- **users:** add Excel and PDF users export to background processes (781e091)
- **users:** add recovery password feature (e1e916e)

### 4.6.3 Build

- **pip:** update requirements.txt (6193153)

## 4.7 0.8.0 (2020-04-29)

### 4.7.1 Features

- **roles:** add role logic (d7a0535)
- **security:** add jwt authentication (fb51089)
- **users:** add role model integration to user model (69bc124)
- **users:** add user get endpoint (018b965)

## 4.8 0.7.0 (2020-04-23)

### 4.8.1 Features

- **doc:** add standard-version NodeJS package ([c1b2cb3](#))

### 4.8.2 0.6.1 (2020-04-23)

### 4.8.3 BREAKING CHANGES

- update python dependencies

### 4.8.4 Features

- **db:** added script for creating database tables ([c14b566](#))
- **logging:** added logging configuration ([297b9c3](#))
- **seeders:** added user seeder ([e78b4c4](#))
- **tests:** add tests and code coverage ([17317b7](#))
- **validation-requests:** add validation requests with cerberus ([a5beed6](#))

### 4.8.5 Bug Fixes

- **commitizen:** fixed problem with the process of commitizen tags ([1d3677d](#))
- **docs\_to\_pdf:** fixed problem about convert a docx file to a pdf file with uWSGI ([aabbcc2d](#))
- **peewee:** fixed problem about a connection already opened error ([6279470](#))
- **peewee:** problem with database connection already opened ([e6c07c9](#))
- **users:** update user endpoint cannot update data ([9dfc4cc](#))
- request search fields in search users, export PDF and export Excel endpoints ([2ae7ab7](#))

### 4.8.6 Build

- update requirements.txt ([f783e78](#))
- update requirements.txt ([b6378ba](#))



**NOTE**

If you find any bugs, odd behavior, or have an idea for a new feature please don't hesitate to on GitHub.



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### a

- app, 8
- app.blueprints, 8
- app.blueprints.auth, 8
- app.blueprints.base, 19
- app.blueprints.documents, 24
- app.blueprints.roles, 39
- app.blueprints.tasks, 52
- app.blueprints.users, 57
- app.celery, 80
- app.celery.excel, 80
- app.celery.excel.tasks, 80
- app.celery.tasks, 81
- app.celery.word, 81
- app.celery.word.tasks, 81
- app.extensions, 129
- app.middleware, 129
- app.models, 130
- app.models.base, 130
- app.models.document, 138
- app.models.role, 148
- app.models.user, 158
- app.utils, 174
- app.utils.decorators, 174
- app.utils.file\_storage, 175
- app.utils.libreoffice, 176
- app.utils.marshmallow\_schema, 177
- app.utils.swagger\_models, 257
- app.utils.swagger\_models.auth, 257
- app.utils.swagger\_models.document, 257
- app.utils.swagger\_models.role, 257
- app.utils.swagger\_models.user, 258

### c

- config, 302

### d

- database, 261
- database.factories, 261
- database.migrations, 266
- database.migrations.aaa\_add\_genre\_column\_on\_user\_table, 266

- database.migrations.aab\_add\_created\_by\_column\_on\_user, 267
- database.migrations.aac\_create\_documents\_table, 268
- database.migrations.aad\_create\_user\_roles\_table, 269
- database.migrations.aaf\_remove\_role\_slug\_column, 272
- database.seeds, 284
- database.seeds.document\_seeder, 284
- database.seeds.role\_seeder, 285
- database.seeds.user\_seeder, 286

### t

- tests, 288
- tests.blueprints, 290
- tests.blueprints.test\_auth, 290
- tests.blueprints.test\_base, 291
- tests.blueprints.test\_documents, 292
- tests.blueprints.test\_roles, 294
- tests.blueprints.test\_users, 295
- tests.celery, 297
- tests.celery.test\_excel, 297
- tests.celery.test\_tasks, 298
- tests.celery.test\_word, 298
- tests.conftest, 299
- tests.test\_config, 300
- tests.test\_db, 301
- tests.test\_mail, 301



## Symbols

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| <code>__init__</code> (method), 48  | <code>__init__</code> (method), 53  |
| <code>__init__</code> (method), 55  | <code>__init__</code> (method), 59  |
| <code>__init__</code> (method), 62  | <code>__init__</code> (method), 64  |
| <code>__init__</code> (method), 67  | <code>__init__</code> (method), 69  |
| <code>__init__</code> (method), 72  | <code>__init__</code> (method), 74  |
| <code>__init__</code> (method), 87  | <code>__init__</code> (method), 102 |
| <code>__init__</code> (method), 130 | <code>__init__</code> (method), 132 |
| <code>__init__</code> (method), 141 | <code>__init__</code> (method), 151 |
| <code>__init__</code> (method), 163 | <code>__init__</code> (method), 175 |
| <code>__init__</code> (method), 179 | <code>__init__</code> (method), 185 |
| <code>__init__</code> (method), 191 | <code>__init__</code> (method), 196 |
| <code>__init__</code> (method), 201 | <code>__init__</code> (method), 206 |
| <code>__init__</code> (method), 206 |                                     |

`__init__` (method), 48  
`__init__` (method), 53  
`__init__` (method), 55  
`__init__` (method), 59  
`__init__` (method), 62  
`__init__` (method), 64  
`__init__` (method), 67  
`__init__` (method), 69  
`__init__` (method), 72  
`__init__` (method), 74  
`__init__` (method), 87  
`__init__` (method), 102  
`__init__` (method), 130  
`__init__` (method), 132  
`__init__` (method), 141  
`__init__` (method), 151  
`__init__` (method), 163  
`__init__` (method), 175  
`__init__` (method), 179  
`__init__` (method), 185  
`__init__` (method), 191  
`__init__` (method), 196  
`__init__` (method), 201  
`__init__` (method), 206

`__init__()` (*app.utils.marshmallow\_schema.UserSchema* backend (*app.celery.ContextTask* attribute), 112  
 method), 209  
`__init__()` (*config.Config* method), 308  
`__init__()` (*config.DevConfig* method), 314  
`__init__()` (*config.Meta* method), 315  
`__init__()` (*config.ProdConfig* method), 321  
`__init__()` (*config.TestConfig* method), 327  
`__init__()` (*database.factories.Factory* method), 263  
`__init__()` (*database.migrations.Migration* method), 277  
`__init__()` (*database.migrations.aaa\_add\_genre\_column\_on\_user\_table* AddGenreColumnOnUserTable  
 method), 267  
`__init__()` (*database.migrations.aab\_add\_created\_by\_column\_on\_user\_table* AddCreatedByColumnOnUserTable  
 method), 268  
`__init__()` (*database.migrations.aac\_create\_documents\_table* CreateDocumentsTable  
 method), 269  
`__init__()` (*database.migrations.aad\_create\_user\_roles\_table* CreateUserRolesTable  
 method), 270  
`__init__()` (*database.migrations.aaf\_remove\_role\_slug\_column\_on\_role\_schema* RemoveRoleSlugColumn  
 method), 273  
`__init__()` (*database.seeds.document\_seeder.DocumentSeeder* method), 285  
`__init__()` (*database.seeds.role\_seeder.RoleSeeder* method), 286  
`__init__()` (*database.seeds.user\_seeder.UserSeeder* method), 287  
`_acquire_connection()` (*app.celery.MyCelery* method), 120  
`_add_excel Autofilter()` (in module *app.celery.excel.tasks*), 80  
`_add_foreign_key_constraint_users_table()` (*database.migrations.aad\_create\_user\_roles\_table* CreateUserRolesTable  
 static method), 270  
`_add_periodic_task()` (*app.celery.MyCelery* method), 120  
`_add_table_column_names()` (in module *app.celery.word.tasks*), 81  
`_add_table_user_data()` (in module *app.celery.word.tasks*), 81  
`_add_unique_constraint_roles_table()` (*database.migrations.aaf\_remove\_role\_slug\_column* RemoveRoleSlugColumn  
 static method), 273  
`_adjust_each_column_width()` (in module *app.celery.excel.tasks*), 80  
`_after_fork()` (*app.celery.MyCelery* method), 120  
`_after_fork_registered` (*app.celery.MyCelery* attribute), 120  
`_app` (*app.celery.ContextTask* attribute), 112  
`_autodiscover_tasks()` (*app.celery.MyCelery* method), 120  
`_autodiscover_tasks_from_fixups()` (*app.celery.MyCelery* method), 120  
`_autodiscover_tasks_from_names()` (*app.celery.MyCelery* method), 120  
`_bind_field()` (*app.utils.marshmallow\_schema.DocumentSchema* method), 213  
`_bind_field()` (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 219  
`_bind_field()` (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 224  
`_bind_field()` (*app.utils.marshmallow\_schema.RoleSchema* method), 229  
`_bind_field()` (*app.utils.marshmallow\_schema.SearchSchema* method), 241  
`_bind_field()` (*app.utils.marshmallow\_schema.UserSchema* method), 247  
`_bind_field()` (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 253  
`_bind_field()` (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 253  
`_bind_to_schema()` (*app.utils.marshmallow\_schema.Timestamp* method), 239  
`_build_and_clause_operators()` (in module *app.utils*), 260  
`_build_order_by()` (in module *app.utils*), 260  
`_build_query_clause()` (in module *app.utils*), 260  
`_build_string_clause()` (in module *app.utils*), 260  
`_call_and_store()` (*app.utils.marshmallow\_schema.DocumentSchema* static method), 213  
`_call_and_store()` (*app.utils.marshmallow\_schema.ExportWordInputSchema* static method), 219  
`_call_and_store()` (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* static method), 224  
`_call_and_store()` (*app.utils.marshmallow\_schema.RoleSchema* static method), 229  
`_call_and_store()` (*app.utils.marshmallow\_schema.SearchSchema* static method), 241  
`_call_and_store()` (*app.utils.marshmallow\_schema.UserSchema* static method), 247  
`_call_and_store()` (*app.utils.marshmallow\_schema.\_SearchOrderSchema* static method), 253  
`_call_and_store()` (*app.utils.marshmallow\_schema.\_SearchValueSchema* static method), 253  
`_canvas` (*app.celery.MyCelery* attribute), 120  
`_coerce` (*app.models.base.Base* attribute), 136  
`_coerce` (*app.models.document.Document* attribute), 146



\_coerce (*app.models.role.Role* attribute), 156  
 \_coerce (*app.models.user.User* attribute), 170  
 \_coerce (*database.migrations.Migration* attribute), 282  
 \_coerce (*database.migrations.aad\_create\_user\_roles\_table.CreateUserRole* attribute), 270  
 \_coerce (*database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlug* attribute), 273  
 \_conf (*app.celery.MyCelery* attribute), 120  
 \_connection () (*app.celery.MyCelery* method), 120  
 \_create\_admin\_role () (*database.seeds.role\_seeder.RoleSeeder* static method), 286  
 \_create\_admin\_user () (*database.seeds.user\_seeder.UserSeeder* static method), 287  
 \_create\_team\_leader () (*database.seeds.role\_seeder.RoleSeeder* static method), 286  
 \_create\_worker\_role () (*database.seeds.role\_seeder.RoleSeeder* static method), 286  
 \_creation\_index (*app.utils.marshmallow\_schema.TimestampSchema* attribute), 239  
 \_declared\_fields (*app.utils.marshmallow\_schema.DocumentSchema* attribute), 213  
 \_declared\_fields (*app.utils.marshmallow\_schema.ExportWordInputSchema* attribute), 219  
 \_declared\_fields (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* attribute), 225  
 \_declared\_fields (*app.utils.marshmallow\_schema.RoleSchema* attribute), 229  
 \_declared\_fields (*app.utils.marshmallow\_schema.SearchSchema* attribute), 235  
 \_declared\_fields (*app.utils.marshmallow\_schema.UserSchema* attribute), 242  
 \_declared\_fields (*app.utils.marshmallow\_schema.\_SearchOrderSchema* attribute), 247  
 \_declared\_fields (*app.utils.marshmallow\_schema.\_SearchValueSchema* attribute), 253  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.DocumentSchema* attribute), 213  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.ExportWordInputSchema* attribute), 219  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* attribute), 225  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.RoleSchema* attribute), 229  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.SearchSchema* attribute), 235  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.UserSchema* attribute), 242  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.\_SearchOrderSchema* attribute), 247  
 \_default\_error\_messages (*app.utils.marshmallow\_schema.\_SearchValueSchema* attribute), 253  
 \_deserialize () (*app.utils.marshmallow\_schema.DocumentSchema* method), 213  
 \_deserialize () (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 219  
 \_deserialize () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 225  
 \_deserialize () (*app.utils.marshmallow\_schema.RoleSchema* method), 229  
 \_deserialize () (*app.utils.marshmallow\_schema.SearchSchema* method), 235  
 \_deserialize () (*app.utils.marshmallow\_schema.TimestampSchema* method), 239  
 \_deserialize () (*app.utils.marshmallow\_schema.UserSchema* method), 242  
 \_deserialize () (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 247  
 \_deserialize () (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 253  
 \_do\_load () (*app.utils.marshmallow\_schema.DocumentSchema* method), 214  
 \_do\_load () (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 219  
 \_do\_load () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 225  
 \_do\_load () (*app.utils.marshmallow\_schema.RoleSchema* method), 230  
 \_do\_load () (*app.utils.marshmallow\_schema.SearchSchema* method), 235  
 \_do\_load () (*app.utils.marshmallow\_schema.UserSchema* method), 242  
 \_do\_load () (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 248  
 \_do\_load () (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 253  
 \_drop\_foreign\_key\_constraint\_users\_table () (*database.migrations.aad\_create\_user\_roles\_table.CreateUserRole* static method), 270  
 \_drop\_unique\_constraint\_roles\_table () (*database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlug* static method), 273  
 \_ensure\_after\_fork () (*app.celery.MyCelery* method), 120

\_exec\_options (app.celery.ContextTask attribute), 112  
 \_exists\_column() (database.migrations.aaa\_add\_genre\_column method), 267  
 \_exists\_column() (database.migrations.aab\_add\_created\_by\_column method), 268  
 \_exists\_column() (database.migrations.aaf\_remove\_role\_slug\_column method), 273  
 \_exists\_table() (database.migrations.aac\_create\_documents\_table method), 269  
 \_exists\_table() (database.migrations.aad\_create\_user\_roles table method), 270  
 \_finalize\_pending\_conf() (app.celery.MyCelery method), 120  
 \_fixups (app.celery.MyCelery attribute), 120  
 \_get\_app() (app.celery.ContextTask class method), 112  
 \_get\_backend() (app.celery.MyCelery method), 120  
 \_get\_default\_loader() (app.celery.MyCelery method), 120  
 \_get\_excel\_column\_names() (in module app.celery.excel.tasks), 80  
 \_get\_excel\_user\_data() (in module app.celery.excel.tasks), 80  
 \_get\_exec\_options() (app.celery.ContextTask method), 112  
 \_get\_request() (app.celery.ContextTask method), 112  
 \_get\_user\_data() (in module app.celery.excel.tasks), 80  
 \_get\_user\_data() (in module app.celery.word.tasks), 81  
 \_has\_processors() (app.utils.marshmallow\_schema.DocumentSchema method), 214  
 \_has\_processors() (app.utils.marshmallow\_schema.ExportWordInputSchema method), 220  
 \_has\_processors() (app.utils.marshmallow\_schema.GetDocumentDataInputSchema method), 225  
 \_has\_processors() (app.utils.marshmallow\_schema.RoleSchema method), 230  
 \_has\_processors() (app.utils.marshmallow\_schema.SearchSchema method), 236  
 \_has\_processors() (app.utils.marshmallow\_schema.UserSchema method), 242  
 \_has\_processors() (app.utils.marshmallow\_schema.\_SearchOrderSchema method), 248  
 \_has\_processors() (app.utils.marshmallow\_schema.\_SearchValueSchema method), 254  
 \_hooks (app.utils.marshmallow\_schema.UserSchema attribute), 242  
 \_hooks (app.utils.marshmallow\_schema.\_SearchOrderSchema attribute), 248  
 \_hooks (app.utils.marshmallow\_schema.\_SearchValueSchema attribute), 254  
 \_init\_app() (in module app), 260  
 \_init\_fields() (app.utils.marshmallow\_schema.DocumentSchema method), 214  
 \_init\_fields() (app.utils.marshmallow\_schema.ExportWordInputSchema method), 220  
 \_init\_fields() (app.utils.marshmallow\_schema.GetDocumentDataInputSchema method), 225  
 \_init\_fields() (app.utils.marshmallow\_schema.RoleSchema method), 230  
 \_init\_fields() (app.utils.marshmallow\_schema.SearchSchema method), 236  
 \_init\_fields() (app.utils.marshmallow\_schema.UserSchema method), 243  
 \_init\_fields() (app.utils.marshmallow\_schema.\_SearchOrderSchema method), 248  
 \_init\_fields() (app.utils.marshmallow\_schema.\_SearchValueSchema method), 254  
 \_init\_logging() (in module app), 260  
 \_make\_dump\_processors() (app.utils.marshmallow\_schema.DocumentSchema method), 214  
 \_make\_dump\_processors() (app.utils.marshmallow\_schema.ExportWordInputSchema method), 220  
 \_invoke\_dump\_processors() (app.utils.marshmallow\_schema.GetDocumentDataInputSchema method), 225  
 \_invoke\_dump\_processors() (app.utils.marshmallow\_schema.RoleSchema method), 230  
 \_invoke\_dump\_processors() (app.utils.marshmallow\_schema.SearchSchema method), 236  
 \_invoke\_dump\_processors() (app.utils.marshmallow\_schema.UserSchema method), 243

|                                                                                                                             |                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>_invoke_dump_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchOrderSchema</i><br>method), 248          | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.DocumentSchema</i><br>method), 214                    |
| <code>_invoke_dump_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchValueSchema</i><br>method), 254          | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.ExportWordInputSchema</i><br>method), 220             |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.DocumentSchema</i><br>method), 214             | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.GetDocumentDataInputSchema</i><br>method), 226        |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.ExportWordInputSchema</i><br>method), 220      | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.RoleSchema</i><br>method), 230                        |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.GetDocumentDataInputSchema</i><br>method), 225 | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.SearchSchema</i><br>method), 236                      |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.RoleSchema</i><br>method), 230                 | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema.UserSchema</i><br>method), 243                        |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.SearchSchema</i><br>method), 236               | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchOrderSchema</i><br>method), 248                |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema.UserSchema</i><br>method), 243                 | <code>_invoke_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchValueSchema</i><br>method), 254                |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema._SearchOrderSchema</i><br>method), 248         | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.DocumentSchema</i><br>method), 214             |
| <code>_invoke_field_validators()</code><br>( <i>app.utils.marshmallow_schema._SearchValueSchema</i><br>method), 254         | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.ExportWordInputSchema</i><br>method), 220      |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.DocumentSchema</i><br>method), 214              | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.GetDocumentDataInputSchema</i><br>method), 226 |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.ExportWordInputSchema</i><br>method), 220       | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.RoleSchema</i><br>method), 230                 |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.GetDocumentDataInputSchema</i><br>method), 226  | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.SearchSchema</i><br>method), 236               |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.RoleSchema</i><br>method), 230                  | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema.UserSchema</i><br>method), 243                 |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.SearchSchema</i><br>method), 236                | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema._SearchOrderSchema</i><br>method), 248         |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema.UserSchema</i><br>method), 243                  | <code>_invoke_schema_validators()</code><br>( <i>app.utils.marshmallow_schema._SearchValueSchema</i><br>method), 254         |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchOrderSchema</i><br>method), 248          | <code>_load_config()</code> ( <i>app.celery.MyCelery</i> method), 120                                                        |
| <code>_invoke_load_processors()</code><br>( <i>app.utils.marshmallow_schema._SearchValueSchema</i><br>method), 254          | <code>_meta</code> ( <i>app.models.base.Base</i> attribute), 136                                                             |
|                                                                                                                             | <code>_meta</code> ( <i>app.models.document.Document</i> attribute), 146                                                     |
|                                                                                                                             | <code>_meta</code> ( <i>app.models.role.Role</i> attribute), 156                                                             |
|                                                                                                                             | <code>_meta</code> ( <i>app.models.user.User</i> attribute), 170                                                             |
|                                                                                                                             | <code>_meta</code> ( <i>database.migrations.Migration</i> attribute), 282                                                    |



300  
 \_rename\_celery\_settings() (*config.Meta* class method), 330  
 \_rgetattr() (*app.celery.MyCelery* method), 121  
 \_run\_validator() (*app.utils.marshmallow\_schema.DocumentSchema* method), 214  
 \_run\_validator() (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 220  
 \_run\_validator() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 226  
 \_run\_validator() (*app.utils.marshmallow\_schema.RoleSchema* method), 231  
 \_run\_validator() (*app.utils.marshmallow\_schema.SearchSchema* method), 236  
 \_run\_validator() (*app.utils.marshmallow\_schema.UserSchema* method), 243  
 \_run\_validator() (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 248  
 \_run\_validator() (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 254  
 \_schema (*app.models.base.Base* attribute), 136  
 \_schema (*app.models.document.Document* attribute), 146  
 \_schema (*app.models.role.Role* attribute), 156  
 \_schema (*app.models.user.User* attribute), 170  
 \_schema (*database.migrations.Migration* attribute), 283  
 \_schema (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 270  
 \_schema (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* attribute), 273  
 \_serialize() (*app.utils.marshmallow\_schema.DocumentSchema* method), 214  
 \_serialize() (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 220  
 \_serialize() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 226  
 \_serialize() (*app.utils.marshmallow\_schema.RoleSchema* method), 231  
 \_serialize() (*app.utils.marshmallow\_schema.SearchSchema* method), 236  
 \_serialize() (*app.utils.marshmallow\_schema.Timestamp* method), 240  
 \_serialize() (*app.utils.marshmallow\_schema.UserSchema* method), 243  
 \_serialize() (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 248  
 \_serialize() (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 254  
 \_sig\_to\_periodic\_task\_entry() (*app.celery.MyCelery* method), 121  
 \_task\_from\_fun() (*app.celery.MyCelery* method), 121  
 \_validate() (*app.utils.marshmallow\_schema.Timestamp* method), 240  
 \_validate\_missing() (*app.utils.marshmallow\_schema.Timestamp* method), 240  
**A**  
 accept\_content (*config.Config* attribute), 307, 328  
 accept\_content (*config.DevConfig* attribute), 313, 329  
 accept\_content (*config.ProdConfig* attribute), 320, 331  
 accept\_content (*config.TestConfig* attribute), 326, 332  
 add\_defaults (*app.celery.ContextTask* attribute), 83, 112  
 acks\_on\_failure\_or\_timeout (*app.celery.ContextTask* attribute), 83, 112  
 active (*app.models.user.User* attribute), 159, 170  
 add\_created\_by\_column\_on\_user\_table (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 270  
 add\_around() (*app.celery.ContextTask* class method), 87, 112  
 add\_defaults() (*app.celery.MyCelery* method), 103, 121  
 add\_index() (*app.models.base.Base* class method), 132, 137  
 add\_index() (*app.models.document.Document* class method), 141, 146  
 add\_index() (*app.models.role.Role* class method), 149, 156  
 add\_index() (*app.models.user.User* class method), 163, 170  
 add\_index() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 270  
 add\_index() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 273  
 add\_index() (*database.migrations.Migration* class method), 277, 283  
 add\_periodic\_task() (*app.celery.MyCelery* method), 103, 121  
 add\_permissions() (*app.models.role.Role* method), 151, 156  
 add\_permissions() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* method), 273  
 add\_to\_backend() (*app.celery.ContextTask* method), 87, 112  
 add\_to\_backend() (*app.celery.ContextTask* method), 88, 112  
 AddCreatedByColumnOnUserTable (class in *database.migrations.aab\_add\_created\_by\_column\_on\_user\_table*), 267, 268  
 AddGenreColumnOnUserTable (class in *database.migrations.aaa\_add\_genre\_column\_on\_user\_table*), 267, 268



266, 267

`after_return()` (*app.celery.ContextTask* method), 88, 113

`alias()` (*app.models.base.Base* class method), 132, 137

`alias()` (*app.models.document.Document* class method), 141, 146

`alias()` (*app.models.role.Role* class method), 151, 156

`alias()` (*app.models.user.User* class method), 163, 170

`alias()` (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 270

`alias()` (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 273

`alias()` (*database.migrations.Migration* class method), 278, 283

`ALLOWED_CONTENT_TYPES` (*config.Config* attribute), 303, 327

`ALLOWED_CONTENT_TYPES` (*config.DevConfig* attribute), 310, 329

`ALLOWED_CONTENT_TYPES` (*config.ProdConfig* attribute), 316, 330

`ALLOWED_CONTENT_TYPES` (*config.TestConfig* attribute), 323, 331

`ALLOWED_MIME_TYPES` (*config.Config* attribute), 304, 327

`ALLOWED_MIME_TYPES` (*config.DevConfig* attribute), 310, 329

`ALLOWED_MIME_TYPES` (*config.ProdConfig* attribute), 317, 330

`ALLOWED_MIME_TYPES` (*config.TestConfig* attribute), 323, 331

`amqp` (*app.celery.MyCelery* attribute), 97, 121

`amqp_cls` (*app.celery.MyCelery* attribute), 98, 121

`annotate()` (*app.celery.ContextTask* class method), 88, 113

`annotations` (*app.celery.MyCelery* attribute), 98, 121

`app`  
module, 8

`app` (*app.celery.ContextTask* attribute), 83, 113

`app()` (in module *tests.conftest*), 299, 300

`app.blueprints`  
module, 8

`app.blueprints.auth`  
module, 8

`app.blueprints.base`  
module, 19

`app.blueprints.documents`  
module, 24

`app.blueprints.roles`  
module, 39

`app.blueprints.tasks`  
module, 52

`app.blueprints.users`  
module, 57

`app.celery`  
module, 80

`app.celery.excel`  
module, 80

`app.celery.excel.tasks`  
module, 80

`app.celery.tasks`  
module, 81

`app.celery.word`  
module, 81

`app.celery.word.tasks`  
module, 81

`app.extensions`  
module, 129

`app.middleware`  
module, 129

`app.models`  
module, 130

`app.models.base`  
module, 130

`app.models.document`  
module, 138

`app.models.role`  
module, 148

`app.models.user`  
module, 158

`app.utils`  
module, 174

`app.utils.decorators`  
module, 174

`app.utils.file_storage`  
module, 175

`app.utils.libreoffice`  
module, 176

`app.utils.marshmallow_schema`  
module, 177

`app.utils.swagger_models`  
module, 257

`app.utils.swagger_models.auth`  
module, 257

`app.utils.swagger_models.document`  
module, 257

`app.utils.swagger_models.role`  
module, 257

`app.utils.swagger_models.user`  
module, 258

`apply()` (*app.celery.ContextTask* method), 88, 113

`apply_async()` (*app.celery.ContextTask* method), 88, 113

`args` (*app.celery.ContextTask.MaxRetriesExceededError* attribute), 112

`args` (*app.celery.ContextTask.OperationalError* attribute), 112

`args` (*app.celery.TaskFailure* attribute), 128

args (*app.utils.FileEmptyError* attribute), 259  
 args (*app.utils.libreoffice.LibreOfficeError* attribute), 177  
 as\_view() (*app.blueprints.auth.AuthUserLoginResource* class method), 10, 17  
 as\_view() (*app.blueprints.auth.AuthUserLogoutResource* class method), 12, 17  
 as\_view() (*app.blueprints.auth.RequestResetPasswordResource* class method), 14, 17  
 as\_view() (*app.blueprints.auth.ResetPasswordResource* class method), 16, 18  
 as\_view() (*app.blueprints.base.BaseResource* class method), 20, 23  
 as\_view() (*app.blueprints.base.WelcomeResource* class method), 22, 23  
 as\_view() (*app.blueprints.documents.DocumentBaseResource* class method), 26, 36  
 as\_view() (*app.blueprints.documents.DocumentResource* class method), 29, 37  
 as\_view() (*app.blueprints.documents.NewDocumentResource* class method), 32, 37  
 as\_view() (*app.blueprints.documents.SearchDocumentResource* class method), 35, 38  
 as\_view() (*app.blueprints.roles.NewRoleResource* class method), 41, 49  
 as\_view() (*app.blueprints.roles.RoleBaseResource* class method), 43, 50  
 as\_view() (*app.blueprints.roles.RoleResource* class method), 46, 50  
 as\_view() (*app.blueprints.roles.RolesSearchResource* class method), 48, 51  
 as\_view() (*app.blueprints.tasks.TaskResource* class method), 53, 56  
 as\_view() (*app.blueprints.tasks.TaskStatusResource* class method), 55, 56  
 as\_view() (*app.blueprints.users.ExportUsersExcelAndWordResource* class method), 59, 75  
 as\_view() (*app.blueprints.users.ExportUsersExcelResource* class method), 62, 76  
 as\_view() (*app.blueprints.users.ExportUsersWordResource* class method), 64, 76  
 as\_view() (*app.blueprints.users.NewUserResource* class method), 67, 77  
 as\_view() (*app.blueprints.users.UserBaseResource* class method), 69, 78  
 as\_view() (*app.blueprints.users.UserResource* class method), 72, 78  
 as\_view() (*app.blueprints.users.UsersSearchResource* class method), 74, 79  
 AsyncResult (*app.celery.MyCelery* attribute), 96, 119  
 AsyncResult() (*app.celery.ContextTask* method), 87, 112  
 auth\_header() (in module *tests.conftest*), 299, 300  
 AuthUserLoginResource (class in *app.blueprints.auth*), 9, 17  
 AuthUserLogoutResource (class in *app.blueprints.auth*), 11, 17  
 autodiscover\_tasks() (*app.celery.MyCelery* method), 103, 121  
 autoregister (*app.celery.ContextTask* attribute), 83, 114

## B

backend (*app.celery.MyCelery* attribute), 98, 122  
 backend() (*app.celery.ContextTask* property), 83, 114  
 backend\_cls (*app.celery.MyCelery* attribute), 98, 122  
 Base (class in *app.models.base*), 130, 136  
 BaseResource (class in *app.blueprints.base*), 19, 23  
 Beat (*app.celery.MyCelery* attribute), 96, 119  
 bind() (*app.celery.ContextTask* class method), 90, 114  
 bind() (*app.models.base.Base* class method), 132, 137  
 bind() (*app.models.document.Document* class method), 141, 146  
 bind() (*app.models.role.Role* class method), 151, 156  
 bind() (*app.models.user.User* class method), 163, 170  
 bind() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 270  
 bind() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 bind() (*database.migrations.Migration* class method), 278, 283  
 bind\_ctx() (*app.models.base.Base* class method), 132, 137  
 bind\_ctx() (*app.models.document.Document* class method), 142, 146  
 bind\_ctx() (*app.models.role.Role* class method), 151, 156  
 bind\_ctx() (*app.models.user.User* class method), 164, 171  
 bind\_ctx() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 270  
 bind\_ctx() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 bind\_ctx() (*database.migrations.Migration* class method), 278, 283  
 birth\_date (*app.models.user.User* attribute), 159, 171  
 birth\_date (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 270  
 broker\_connection() (*app.celery.MyCelery* method), 104, 122  
 broker\_url (*config.Config* attribute), 307, 328  
 broker\_url (*config.DevConfig* attribute), 313, 329  
 broker\_url (*config.ProdConfig* attribute), 320, 331  
 broker\_url (*config.TestConfig* attribute), 326, 332  
 bugreport() (*app.celery.MyCelery* method), 104, 122  
 builtin\_fixups (*app.celery.MyCelery* attribute), 98, 122

`bulk_create()` (*app.models.base.Base class method*), 132, 137  
`bulk_create()` (*app.models.document.Document class method*), 142, 146  
`bulk_create()` (*app.models.role.Role class method*), 152, 157  
`bulk_create()` (*app.models.user.User class method*), 164, 171  
`bulk_create()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser class method*), 270  
`bulk_create()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole class method*), 274  
`bulk_create()` (*database.migrations.Migration class method*), 278, 283  
`bulk_update()` (*app.models.base.Base class method*), 132, 137  
`bulk_update()` (*app.models.document.Document class method*), 142, 146  
`bulk_update()` (*app.models.role.Role class method*), 152, 157  
`bulk_update()` (*app.models.user.User class method*), 164, 171  
`bulk_update()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser class method*), 270  
`bulk_update()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole class method*), 274  
`bulk_update()` (*database.migrations.Migration class method*), 278, 283

## C

`calc_username()` (*app.models.user.User method*), 164, 171  
`characters_written` (*app.utils.FileEmptyError attribute*), 259  
`children` (*app.models.user.User attribute*), 159, 171  
`children` (*database.migrations.aad\_create\_user\_roles\_table.OldUser attribute*), 271  
`chunks()` (*app.celery.ContextTask method*), 90, 115  
`class_for_name()` (*in module app.utils*), 258, 260  
`client()` (*in module tests.conftest*), 299, 300  
`clone()` (*app.models.base.Base method*), 133, 137  
`clone()` (*app.models.document.Document method*), 142, 146  
`clone()` (*app.models.role.Role method*), 152, 157  
`clone()` (*app.models.user.User method*), 164, 171  
`clone()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser method*), 271  
`clone()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole method*), 274  
`clone()` (*database.migrations.Migration method*), 278, 283  
`close()` (*app.celery.MyCelery method*), 104, 122  
`coerce()` (*app.models.base.Base method*), 133, 137  
`coerce()` (*app.models.document.Document method*), 142, 146  
`coerce()` (*app.models.role.Role method*), 152, 157  
`coerce()` (*app.models.user.User method*), 164, 171  
`coerce()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser method*), 271  
`coerce()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole method*), 274  
`coerce()` (*database.migrations.Migration method*), 278, 283  
`config` (*app.celery.MyCelery property*), 98, 123  
`config` (*module*), 302  
`Config` (*class in config*), 302, 327  
`config_from_cmdline()` (*app.celery.MyCelery method*), 105, 123  
`config_from_envvar()` (*app.celery.MyCelery method*), 105, 123  
`config_from_object()` (*app.celery.MyCelery method*), 105, 123  
`connection()` (*app.celery.MyCelery method*), 106, 123  
`connection_or_acquire()` (*app.celery.MyCelery method*), 107, 124  
`context()` (*app.utils.marshmallow\_schema.Timestamp property*), 205, 240  
`ContextTask` (*class in app.celery*), 82, 112  
`ContextTask.MaxRetriesExceededError`, 112  
`ContextTask.OperationalError`, 112  
`control` (*app.celery.MyCelery attribute*), 98, 124  
`control_cls` (*app.celery.MyCelery attribute*), 98, 124  
`convert_to_integer()` (*in module app.utils.libreoffice*), 177  
`convert_to_integer()` (*app.utils.marshmallow\_schema.ExportWordInputSchema method*), 185, 220  
`convert_to_integer()` (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema method*), 191, 226  
`copy()` (*app.models.base.Base static method*), 133, 137  
`copy()` (*app.models.document.Document static method*), 142, 146  
`copy()` (*app.models.role.Role static method*), 152, 157  
`copy()` (*app.models.user.User static method*), 164, 171  
`copy()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser static method*), 271  
`copy()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole static method*), 274  
`copy()` (*database.migrations.Migration static method*), 278, 283  
`copy_file()` (*app.utils.file\_storage.FileStorage static*



*method*), 176  
 create() (*app.models.base.Base* class method), 133, 137  
 create() (*app.models.document.Document* class method), 142, 146  
 create() (*app.models.role.Role* class method), 152, 157  
 create() (*app.models.user.User* class method), 164, 171  
 create() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 271  
 create() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 create() (*database.migrations.Migration* class method), 278, 283  
 create\_app() (in module *app*), 260  
 create\_search\_query() (*app.blueprints.base.BaseResource* method), 20, 23  
 create\_search\_query() (*app.blueprints.documents.DocumentBaseResource* method), 26, 36  
 create\_search\_query() (*app.blueprints.documents.DocumentResource* method), 29, 37  
 create\_search\_query() (*app.blueprints.documents.NewDocumentResource* method), 32, 37  
 create\_search\_query() (*app.blueprints.documents.SearchDocumentResource* method), 35, 38  
 create\_search\_query() (*app.blueprints.roles.NewRoleResource* method), 41, 49  
 create\_search\_query() (*app.blueprints.roles.RoleBaseResource* method), 44, 50  
 create\_search\_query() (*app.blueprints.roles.RoleResource* method), 46, 50  
 create\_search\_query() (*app.blueprints.roles.RolesSearchResource* method), 49, 51  
 create\_search\_query() (*app.blueprints.users.ExportUsersExcelAndWordResource* method), 59, 75  
 create\_search\_query() (*app.blueprints.users.ExportUsersExcelResource* method), 62, 76  
 create\_search\_query() (*app.blueprints.users.ExportUsersWordResource* method), 65, 76  
 create\_search\_query() (*app.blueprints.users.NewUserResource* method), 67, 77  
 create\_search\_query() (*app.blueprints.users.UserBaseResource* method), 70, 78  
 create\_search\_query() (*app.blueprints.users.UserResource* method), 72, 78  
 create\_search\_query() (*app.blueprints.users.UsersSearchResource* method), 75, 79  
 create\_search\_query() (in module *app.utils*), 260  
 create\_table() (*app.models.base.Base* class method), 133, 137  
 create\_table() (*app.models.document.Document* class method), 142, 146  
 create\_table() (*app.models.role.Role* class method), 152, 157  
 create\_table() (*app.models.user.User* class method), 165, 171  
 create\_table() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 271  
 create\_table() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 create\_table() (*database.migrations.Migration* class method), 279, 283  
 create\_task\_cls() (*app.celery.MyCelery* method), 107, 124  
 create\_user\_email() (in module *app.celery.tasks*), 81  
 create\_word\_and\_excel\_documents() (in module *app.celery.tasks*), 81  
 created\_at (*app.models.document.Document* attribute), 139, 146  
 created\_at (*app.models.role.Role* attribute), 148, 157  
 created\_at (*app.models.user.User* attribute), 160, 171  
 created\_at (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 271  
 created\_at (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* attribute), 274  
 created\_by (*app.models.document.Document* attribute), 139, 146  
 created\_by (*app.models.user.User* attribute), 160, 171  
 created\_by (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 271  
 created\_by\_id (*app.models.document.Document* attribute), 139, 146  
 created\_by\_id (*app.models.user.User* attribute), 160, 171  
 created\_by\_id (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 271  
 CreateDocumentsTable (class in

[database.migrations.aac\\_create\\_documents\\_table](#) (class in [db\\_model](#) ([app.blueprints.roles.RoleResource](#) attribute), 268, 269  
[CreateUserRolesTable](#) (class in [db\\_model](#) ([app.blueprints.roles.RolesSearchResource](#) attribute), 269, 270  
[current\\_task\(\)](#) ([app.celery.MyCelery](#) property), 98, 124  
[current\\_worker\\_task\(\)](#) ([app.celery.MyCelery](#) property), 99, 124

## D

[database](#)  
   module, 261  
[DATABASE](#) ([config.Config](#) attribute), 304, 327  
[DATABASE](#) ([config.DevConfig](#) attribute), 310, 329  
[DATABASE](#) ([config.ProdConfig](#) attribute), 317, 330  
[DATABASE](#) ([config.TestConfig](#) attribute), 323, 332  
[database.factories](#)  
   module, 261  
[database.migrations](#)  
   module, 266  
[database.migrations.aaa\\_add\\_genre\\_column\\_on\\_user\\_base\\_table](#)  
   module, 266  
[database.migrations.aab\\_add\\_created\\_by\\_column\\_on\\_user\\_base\\_table](#)  
   module, 267  
[database.migrations.aac\\_create\\_documents\\_table](#)  
   module, 268  
[database.migrations.aad\\_create\\_user\\_roles\\_table](#)  
   module, 269  
[database.migrations.aaf\\_remove\\_role\\_slug\\_column\\_on\\_user\\_base\\_table](#)  
   module, 272  
[database.seeds](#)  
   module, 284  
[database.seeds.document\\_seeder](#)  
   module, 284  
[database.seeds.role\\_seeder](#)  
   module, 285  
[database.seeds.user\\_seeder](#)  
   module, 286  
[db\\_model](#) ([app.blueprints.base.BaseResource](#) attribute), 23  
[db\\_model](#) ([app.blueprints.documents.DocumentBaseResource](#) attribute), 36  
[db\\_model](#) ([app.blueprints.documents.DocumentResource](#) attribute), 37  
[db\\_model](#) ([app.blueprints.documents.NewDocumentResource](#) attribute), 37  
[db\\_model](#) ([app.blueprints.documents.SearchDocumentResource](#) attribute), 38  
[db\\_model](#) ([app.blueprints.roles.NewRoleResource](#) attribute), 49  
[db\\_model](#) ([app.blueprints.roles.RoleBaseResource](#) attribute), 50  
[db\\_model](#) ([app.blueprints.roles.RoleResource](#) attribute), 50  
[db\\_model](#) ([app.blueprints.roles.RolesSearchResource](#) attribute), 51  
[db\\_model](#) ([app.blueprints.users.ExportUsersExcelAndWordResource](#) attribute), 75  
[db\\_model](#) ([app.blueprints.users.ExportUsersExcelResource](#) attribute), 76  
[db\\_model](#) ([app.blueprints.users.ExportUsersWordResource](#) attribute), 77  
[db\\_model](#) ([app.blueprints.users.NewUserResource](#) attribute), 77  
[db\\_model](#) ([app.blueprints.users.UserBaseResource](#) attribute), 78  
[db\\_model](#) ([app.blueprints.users.UserResource](#) attribute), 78  
[db\\_model](#) ([app.blueprints.users.UsersSearchResource](#) attribute), 79  
[DEBUG](#) ([config.Config](#) attribute), 304, 327  
[DEBUG](#) ([config.DevConfig](#) attribute), 310, 329  
[DEBUG](#) ([config.ProdConfig](#) attribute), 317, 330  
[DEBUG](#) ([config.TestConfig](#) attribute), 323, 332  
[decorators](#) ([app.blueprints.auth.AuthUserLoginResource](#) attribute), 9, 17  
[decorators](#) ([app.blueprints.auth.AuthUserLogoutResource](#) attribute), 11, 17  
[decorators](#) ([app.blueprints.auth.RequestResetPasswordResource](#) attribute), 13, 18  
[decorators](#) ([app.blueprints.auth.ResetPasswordResource](#) attribute), 15, 18  
[decorators](#) ([app.blueprints.base.BaseResource](#) attribute), 19, 23  
[decorators](#) ([app.blueprints.base.WelcomeResource](#) attribute), 21, 23  
[decorators](#) ([app.blueprints.documents.DocumentBaseResource](#) attribute), 24, 36  
[decorators](#) ([app.blueprints.documents.DocumentResource](#) attribute), 27, 37  
[decorators](#) ([app.blueprints.documents.NewDocumentResource](#) attribute), 31, 38  
[decorators](#) ([app.blueprints.documents.SearchDocumentResource](#) attribute), 34, 38  
[decorators](#) ([app.blueprints.roles.NewRoleResource](#) attribute), 40, 49  
[decorators](#) ([app.blueprints.roles.RoleBaseResource](#) attribute), 42, 50  
[decorators](#) ([app.blueprints.roles.RoleResource](#) attribute), 45, 51  
[decorators](#) ([app.blueprints.roles.RolesSearchResource](#) attribute), 47, 51  
[decorators](#) ([app.blueprints.tasks.TaskResource](#) attribute), 52, 56  
[decorators](#) ([app.blueprints.tasks.TaskStatusResource](#) attribute), 54, 56

decorators (*app.blueprints.users.ExportUsersExcelAndWordResource* attribute), 58, 75  
 decorators (*app.blueprints.users.ExportUsersExcelResource* attribute), 60, 76  
 decorators (*app.blueprints.users.ExportUsersWordResource* attribute), 63, 77  
 decorators (*app.blueprints.users.NewUserResource* attribute), 66, 77  
 decorators (*app.blueprints.users.UserBaseResource* attribute), 68, 78  
 decorators (*app.blueprints.users.UserResource* attribute), 71, 78  
 decorators (*app.blueprints.users.UsersSearchResource* attribute), 73, 79  
 default\_connection() (*app.celery.MyCelery* method), 107, 124  
 default\_error\_messages (*app.utils.marshmallow\_schema.Timestamp* attribute), 205, 240  
 default\_producer() (*app.celery.MyCelery* method), 107, 125  
 default\_retry\_delay (*app.celery.ContextTask* attribute), 83, 115  
 delay() (*app.celery.ContextTask* method), 90, 115  
 delete() (*app.blueprints.documents.DocumentResource* method), 29, 37  
 delete() (*app.blueprints.roles.RoleResource* method), 46, 51  
 delete() (*app.blueprints.users.UserResource* method), 72, 78  
 delete() (*app.models.base.Base* class method), 133, 137  
 delete() (*app.models.document.Document* class method), 142, 146  
 delete() (*app.models.role.Role* class method), 152, 157  
 delete() (*app.models.user.User* class method), 165, 171  
 delete() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 271  
 delete() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 delete() (*database.migrations.Migration* class method), 279, 283  
 delete\_by\_id() (*app.models.base.Base* class method), 133, 137  
 delete\_by\_id() (*app.models.document.Document* class method), 143, 146  
 delete\_by\_id() (*app.models.role.Role* class method), 152, 157  
 delete\_by\_id() (*app.models.user.User* class method), 165, 171  
 delete\_by\_id() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 271  
 delete\_by\_id() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 274  
 delete\_by\_id() (*database.migrations.Migration* class method), 279, 283  
 description (*app.models.role.Role* attribute), 149, 157  
 description (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* attribute), 274  
 deserialize() (*app.utils.marshmallow\_schema.Timestamp* method), 206, 240  
 deserialize\_request\_data() (*app.blueprints.roles.NewRoleResource* method), 41, 49

`deserialize_request_data()`  
     (`app.blueprints.roles.RoleBaseResource`  
     *method*), 44, 50  
`deserialize_request_data()`  
     (`app.blueprints.roles.RoleResource` *method*),  
     46, 51  
`deserialize_request_data()`  
     (`app.blueprints.roles.RolesSearchResource`  
     *method*), 49, 51  
`deserialize_request_data()`  
     (`app.blueprints.users.ExportUsersExcelAndWordResource` *property*), 274  
     *method*), 59, 75  
`deserialize_request_data()`  
     (`app.blueprints.users.ExportUsersExcelResource` *method*), 62, 76  
`deserialize_request_data()`  
     (`app.blueprints.users.ExportUsersWordResource` *method*), 65, 77  
`deserialize_request_data()`  
     (`app.blueprints.users.NewUserResource`  
     *method*), 67, 77  
`deserialize_request_data()`  
     (`app.blueprints.users.UserBaseResource`  
     *method*), 70, 78  
`deserialize_request_data()`  
     (`app.blueprints.users.UserResource` *method*),  
     72, 78  
`deserialize_request_data()`  
     (`app.blueprints.users.UsersSearchResource`  
     *method*), 75, 79  
`DevConfig` (*class in* `config`), 308, 329  
`DEVELOPMENT` (`config.Config` *attribute*), 304, 327  
`DEVELOPMENT` (`config.DevConfig` *attribute*), 310, 329  
`DEVELOPMENT` (`config.ProdConfig` *attribute*), 317, 330  
`DEVELOPMENT` (`config.TestConfig` *attribute*), 323, 332  
`dict_class()` (`app.utils.marshmallow_schema._SearchOrderSchema`  
     *property*), 249  
`dict_class()` (`app.utils.marshmallow_schema._SearchValueSchema`  
     *property*), 254  
`dict_class()` (`app.utils.marshmallow_schema.DocumentSchema` *method*), 32, 38  
     *property*), 178, 215  
`dict_class()` (`app.utils.marshmallow_schema.ExportWordInputSchema`  
     *property*), 184, 220  
`dict_class()` (`app.utils.marshmallow_schema.GetDocumentDataInputSchema`  
     *property*), 190, 226  
`dict_class()` (`app.utils.marshmallow_schema.RoleSchema`  
     *property*), 195, 231  
`dict_class()` (`app.utils.marshmallow_schema.SearchSchema`  
     *property*), 200, 236  
`dict_class()` (`app.utils.marshmallow_schema.UserSchema`  
     *property*), 208, 243  
`directory_path` (`app.models.document.Document`  
     *attribute*), 139, 147  
`dirty_fields()` (`app.models.base.Base` *property*),  
     131, 137  
`dirty_fields()` (`app.models.document.Document`  
     *property*), 139, 147  
`dirty_fields()` (`app.models.role.Role` *property*),  
     149, 157  
`dirty_fields()` (`app.models.user.User` *property*),  
     160, 171  
`dirty_fields()` (`database.migrations.aad_create_user_roles_table._C`  
     *property*), 271  
`dirty_fields()` (`database.migrations.aaf_remove_role_slug_column._C`  
     *property*), 276, 283  
`dispatch_request()`  
     (`app.blueprints.auth.AuthUserLoginResource`  
     *method*), 10, 17  
`dispatch_request()`  
     (`app.blueprints.auth.AuthUserLogoutResource`  
     *method*), 12, 17  
`dispatch_request()`  
     (`app.blueprints.auth.RequestResetPasswordResource`  
     *method*), 14, 18  
`dispatch_request()`  
     (`app.blueprints.auth.ResetPasswordResource`  
     *method*), 16, 18  
`dispatch_request()`  
     (`app.blueprints.base.BaseResource` *method*),  
     20, 23  
`dispatch_request()`  
     (`app.blueprints.base.WelcomeResource`  
     *method*), 22, 23  
`dispatch_request()`  
     (`app.blueprints.documents.DocumentBaseResource`  
     *method*), 26, 36  
`dispatch_request()`  
     (`app.blueprints.documents.DocumentResource`  
     *method*), 29, 37  
`dispatch_request()`  
     (`app.blueprints.documents.NewDocumentResource`  
     *method*), 32, 38  
`dispatch_request()`  
     (`app.blueprints.documents.SearchDocumentResource`  
     *method*), 35, 38  
`dispatch_request()`  
     (`app.blueprints.roles.NewRoleResource`  
     *method*), 41, 49  
`dispatch_request()`  
     (`app.blueprints.roles.RoleBaseResource`  
     *method*), 44, 50  
`dispatch_request()`  
     (`app.blueprints.roles.RoleResource` *method*),  
     46, 51  
`dispatch_request()`  
     (`app.blueprints.roles.RolesSearchResource`



*method*), 49, 51  
 dispatch\_request ()  
     (*app.blueprints.tasks.TaskResource method*), 53, 56  
 dispatch\_request ()  
     (*app.blueprints.tasks.TaskStatusResource method*), 56, 57  
 dispatch\_request ()  
     (*app.blueprints.users.ExportUsersExcelAndWordResource method*), 59, 75  
 dispatch\_request ()  
     (*app.blueprints.users.ExportUsersExcelResource method*), 62, 76  
 dispatch\_request ()  
     (*app.blueprints.users.ExportUsersWordResource method*), 65, 77  
 dispatch\_request ()  
     (*app.blueprints.users.NewUserResource method*), 67, 77  
 dispatch\_request ()  
     (*app.blueprints.users.UserBaseResource method*), 70, 78  
 dispatch\_request ()  
     (*app.blueprints.users.UserResource method*), 72, 78  
 dispatch\_request ()  
     (*app.blueprints.users.UsersSearchResource method*), 75, 79  
 Document (class in *app.models.document*), 138, 146  
 document\_serializer  
     (*app.blueprints.documents.DocumentBaseResource attribute*), 25, 36  
 document\_serializer  
     (*app.blueprints.documents.DocumentResource attribute*), 28, 37  
 document\_serializer  
     (*app.blueprints.documents.NewDocumentResource attribute*), 31, 38  
 document\_serializer  
     (*app.blueprints.documents.SearchDocumentResource attribute*), 34, 38  
 document\_set (*app.models.user.User attribute*), 160, 171  
 DocumentBaseResource (class in *app.blueprints.documents*), 24, 36  
 DocumentResource (class in *app.blueprints.documents*), 27, 37  
 DocumentSchema (class in *app.utils.marshmallow\_schema*), 178, 213  
 DocumentSchema.Meta (class in *app.utils.marshmallow\_schema*), 213  
 DocumentSeeder (class in *database.seeds.document\_seeder*), 285  
 DoesNotExist (*app.models.base.Base attribute*), 136  
 DoesNotExist (*app.models.document.Document attribute*), 146  
 DoesNotExist (*app.models.role.Role attribute*), 156  
 DoesNotExist (*app.models.user.User attribute*), 170  
 DoesNotExist (*database.migrations.aad\_create\_user\_roles\_table.Old class method*), 270  
 DoesNotExist (*database.migrations.aaf\_remove\_role\_slug\_column.Old class method*), 273  
 DoesNotExist (*database.migrations.Migration class method*), 282  
 down () (*database.migrations.aaa\_add\_genre\_column\_on\_user\_table.Add class method*), 267  
 down () (*database.migrations.aab\_add\_created\_by\_column\_on\_user\_table.Add class method*), 268  
 down () (*database.migrations.aac\_create\_documents\_table.CreateDocuments class method*), 269  
 down () (*database.migrations.aad\_create\_user\_roles\_table.CreateUserRoles class method*), 270  
 down () (*database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoles class method*), 273  
 drop\_table () (*app.models.base.Base class method*), 134, 137  
 drop\_table () (*app.models.document.Document class method*), 143, 147  
 drop\_table () (*app.models.role.Role class method*), 153, 157  
 drop\_table () (*app.models.user.User class method*), 165, 171  
 drop\_table () (*database.migrations.aad\_create\_user\_roles\_table.Old class method*), 271  
 drop\_table () (*database.migrations.aaf\_remove\_role\_slug\_column.Old class method*), 274  
 drop\_table () (*database.migrations.Migration class method*), 279, 283  
 dump () (*app.utils.marshmallow\_schema.\_SearchOrderSchema method*), 249  
 dump () (*app.utils.marshmallow\_schema.\_SearchValueSchema method*), 254  
 dump () (*app.utils.marshmallow\_schema.DocumentSchema method*), 180, 215  
 dump () (*app.utils.marshmallow\_schema.ExportWordInputSchema method*), 185, 220  
 dump () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema method*), 191, 226  
 dump () (*app.utils.marshmallow\_schema.RoleSchema method*), 196, 231  
 dump () (*app.utils.marshmallow\_schema.SearchSchema method*), 201, 236  
 dump () (*app.utils.marshmallow\_schema.UserSchema method*), 209, 243  
 dumps () (*app.utils.marshmallow\_schema.\_SearchOrderSchema method*), 249  
 dumps () (*app.utils.marshmallow\_schema.\_SearchValueSchema method*), 255

dumps () (*app.utils.marshmallow\_schema.DocumentSchema* *export\_user\_data\_in\_word()* (in module *method*), 180, 215 *app.celery.word.tasks*), 81  
 dumps () (*app.utils.marshmallow\_schema.ExportWordInputSchema* *UsersExcelAndWordResource* (class in *method*), 186, 221 *app.blueprints.users*), 57, 75  
 dumps () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* *ExcelResource* (class in *method*), 191, 226 *app.blueprints.users*), 60, 76  
 dumps () (*app.utils.marshmallow\_schema.RoleSchema* *ExportUsersWordResource* (class in *method*), 196, 231 *app.blueprints.users*), 63, 76  
 dumps () (*app.utils.marshmallow\_schema.SearchSchema* *ExportWordInputSchema* (class in *method*), 201, 237 *app.utils.marshmallow\_schema*), 183, 217  
 dumps () (*app.utils.marshmallow\_schema.UserSchema* *ExportWordInputSchema.Meta* (class in *method*), 209, 243 *app.utils.marshmallow\_schema*), 218

## E

either () (*app.celery.MyCelery* *method*), 107, 125  
 email (*app.models.user.User* *attribute*), 160, 171  
 email (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* *attribute*), 271  
 enable\_utc (*config.Config* *attribute*), 307, 328  
 enable\_utc (*config.DevConfig* *attribute*), 313, 330  
 enable\_utc (*config.ProdConfig* *attribute*), 320, 331  
 enable\_utc (*config.TestConfig* *attribute*), 326, 332  
 ensure\_password () (*app.models.user.User* *static method*), 165, 171  
 errno (*app.utils.FileEmptyError* *attribute*), 259  
 ERROR\_404\_HELP (*config.Config* *attribute*), 304, 327  
 ERROR\_404\_HELP (*config.DevConfig* *attribute*), 310, 329  
 ERROR\_404\_HELP (*config.ProdConfig* *attribute*), 317, 330  
 ERROR\_404\_HELP (*config.TestConfig* *attribute*), 323, 332  
 error\_messages (*app.utils.marshmallow\_schema.\_SearchOrderSchema* *attribute*), 249  
 error\_messages (*app.utils.marshmallow\_schema.\_SearchValueSchema* *attribute*), 255  
 error\_messages (*app.utils.marshmallow\_schema.DocumentSchema* *attribute*), 178, 215  
 error\_messages (*app.utils.marshmallow\_schema.ExportWordInputSchema* *attribute*), 184, 221  
 error\_messages (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* *attribute*), 190, 227  
 error\_messages (*app.utils.marshmallow\_schema.RoleSchema* *attribute*), 195, 231  
 error\_messages (*app.utils.marshmallow\_schema.SearchSchema* *attribute*), 200, 237  
 error\_messages (*app.utils.marshmallow\_schema.UserSchema* *attribute*), 208, 244  
 events (*app.celery.MyCelery* *attribute*), 99, 125  
 events\_cls (*app.celery.MyCelery* *attribute*), 99, 125  
 expires (*app.celery.ContextTask* *attribute*), 84, 115  
 export\_user\_data\_in\_excel () (in module *app.celery.excel.tasks*), 80

## F

Factory (class in *database.factories*), 262, 264  
 factory () (in module *tests.conftest*), 300  
 fields (*app.utils.marshmallow\_schema.DocumentSchema.Meta* *attribute*), 213  
 fields (*app.utils.marshmallow\_schema.RoleSchema.Meta* *attribute*), 229  
 fields (*app.utils.marshmallow\_schema.UserSchema.Meta* *attribute*), 241  
 FileEmptyError, 259  
 filename (*app.utils.FileEmptyError* *attribute*), 259  
 filename2 (*app.utils.FileEmptyError* *attribute*), 259  
 FileStorage (class in *app.utils.file\_storage*), 175, 176  
 filter () (*app.models.base.Base* *class method*), 134, 137  
 filter () (*app.models.document.Document* *class method*), 143, 147  
 filter\_order (*app.models.role.Role* *class method*), 153, 157  
 filter\_value (*app.models.user.User* *class method*), 165, 171  
 find\_in\_schema (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* *class method*), 271  
 find\_in\_schema (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* *class method*), 274  
 find\_in\_schema (*database.migrations.Migration* *class method*), 279, 283  
 find\_size () (*app.celery.MyCelery* *method*), 107, 125  
 find\_longest\_word () (in module *app.utils*), 258,  
 FLASK\_RESTFUL\_PREFIX (*config.Config* *attribute*), 304, 327  
 FLASK\_RESTFUL\_PREFIX (*config.DevConfig* *attribute*), 310, 329  
 FLASK\_RESTFUL\_PREFIX (*config.ProdConfig* *attribute*), 317, 330  
 FLASK\_RESTFUL\_PREFIX (*config.TestConfig* *attribute*), 323, 332

from\_config (*app.celery.ContextTask attribute*), 84, *get\_attribute()* (*app.utils.marshmallow\_schema.ExportWordInputSchema class method*), 186, 221  
 from\_dict () (*app.utils.marshmallow\_schema.\_SearchOrderSchema attribute*), 192, 227  
 from\_dict () (*app.utils.marshmallow\_schema.\_SearchValueSchema attribute*), 197, 232  
 from\_dict () (*app.utils.marshmallow\_schema.DocumentSchema attribute*), 202, 237  
 from\_dict () (*app.utils.marshmallow\_schema.ExportWordInputSchema attribute*), 210, 244  
 from\_dict () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema attribute*), 166, 171  
 from\_dict () (*app.utils.marshmallow\_schema.RoleSchema attribute*), 176  
 from\_dict () (*app.utils.marshmallow\_schema.SearchSchema attribute*), 134, 137  
 from\_dict () (*app.utils.marshmallow\_schema.UserSchema attribute*), 143, 147  
 G  
 gen\_task\_name () (*app.celery.MyCelery method*), 108, 125  
 genre (*app.models.user.User attribute*), 160, 171  
 genre (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser class method*), 271  
 get () (*app.blueprints.auth.ResetPasswordResource method*), 16, 18  
 get () (*app.blueprints.base.WelcomeResource method*), 22, 23  
 get () (*app.blueprints.documents.DocumentResource method*), 29, 37  
 get () (*app.blueprints.roles.RoleResource method*), 46, 51  
 get () (*app.blueprints.tasks.TaskStatusResource method*), 56, 57  
 get () (*app.blueprints.users.UserResource method*), 72, 78  
 get () (*app.models.base.Base class method*), 134, 137  
 get () (*app.models.document.Document class method*), 143, 147  
 get () (*app.models.role.Role class method*), 153, 157  
 get () (*app.models.user.User class method*), 165, 171  
 get () (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser class method*), 271  
 get () (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole class method*), 274  
 get () (*database.migrations.Migration class method*), 279, 283  
 get\_attribute () (*app.utils.marshmallow\_schema.\_SearchOrderSchema method*), 250  
 get\_attribute () (*app.utils.marshmallow\_schema.\_SearchValueSchema method*), 255  
 get\_attribute () (*app.utils.marshmallow\_schema.DocumentSchema method*), 181, 216  
 get\_attribute () (*app.utils.marshmallow\_schema.ExportWordInputSchema method*), 210, 244  
 get\_attribute () (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema method*), 166, 171  
 get\_attribute () (*app.utils.marshmallow\_schema.RoleSchema method*), 176  
 get\_attribute () (*app.utils.marshmallow\_schema.SearchSchema method*), 134, 137  
 get\_attribute () (*app.utils.marshmallow\_schema.UserSchema method*), 143, 147  
 get\_by\_id () (*app.models.base.Base class method*), 134, 137  
 get\_by\_id () (*app.models.role.Role class method*), 153, 157  
 get\_by\_id () (*app.models.user.User class method*), 166, 171  
 get\_by\_id () (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser class method*), 271  
 get\_by\_id () (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole class method*), 274  
 get\_by\_id () (*database.migrations.Migration class method*), 279, 283  
 get\_document\_content () (*app.blueprints.documents.DocumentBaseResource static method*), 26, 36  
 get\_document\_content () (*app.blueprints.documents.DocumentResource static method*), 29, 37  
 get\_document\_content () (*app.blueprints.documents.NewDocumentResource static method*), 32, 38  
 get\_document\_content () (*app.blueprints.documents.SearchDocumentResource static method*), 35, 38  
 get\_document\_data () (*app.blueprints.documents.DocumentBaseResource static method*), 26, 36  
 get\_document\_data () (*app.blueprints.documents.DocumentResource static method*), 30, 37  
 get\_document\_data () (*app.blueprints.documents.NewDocumentResource static method*), 33, 38  
 get\_document\_data () (*app.blueprints.documents.SearchDocumentResource static method*), 35, 38  
 get\_fields () (*app.models.base.Base class method*), 134, 137

|                                                                                                                    |                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <code>get_fields()</code> ( <i>app.models.document.Document class method</i> ), 143, 147                           | <code>get_or_none()</code> ( <i>database.migrations.Migration class method</i> ), 280, 283                             |
| <code>get_fields()</code> ( <i>app.models.role.Role class method</i> ), 153, 157                                   | <code>get_permissions()</code> ( <i>app.models.role.Role method</i> ), 154, 157                                        |
| <code>get_fields()</code> ( <i>app.models.user.User class method</i> ), 166, 171                                   | <code>get_permissions()</code> ( <i>database.migrations.aaf_remove_role_slug_column._OldRole class method</i> ), 274   |
| <code>get_fields()</code> ( <i>database.migrations.aad_create_user_roles_table._OldUser class method</i> ), 271    | <code>get_redirect_qparams()</code> ( <i>app.models.user.User class method</i> ), 166, 172                             |
| <code>get_fields()</code> ( <i>database.migrations.aaf_remove_role_slug_column._OldRole class method</i> ), 274    | <code>get_request_file()</code> ( <i>app.blueprints.documents.DocumentBaseResource method</i> ), 26, 36                |
| <code>get_filepath()</code> ( <i>app.models.document.Document method</i> ), 143, 147                               | <code>get_request_file()</code> ( <i>app.blueprints.documents.DocumentResource method</i> ), 30, 37                    |
| <code>get_filesize()</code> ( <i>app.utils.file_storage.FileStorage static method</i> ), 176                       | <code>get_request_file()</code> ( <i>app.blueprints.documents.NewDocumentResource method</i> ), 33, 38                 |
| <code>get_id()</code> ( <i>app.models.base.Base method</i> ), 134, 137                                             | <code>get_request_file()</code> ( <i>app.blueprints.documents.SearchDocumentResource method</i> ), 36, 38              |
| <code>get_id()</code> ( <i>app.models.document.Document method</i> ), 144, 147                                     | <code>get_request_query_fields()</code> ( <i>app.blueprints.base.BaseResource method</i> ), 21, 23                     |
| <code>get_id()</code> ( <i>app.models.role.Role method</i> ), 153, 157                                             | <code>get_request_query_fields()</code> ( <i>app.blueprints.documents.DocumentBaseResource method</i> ), 27, 36        |
| <code>get_id()</code> ( <i>app.models.user.User method</i> ), 166, 171                                             | <code>get_request_query_fields()</code> ( <i>app.blueprints.documents.DocumentResource method</i> ), 30, 37            |
| <code>get_id()</code> ( <i>database.migrations.aad_create_user_roles_table._OldUser method</i> ), 271              | <code>get_request_query_fields()</code> ( <i>app.blueprints.documents.NewDocumentResource method</i> ), 33, 38         |
| <code>get_id()</code> ( <i>database.migrations.aaf_remove_role_slug_column._OldRole method</i> ), 274              | <code>get_request_query_fields()</code> ( <i>app.blueprints.documents.SearchDocumentResource method</i> ), 36, 38      |
| <code>get_id()</code> ( <i>database.migrations.Migration method</i> ), 280, 283                                    | <code>get_request_query_fields()</code> ( <i>app.blueprints.roles.NewRoleResource method</i> ), 44, 50                 |
| <code>get_migration_names()</code> ( <i>in module database.migrations</i> ), 282, 284                              | <code>get_request_query_fields()</code> ( <i>app.blueprints.roles.RoleBaseResource method</i> ), 44, 50                |
| <code>get_models()</code> ( <i>in module app.models</i> ), 174                                                     | <code>get_request_query_fields()</code> ( <i>app.blueprints.roles.RoleResource method</i> ), 46, 51                    |
| <code>get_or_create()</code> ( <i>app.models.base.Base class method</i> ), 134, 137                                | <code>get_request_query_fields()</code> ( <i>app.blueprints.roles.RolesSearchResource method</i> ), 49, 51             |
| <code>get_or_create()</code> ( <i>app.models.document.Document class method</i> ), 144, 147                        | <code>get_request_query_fields()</code> ( <i>app.blueprints.users.ExportUsersExcelAndWordResource method</i> ), 60, 75 |
| <code>get_or_create()</code> ( <i>app.models.role.Role class method</i> ), 153, 157                                | <code>get_request_query_fields()</code> ( <i>app.blueprints.users.ExportUsersExcelResource method</i> ), 62, 76        |
| <code>get_or_create()</code> ( <i>app.models.user.User class method</i> ), 166, 172                                |                                                                                                                        |
| <code>get_or_create()</code> ( <i>database.migrations.aad_create_user_roles_table._OldUser class method</i> ), 271 |                                                                                                                        |
| <code>get_or_create()</code> ( <i>database.migrations.aaf_remove_role_slug_column._OldRole class method</i> ), 274 |                                                                                                                        |
| <code>get_or_create()</code> ( <i>database.migrations.Migration class method</i> ), 280, 283                       |                                                                                                                        |
| <code>get_or_none()</code> ( <i>app.models.base.Base class method</i> ), 134, 137                                  |                                                                                                                        |
| <code>get_or_none()</code> ( <i>app.models.document.Document class method</i> ), 144, 147                          |                                                                                                                        |
| <code>get_or_none()</code> ( <i>app.models.role.Role class method</i> ), 154, 157                                  |                                                                                                                        |
| <code>get_or_none()</code> ( <i>app.models.user.User class method</i> ), 166, 172                                  |                                                                                                                        |
| <code>get_or_none()</code> ( <i>database.migrations.aad_create_user_roles_table._OldUser class method</i> ), 271   |                                                                                                                        |
| <code>get_or_none()</code> ( <i>database.migrations.aaf_remove_role_slug_column._OldRole class method</i> ), 274   |                                                                                                                        |



`get_request_query_fields()`  
     (`app.blueprints.users.ExportUsersWordResource`  
     *method*), 65, 77  
`get_request_query_fields()`  
     (`app.blueprints.users.NewUserResource`  
     *method*), 67, 77  
`get_request_query_fields()`  
     (`app.blueprints.users.UserBaseResource`  
     *method*), 70, 78  
`get_request_query_fields()`  
     (`app.blueprints.users.UserResource` *method*),  
     72, 79  
`get_request_query_fields()`  
     (`app.blueprints.users.UsersSearchResource`  
     *method*), 75, 79  
`get_request_query_fields()` (in module  
     `app.utils`), 258, 260  
`get_reset_token()` (`app.models.user.User`  
     *method*), 167, 172  
`get_security_payload()` (`app.models.user.User`  
     *method*), 167, 172  
`get_seeders()` (in module `database.seeds`), 287, 288  
`get_task()` (`app.blueprints.tasks.TaskResource` *static*  
     *method*), 54, 56  
`get_task()` (`app.blueprints.tasks.TaskStatusResource`  
     *static method*), 56, 57  
`get_value()` (`app.utils.marshmallow_schema.Timestamp`  
     *method*), 206, 240  
`GetDocumentDataInputSchema` (class in  
     `app.utils.marshmallow_schema`), 189, 223  
`GetDocumentDataInputSchema.Meta` (class in  
     `app.utils.marshmallow_schema`), 223  
`GroupResult` (`app.celery.MyCelery` attribute), 96, 119

## H

`handle_error()` (`app.utils.marshmallow_schema.SearchOrderSchema`  
     *method*), 250  
`handle_error()` (`app.utils.marshmallow_schema.SearchValueSchema`  
     *method*), 255  
`handle_error()` (`app.utils.marshmallow_schema.DocumentSchema`  
     *method*), 181, 216  
`handle_error()` (`app.utils.marshmallow_schema.ExportWordInputSchema`  
     *method*), 187, 221  
`handle_error()` (`app.utils.marshmallow_schema.GetDocumentDataInputSchema`  
     *method*), 192, 227  
`handle_error()` (`app.utils.marshmallow_schema.RoleSchema` *static*  
     *method*), 197, 232  
`handle_error()` (`app.utils.marshmallow_schema.SearchSchema` *static*  
     *method*), 202, 237  
`handle_error()` (`app.utils.marshmallow_schema.UserSchema` *static*  
     *method*), 210, 244  
`has_permission()` (`app.models.user.User` *method*),  
     167, 172  
`has_role()` (`app.models.user.User` *method*), 167, 172  
`HOME` (`config.Config` attribute), 304, 327  
`HOME` (`config.DevConfig` attribute), 310, 329  
`HOME` (`config.ProdConfig` attribute), 317, 330  
`HOME` (`config.TestConfig` attribute), 323, 332

## I

`id` (`app.models.base.Base` attribute), 131, 137  
`id` (`app.models.document.Document` attribute), 139, 147  
`id` (`app.models.role.Role` attribute), 149, 157  
`id` (`app.models.user.User` attribute), 160, 172  
`id` (`database.migrations.aad_create_user_roles_table._OldUser`  
     attribute), 271  
`id` (`database.migrations.aaf_remove_role_slug_column._OldRole`  
     attribute), 274  
`id` (`database.migrations.Migration` attribute), 276, 283  
`ignore_keys()` (in module `app.utils`), 258, 260  
`ignore_result` (`app.celery.ContextTask` attribute),  
     84, 115  
`include` (`config.Config` attribute), 307, 328  
`include` (`config.DevConfig` attribute), 313, 330  
`include` (`config.ProdConfig` attribute), 320, 331  
`include` (`config.TestConfig` attribute), 326, 332  
`index()` (`app.models.base.Base` class *method*), 134,  
     137  
`index()` (`app.models.document.Document` class  
     *method*), 144, 147  
`index()` (`app.models.role.Role` class *method*), 154, 157  
`index()` (`app.models.user.User` class *method*), 167, 172  
`index()` (`database.migrations.aad_create_user_roles_table._OldUser`  
     class *method*), 271  
`index()` (`database.migrations.aaf_remove_role_slug_column._OldRole`  
     class *method*), 274  
`index()` (`database.migrations.Migration` class  
     *method*), 280, 283  
`init_app()` (in module `app.extensions`), 129  
`init_order_schema()` (in module `app.celery`), 111, 128  
`init_database()` (in module `database`), 288  
`init_value_schema()` (in module  
     `database.migrations`), 282, 284  
`insert_schema()` (in module `database.seeds`), 288  
`insert()` (`app.models.base.Base` class *method*), 135,  
     137  
`insert()` (`app.models.document.Document` class  
     *method*), 144, 147  
`insert()` (`app.models.role.Role` class *method*), 154,  
     157  
`insert()` (`app.models.user.User` class *method*), 167,  
     172  
`insert()` (`database.migrations.aad_create_user_roles_table._OldUser`  
     class *method*), 271  
`insert()` (`database.migrations.aaf_remove_role_slug_column._OldRole`  
     class *method*), 274  
`insert()` (`database.migrations.Migration` class  
     *method*), 280, 283

`insert_from()` (*app.models.base.Base* class method), 135, 137  
`insert_from()` (*app.models.document.Document* class method), 144, 147  
`insert_from()` (*app.models.role.Role* class method), 154, 157  
`insert_from()` (*app.models.user.User* class method), 167, 172  
`insert_from()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 271  
`insert_from()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 274  
`insert_from()` (*database.migrations.Migration* class method), 280, 283  
`insert_many()` (*app.models.base.Base* class method), 135, 137  
`insert_many()` (*app.models.document.Document* class method), 144, 147  
`insert_many()` (*app.models.role.Role* class method), 154, 157  
`insert_many()` (*app.models.user.User* class method), 167, 172  
`insert_many()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 271  
`insert_many()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 274  
`insert_many()` (*database.migrations.Migration* class method), 280, 283  
`internal_filename` (*app.models.document.Document* attribute), 139, 147  
`is_active()` (*app.models.user.User* property), 161, 172  
`is_active()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* property), 271  
`is_alias()` (*app.models.base.Base* method), 135, 137  
`is_alias()` (*app.models.document.Document* method), 144, 147  
`is_alias()` (*app.models.role.Role* method), 154, 157  
`is_alias()` (*app.models.user.User* method), 168, 172  
`is_alias()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* method), 271  
`is_alias()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* method), 274  
`is_alias()` (*database.migrations.Migration* method), 280, 283  
`is_anonymous()` (*app.models.user.User* property), 161, 172  
`is_anonymous()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* property), 271  
`is_authenticated()` (*app.models.user.User* property), 161, 172  
`is_authenticated()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* property), 271  
`is_dirty()` (*app.models.base.Base* method), 135, 137  
`is_dirty()` (*app.models.document.Document* method), 144, 147  
`is_dirty()` (*app.models.role.Role* method), 154, 157  
`is_dirty()` (*app.models.user.User* method), 168, 172  
`is_dirty()` (*database.migrations.aad\_create\_user\_roles\_table.OldUser* method), 271  
`is_dirty()` (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* method), 275  
`is_dirty()` (*database.migrations.Migration* method), 280, 283  
`IS_macOS` (*app.celery.MyCelery* attribute), 97, 120  
`IS_WINDOWS` (*app.celery.MyCelery* attribute), 97, 119

## J

`jsonify()` (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 250  
`jsonify()` (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 256  
`jsonify()` (*app.utils.marshmallow\_schema.DocumentSchema* method), 181, 216  
`jsonify()` (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 187, 222  
`jsonify()` (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 193, 227  
`jsonify()` (*app.utils.marshmallow\_schema.RoleSchema* method), 198, 232  
`jsonify()` (*app.utils.marshmallow\_schema.SearchSchema* method), 203, 238  
`jsonify()` (*app.utils.marshmallow\_schema.UserSchema* method), 211, 244

- `load()` (*app.utils.marshmallow\_schema.SearchSchema* method), 203, 238
- `load()` (*app.utils.marshmallow\_schema.UserSchema* method), 211, 245
- `loader` (*app.celery.MyCelery* attribute), 99, 125
- `loader_cls` (*app.celery.MyCelery* attribute), 99, 125
- `loads()` (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 251
- `loads()` (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 256
- `loads()` (*app.utils.marshmallow\_schema.DocumentSchema* method), 182, 217
- `loads()` (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 188, 222
- `loads()` (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 193, 228
- `loads()` (*app.utils.marshmallow\_schema.RoleSchema* method), 198, 233
- `loads()` (*app.utils.marshmallow\_schema.SearchSchema* method), 203, 238
- `loads()` (*app.utils.marshmallow\_schema.UserSchema* method), 211, 245
- `log` (*app.celery.MyCelery* attribute), 99, 125
- `log_cls` (*app.celery.MyCelery* attribute), 99, 125
- `LOG_DIRECTORY` (*config.Config* attribute), 304, 327
- `LOG_DIRECTORY` (*config.DevConfig* attribute), 310, 329
- `LOG_DIRECTORY` (*config.ProdConfig* attribute), 317, 330
- `LOG_DIRECTORY` (*config.TestConfig* attribute), 323, 332
- `LOGIN_DISABLED` (*config.Config* attribute), 304, 327
- `LOGIN_DISABLED` (*config.DevConfig* attribute), 310, 329
- `LOGIN_DISABLED` (*config.ProdConfig* attribute), 317, 330
- `LOGIN_DISABLED` (*config.TestConfig* attribute), 323, 332
- M**
- `MAIL_PASSWORD` (*config.Config* attribute), 304, 328
- `MAIL_PASSWORD` (*config.DevConfig* attribute), 311, 329
- `MAIL_PASSWORD` (*config.ProdConfig* attribute), 317, 330
- `MAIL_PASSWORD` (*config.TestConfig* attribute), 324, 332
- `MAIL_PORT` (*config.Config* attribute), 305, 328
- `MAIL_PORT` (*config.DevConfig* attribute), 311, 329
- `MAIL_PORT` (*config.ProdConfig* attribute), 318, 330
- `MAIL_PORT` (*config.TestConfig* attribute), 324, 332
- `MAIL_SERVER` (*config.Config* attribute), 305, 328
- `MAIL_SERVER` (*config.DevConfig* attribute), 311, 329
- `MAIL_SERVER` (*config.ProdConfig* attribute), 318, 330
- `MAIL_SERVER` (*config.TestConfig* attribute), 324, 332
- `MAIL_USE_SSL` (*config.Config* attribute), 305, 328
- `MAIL_USE_SSL` (*config.DevConfig* attribute), 311, 329
- `MAIL_USE_SSL` (*config.ProdConfig* attribute), 318, 331
- `MAIL_USE_SSL` (*config.TestConfig* attribute), 324, 332
- `MAIL_USE_TLS` (*config.Config* attribute), 305, 328
- `MAIL_USE_TLS` (*config.DevConfig* attribute), 311, 329
- `MAIL_USE_TLS` (*config.ProdConfig* attribute), 318, 331
- `MAIL_USE_TLS` (*config.TestConfig* attribute), 324, 332
- `MAIL_USERNAME` (*config.Config* attribute), 305, 328
- `MAIL_USERNAME` (*config.DevConfig* attribute), 311, 329
- `MAIL_USERNAME` (*config.ProdConfig* attribute), 318, 330
- `MAIL_USERNAME` (*config.TestConfig* attribute), 324, 332
- `main` (*app.celery.MyCelery* attribute), 99, 125
- `make()` (*database.factories.Factory* method), 262, 264, 265
- `make_error()` (*app.utils.marshmallow\_schema.Timestamp* method), 207, 241
- `make_url()` (*app.utils.marshmallow\_schema.DocumentSchema* method), 182, 217
- `map()` (*app.celery.ContextTask* method), 90, 115
- `max_retries` (*app.celery.ContextTask* attribute), 84, 115
- `Meta` (class in *config*), 315, 330
- `method_decorators` (*app.blueprints.auth.AuthUserLoginResource* attribute), 9, 17
- `method_decorators` (*app.blueprints.auth.AuthUserLogoutResource* attribute), 11, 17
- `method_decorators` (*app.blueprints.auth.RequestResetPasswordResource* attribute), 13, 18
- `method_decorators` (*app.blueprints.auth.ResetPasswordResource* attribute), 15, 18
- `method_decorators` (*app.blueprints.base.BaseResource* attribute), 19, 23
- `method_decorators` (*app.blueprints.base.WelcomeResource* attribute), 21, 23
- `method_decorators` (*app.blueprints.documents.DocumentBaseResource* attribute), 25, 36
- `method_decorators` (*app.blueprints.documents.DocumentResource* attribute), 28, 37
- `method_decorators` (*app.blueprints.documents.NewDocumentResource* attribute), 31, 38

- method\_decorators (app.blueprints.documents.SearchDocumentResource attribute), 34, 38
- method\_decorators (app.blueprints.roles.NewRoleResource attribute), 40, 50
- method\_decorators (app.blueprints.roles.RoleBaseResource attribute), 42, 50
- method\_decorators (app.blueprints.roles.RoleResource attribute), 45, 51
- method\_decorators (app.blueprints.roles.RolesSearchResource attribute), 47, 51
- method\_decorators (app.blueprints.tasks.TaskResource attribute), 52, 56
- method\_decorators (app.blueprints.tasks.TaskStatusResource attribute), 54, 57
- method\_decorators (app.blueprints.users.ExportUsersExcelAndWordResource attribute), 58, 76
- method\_decorators (app.blueprints.users.ExportUsersExcelResource attribute), 61, 76
- method\_decorators (app.blueprints.users.ExportUsersWordResource attribute), 63, 77
- method\_decorators (app.blueprints.users.NewUserResource attribute), 66, 77
- method\_decorators (app.blueprints.users.UserBaseResource attribute), 68, 78
- method\_decorators (app.blueprints.users.UserResource attribute), 71, 79
- method\_decorators (app.blueprints.users.UsersSearchResource attribute), 73, 79
- methods (app.blueprints.auth.AuthUserLoginResource attribute), 9, 17
- methods (app.blueprints.auth.AuthUserLogoutResource attribute), 11, 17
- methods (app.blueprints.auth.RequestResetPasswordResource attribute), 13, 18
- methods (app.blueprints.auth.ResetPasswordResource attribute), 15, 18
- methods (app.blueprints.base.BaseResource attribute), 19, 23
- methods (app.blueprints.base.WelcomeResource attribute), 21, 23
- methods (app.blueprints.documents.DocumentBaseResource attribute), 25, 36
- methods (app.blueprints.documents.DocumentResource attribute), 28, 37
- methods (app.blueprints.documents.NewDocumentResource attribute), 31, 38
- methods (app.blueprints.documents.SearchDocumentResource attribute), 34, 38
- methods (app.blueprints.roles.NewRoleResource attribute), 40, 50
- methods (app.blueprints.roles.RoleBaseResource attribute), 42, 50
- methods (app.blueprints.roles.RoleResource attribute), 45, 51
- methods (app.blueprints.roles.RolesSearchResource attribute), 47, 51
- methods (app.blueprints.tasks.TaskResource attribute), 53, 56
- methods (app.blueprints.tasks.TaskStatusResource attribute), 55, 57
- methods (app.blueprints.users.ExportUsersExcelAndWordResource attribute), 58, 76
- methods (app.blueprints.users.ExportUsersExcelResource attribute), 61, 76
- methods (app.blueprints.users.ExportUsersWordResource attribute), 63, 77
- methods (app.blueprints.users.NewUserResource attribute), 66, 77
- methods (app.blueprints.users.UserBaseResource attribute), 68, 78
- methods (app.blueprints.users.UserResource attribute), 71, 79
- methods (app.blueprints.users.UsersSearchResource attribute), 73, 79
- Middleware (class in app.middleware), 129, 130
- migrate\_actions() (in module database.migrations), 282, 284
- Migration (class in database.migrations), 275, 282
- mime\_type (app.models.document.Document attribute), 139, 147
- module
  - app, 8
  - app.blueprints, 8
  - app.blueprints.auth, 8
  - app.blueprints.base, 19
  - app.blueprints.documents, 24
  - app.blueprints.roles, 39
  - app.blueprints.tasks, 52
  - app.blueprints.users, 57
  - app.celery, 80
  - app.celery.excel, 80
  - app.celery.excel.tasks, 80
  - app.celery.tasks, 81
  - app.celery.word, 81



app.celery.word.tasks, 81  
 app.extensions, 129  
 app.middleware, 129  
 app.models, 130  
 app.models.base, 130  
 app.models.document, 138  
 app.models.role, 148  
 app.models.user, 158  
 app.utils, 174  
 app.utils.decorators, 174  
 app.utils.file\_storage, 175  
 app.utils.libreoffice, 176  
 app.utils.marshmallow\_schema, 177  
 app.utils.swagger\_models, 257  
 app.utils.swagger\_models.auth, 257  
 app.utils.swagger\_models.document, 257  
 app.utils.swagger\_models.role, 257  
 app.utils.swagger\_models.user, 258  
 config, 302  
 database, 261  
 database.factories, 261  
 database.migrations, 266  
 database.migrations.aaa\_add\_genre\_column\_on\_user\_table, 266  
 database.migrations.aab\_add\_created\_by\_column\_on\_user\_table, 267  
 database.migrations.aac\_create\_documents\_table, 268  
 database.migrations.aad\_create\_user\_roles\_table, 269  
 database.migrations.aaf\_remove\_role\_slug\_column, 272  
 database.seeds, 284  
 database.seeds.document\_seeder, 284  
 database.seeds.role\_seeder, 285  
 database.seeds.user\_seeder, 286  
 tests, 288  
 tests.blueprints, 290  
 tests.blueprints.test\_auth, 290  
 tests.blueprints.test\_base, 291  
 tests.blueprints.test\_documents, 292  
 tests.blueprints.test\_roles, 294  
 tests.blueprints.test\_users, 295  
 tests.celery, 297  
 tests.celery.test\_excel, 297  
 tests.celery.test\_tasks, 298  
 tests.celery.test\_word, 298  
 tests.conftest, 299  
 tests.test\_config, 300  
 tests.test\_db, 301  
 tests.test\_mail, 301  
 mro() (*config.Meta* method), 315, 330  
 MyCelery (*class in app.celery*), 95, 119

## N

name (*app.celery.ContextTask* attribute), 84, 115  
 name (*app.models.document.Document* attribute), 140, 147  
 name (*app.models.role.Role* attribute), 149, 158  
 name (*app.models.user.User* attribute), 161, 172  
 name (*app.utils.marshmallow\_schema.Timestamp* attribute), 205, 241  
 name (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 272  
 name (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* attribute), 275  
 name (*database.migrations.Migration* attribute), 276, 283  
 name (*database.seeds.document\_seeder.DocumentSeeder* attribute), 285  
 name (*database.seeds.role\_seeder.RoleSeeder* attribute), 286  
 name (*database.seeds.user\_seeder.UserSeeder* attribute), 287  
 NewDocumentResource (*class in app.blueprints.documents*), 30, 37  
 NewRoleResource (*class in app.blueprints.roles*), 39  
 NewUserResource (*class in app.blueprints.users*), 65  
 noop() (*app.models.base.Base* class method), 135, 137  
 noop() (*app.models.document.Document* class method), 145, 147  
 noop() (*app.models.role.Role* class method), 155, 158  
 noop() (*app.models.user.User* class method), 168, 172  
 noop() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* class method), 272  
 noop() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* class method), 275  
 noop() (*database.migrations.Migration* class method), 281, 283  
 now() (*app.celery.MyCelery* method), 108, 125

## O

oid (*app.celery.MyCelery* attribute), 100, 126  
 on\_after\_configure (*app.celery.MyCelery* attribute), 100, 126  
 on\_after\_finalize (*app.celery.MyCelery* attribute), 100, 126  
 on\_after\_fork (*app.celery.MyCelery* attribute), 100, 126  
 on\_bind\_field() (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 251  
 on\_bind\_field() (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 257  
 on\_bind\_field() (*app.utils.marshmallow\_schema.DocumentSchema* method), 182, 217

- on\_bind\_field() (*app.utils.marshmallow\_schema.ExportWordInputSchema.marshmallow\_schema.RoleSchema.Meta*  
*method*), 188, 223
- on\_bind\_field() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema.marshmallow\_schema.UserSchema.Meta*  
*method*), 194, 228
- on\_bind\_field() (*app.utils.marshmallow\_schema.RoleSchema*  
*method*), 199, 233
- on\_bind\_field() (*app.utils.marshmallow\_schema.SearchSchema*  
*method*), 204, 239
- on\_bind\_field() (*app.utils.marshmallow\_schema.UserSchema*  
*method*), 212, 245
- on\_bound() (*app.celery.ContextTask* class *method*),  
 90, 115
- on\_configure (*app.celery.MyCelery* attribute), 100,  
 126
- on\_failure() (*app.celery.ContextTask* *method*), 90,  
 115
- on\_init() (*app.celery.MyCelery* *method*), 108, 126
- on\_retry() (*app.celery.ContextTask* *method*), 91, 115
- on\_success() (*app.celery.ContextTask* *method*), 91,  
 116
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.\_SearchOrderSchema*  
*attribute*), 247
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.\_SearchValueSchema*  
*attribute*), 252
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.DocumentSchema*  
*attribute*), 213
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.ExportWordInputSchema*  
*attribute*), 219
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema*  
*attribute*), 224
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.RoleSchema*  
*attribute*), 229
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.SearchSchema*  
*attribute*), 234
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.UserSchema*  
*attribute*), 241
- opts (*app.utils.marshmallow\_schema.\_SearchOrderSchema*  
*attribute*), 251
- opts (*app.utils.marshmallow\_schema.\_SearchValueSchema*  
*attribute*), 257
- opts (*app.utils.marshmallow\_schema.DocumentSchema*  
*attribute*), 178, 217
- opts (*app.utils.marshmallow\_schema.ExportWordInputSchema*  
*attribute*), 184, 223
- opts (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema*  
*attribute*), 190, 228
- opts (*app.utils.marshmallow\_schema.RoleSchema* *at-*  
*tribute*), 195, 233
- opts (*app.utils.marshmallow\_schema.SearchSchema* *at-*  
*tribute*), 200, 239
- opts (*app.utils.marshmallow\_schema.UserSchema* *at-*  
*tribute*), 208, 245
- ordered (*app.utils.marshmallow\_schema.DocumentSchema.Meta*  
*attribute*), 213
- on\_bind\_field() (*app.utils.marshmallow\_schema.ExportWordInputSchema.marshmallow\_schema.RoleSchema.Meta*  
*attribute*), 229
- on\_bind\_field() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema.marshmallow\_schema.UserSchema.Meta*  
*attribute*), 241
- on\_bind\_field() (*app.utils.marshmallow\_schema.RoleSchema*  
*attribute*), 233
- on\_bind\_field() (*app.utils.marshmallow\_schema.SearchSchema*  
*attribute*), 239
- on\_bind\_field() (*app.utils.marshmallow\_schema.UserSchema*  
*attribute*), 245
- on\_bound() (*app.celery.ContextTask* class *method*),  
 115
- on\_configure (*app.celery.MyCelery* attribute), 126
- on\_failure() (*app.celery.ContextTask* *method*), 115
- on\_init() (*app.celery.MyCelery* *method*), 126
- on\_retry() (*app.celery.ContextTask* *method*), 115
- on\_success() (*app.celery.ContextTask* *method*), 116
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.\_SearchOrderSchema*  
*attribute*), 247
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.\_SearchValueSchema*  
*attribute*), 252
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.DocumentSchema*  
*attribute*), 213
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.ExportWordInputSchema*  
*attribute*), 219
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema*  
*attribute*), 224
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.RoleSchema*  
*attribute*), 229
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.SearchSchema*  
*attribute*), 234
- OPTIONS\_CLASS (*app.utils.marshmallow\_schema.UserSchema*  
*attribute*), 241
- opts (*app.utils.marshmallow\_schema.\_SearchOrderSchema*  
*attribute*), 251
- opts (*app.utils.marshmallow\_schema.\_SearchValueSchema*  
*attribute*), 257
- opts (*app.utils.marshmallow\_schema.DocumentSchema*  
*attribute*), 217
- opts (*app.utils.marshmallow\_schema.ExportWordInputSchema*  
*attribute*), 223
- opts (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema*  
*attribute*), 228
- opts (*app.utils.marshmallow\_schema.RoleSchema* *at-*  
*tribute*), 233
- opts (*app.utils.marshmallow\_schema.SearchSchema* *at-*  
*tribute*), 239
- opts (*app.utils.marshmallow\_schema.UserSchema* *at-*  
*tribute*), 245
- ordered (*app.utils.marshmallow\_schema.DocumentSchema.Meta*  
*attribute*), 213
- password (*app.models.user.User* attribute), 161, 172
- password (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser*  
*attribute*), 272
- Pickler (*app.celery.MyCelery* attribute), 120
- pool() (*app.celery.MyCelery* property), 100, 126
- pop\_request() (*app.celery.ContextTask* *method*), 91,  
 116
- pos\_to\_char() (in module *app.utils*), 258, 260
- post() (*app.blueprints.auth.AuthUserLoginResource*  
*method*), 10, 17
- post() (*app.blueprints.auth.AuthUserLogoutResource*  
*method*), 12, 17
- post() (*app.blueprints.auth.RequestResetPasswordResource*  
*method*), 14, 18
- post() (*app.blueprints.auth.ResetPasswordResource*  
*method*), 16, 18
- post() (*app.blueprints.documents.NewDocumentResource*  
*method*), 33, 38
- post() (*app.blueprints.documents.SearchDocumentResource*  
*method*), 36, 39
- post() (*app.blueprints.roles.NewRoleResource*  
*method*), 41, 50
- post() (*app.blueprints.roles.RolesSearchResource*  
*method*), 49, 51
- post() (*app.blueprints.users.ExportUsersExcelAndWordResource*  
*method*), 60, 76
- post() (*app.blueprints.users.ExportUsersExcelResource*  
*method*), 62, 76
- post() (*app.blueprints.users.ExportUsersWordResource*  
*method*), 65, 77
- post() (*app.blueprints.users.NewUserResource*  
*method*), 67, 77
- post() (*app.blueprints.users.UsersSearchResource*  
*method*), 75, 79
- provide\_config() (*app.celery.MyCelery* *method*),  
 108, 126
- priority (*app.celery.ContextTask* attribute), 84, 116
- ProdConfig (class in *config*), 315, 330
- producer\_or\_acquire() (*app.celery.MyCelery*  
*method*), 108, 126
- producer\_pool() (*app.celery.MyCelery* property),  
 100, 126
- provide\_automatic\_options  
 (*app.blueprints.auth.AuthUserLoginResource*

*attribute*), 9, 17  
 provide\_automatic\_options  
     (*app.blueprints.auth.AuthUserLogoutResource attribute*), 11, 17  
 provide\_automatic\_options  
     (*app.blueprints.auth.RequestResetPasswordResource attribute*), 13, 18  
 provide\_automatic\_options  
     (*app.blueprints.auth.ResetPasswordResource attribute*), 15, 18  
 provide\_automatic\_options  
     (*app.blueprints.base.BaseResource attribute*), 19, 23  
 provide\_automatic\_options  
     (*app.blueprints.base.WelcomeResource attribute*), 21, 23  
 provide\_automatic\_options  
     (*app.blueprints.documents.DocumentBaseResource attribute*), 25, 36  
 provide\_automatic\_options  
     (*app.blueprints.documents.DocumentResource attribute*), 28, 37  
 provide\_automatic\_options  
     (*app.blueprints.documents.NewDocumentResource attribute*), 31, 38  
 provide\_automatic\_options  
     (*app.blueprints.documents.SearchDocumentResource attribute*), 34, 39  
 provide\_automatic\_options  
     (*app.blueprints.roles.NewRoleResource attribute*), 40, 50  
 provide\_automatic\_options  
     (*app.blueprints.roles.RoleBaseResource attribute*), 42, 50  
 provide\_automatic\_options  
     (*app.blueprints.roles.RoleResource attribute*), 45, 51  
 provide\_automatic\_options  
     (*app.blueprints.roles.RolesSearchResource attribute*), 48, 51  
 provide\_automatic\_options  
     (*app.blueprints.tasks.TaskResource attribute*), 53, 56  
 provide\_automatic\_options  
     (*app.blueprints.tasks.TaskStatusResource attribute*), 55, 57  
 provide\_automatic\_options  
     (*app.blueprints.users.ExportUsersExcelAndWordResource attribute*), 58, 76  
 provide\_automatic\_options  
     (*app.blueprints.users.ExportUsersExcelResource attribute*), 61, 76  
 provide\_automatic\_options  
     (*app.blueprints.users.ExportUsersWordResource attribute*), 63, 77  
 provide\_automatic\_options  
     (*app.blueprints.users.NewUserResource attribute*), 66, 77  
 provide\_automatic\_options  
     (*app.blueprints.users.UserBaseResource attribute*), 68, 78  
 provide\_automatic\_options  
     (*app.blueprints.users.UserResource attribute*), 71, 79  
 provide\_automatic\_options  
     (*app.blueprints.users.UsersSearchResource attribute*), 74, 79  
 push\_request() (*app.celery.ContextTask method*), 91, 116  
 put() (*app.blueprints.documents.DocumentResource method*), 30, 37  
 put() (*app.blueprints.roles.RoleResource method*), 47, 51  
 put() (*app.blueprints.users.UserResource method*), 73, 79

## R

rate\_limit (*app.celery.ContextTask attribute*), 84, 116  
 raw() (*app.models.base.Base class method*), 135, 137  
 raw() (*app.models.document.Document class method*), 145, 147  
 raw() (*app.models.role.Role class method*), 155, 158  
 raw() (*app.models.user.User class method*), 168, 172  
 raw() (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser class method*), 272  
 raw() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole class method*), 275  
 raw() (*database.migrations.Migration class method*), 281, 283  
 register\_task() (*app.celery.MyCelery method*), 109, 126  
 registry\_cls (*app.celery.MyCelery attribute*), 100, 126  
 reject\_on\_worker\_lost (*app.celery.ContextTask attribute*), 84, 116  
 remove\_permissions() (*app.models.role.Role method*), 155, 158  
 remove\_permissions() (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole method*), 275  
 RoleSlugColumn (class in *database.migrations.aaf\_remove\_role\_slug\_column*), 272, 273  
 rename() (*app.utils.file\_storage.FileStorage static method*), 176  
 replace() (*app.celery.ContextTask method*), 91, 116

- replace() (*app.models.base.Base* class method), 135, 137
- replace() (*app.models.document.Document* class method), 145, 147
- replace() (*app.models.role.Role* class method), 155, 158
- replace() (*app.models.user.User* class method), 168, 172
- replace() (*database.migrations.aad\_create\_user\_roles\_table* class method), 272
- replace() (*database.migrations.aaf\_remove\_role\_slug\_column* class method), 275
- replace() (*database.migrations.Migration* class method), 281, 283
- replace\_many() (*app.models.base.Base* class method), 135, 137
- replace\_many() (*app.models.document.Document* class method), 145, 147
- replace\_many() (*app.models.role.Role* class method), 155, 158
- replace\_many() (*app.models.user.User* class method), 168, 172
- replace\_many() (*database.migrations.aad\_create\_user\_roles\_table* class method), 272
- replace\_many() (*database.migrations.aaf\_remove\_role\_slug\_column* class method), 275
- replace\_many() (*database.migrations.Migration* class method), 281, 284
- representations(*app.blueprints.auth.AuthUserLoginResource* attribute), 9, 17
- representations(*app.blueprints.auth.AuthUserLogoutResource* attribute), 11, 17
- representations(*app.blueprints.auth.RequestResetPasswordResource* attribute), 13, 18
- representations(*app.blueprints.auth.ResetPasswordResource* attribute), 15, 18
- representations(*app.blueprints.base.BaseResource* attribute), 20, 23
- representations(*app.blueprints.base>WelcomeResource* attribute), 22, 23
- representations(*app.blueprints.documents.DocumentBaseResource* attribute), 25, 36
- representations(*app.blueprints.documents.DocumentResource* attribute), 28, 37
- representations(*app.blueprints.documents.NewDocumentResource* attribute), 31, 38
- representations(*app.blueprints.documents.SearchDocumentResource* attribute), 34, 39
- representations(*app.blueprints.roles.NewRoleResource* attribute), 40, 50
- representations(*app.blueprints.roles.RoleBaseResource* attribute), 42, 50
- representations(*app.blueprints.roles.RoleResource* attribute), 45, 51
- representations(*app.blueprints.roles.RolesSearchResource* attribute), 48, 51
- representations(*app.blueprints.tasks.TaskResource* attribute), 53, 56
- representations(*app.blueprints.tasks.TaskStatusResource* attribute), 55, 57
- representations(*app.blueprints.users.ExportUsersExcelAndWordResource* attribute), 58, 76
- representations(*app.blueprints.users.ExportUsersExcelResource* attribute), 61, 76
- representations(*app.blueprints.users.ExportUsersWordResource* attribute), 64, 77
- representations(*app.blueprints.users.NewUserResource* attribute), 66, 77
- representations(*app.blueprints.users.UserBaseResource* attribute), 68, 78
- representations(*app.blueprints.users.UserResource* attribute), 71, 79
- representations(*app.blueprints.users.UsersSearchResource* attribute), 74, 79
- Request (*app.celery.ContextTask* attribute), 83, 112
- request() (*app.celery.ContextTask* property), 84, 116
- request\_field\_name (*app.blueprints.documents.DocumentBaseResource* attribute), 28, 37
- request\_field\_name (*app.blueprints.documents.DocumentResource* attribute), 28, 37
- request\_field\_name (*app.blueprints.documents.NewDocumentResource* attribute), 31, 38
- request\_field\_name (*app.blueprints.documents.SearchDocumentResource* attribute), 34, 39
- request\_stack (*app.celery.ContextTask* attribute), 85, 116
- RequestResetPasswordResource (class in *app.blueprints.auth*), 13, 17
- reset\_password\_email() (in module *app.celery.tasks*), 81
- RESET\_PASSWORD\_EXPIRES (*config.Config* attribute), 305, 328
- RESET\_PASSWORD\_EXPIRES (*config.DevConfig* attribute), 311, 329
- RESET\_PASSWORD\_EXPIRES (*config.ProdConfig* attribute), 318, 331
- RESET\_PASSWORD\_EXPIRES (*config.TestConfig* attribute), 324, 332
- ResetPasswordResource (class in *app.blueprints.auth*), 15, 18
- RESTX\_MASK\_SWAGGER (*config.Config* attribute), 305, 328
- RESTX\_MASK\_SWAGGER (*config.DevConfig* attribute), 311, 329



- RESTX\_MASK\_SWAGGER (*config.ProdConfig* attribute), 318, 331  
 RESTX\_MASK\_SWAGGER (*config.TestConfig* attribute), 324, 332  
 result\_backend (*config.Config* attribute), 307, 328  
 result\_backend (*config.DevConfig* attribute), 313, 330  
 result\_backend (*config.ProdConfig* attribute), 320, 331  
 result\_backend (*config.TestConfig* attribute), 326, 332  
 result\_expires (*config.Config* attribute), 307, 328  
 result\_expires (*config.DevConfig* attribute), 313, 330  
 result\_expires (*config.ProdConfig* attribute), 320, 331  
 result\_expires (*config.TestConfig* attribute), 326, 332  
 result\_extended (*config.Config* attribute), 307, 328  
 result\_extended (*config.DevConfig* attribute), 313, 330  
 result\_extended (*config.ProdConfig* attribute), 320, 331  
 result\_extended (*config.TestConfig* attribute), 326, 333  
 result\_serializer (*config.Config* attribute), 307, 328  
 result\_serializer (*config.DevConfig* attribute), 313, 330  
 result\_serializer (*config.ProdConfig* attribute), 320, 331  
 result\_serializer (*config.TestConfig* attribute), 326, 333  
 resultrepr\_maxsize (*app.celery.ContextTask* attribute), 85, 116  
 ResultSet (*app.celery.MyCelery* attribute), 97, 120  
 retry () (*app.celery.ContextTask* method), 92, 116  
 Role (class in *app.models.role*), 148, 156  
 role (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 272  
 role\_id (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* attribute), 272  
 role\_serializer (*app.blueprints.roles.NewRoleResource* attribute), 40, 50  
 role\_serializer (*app.blueprints.roles.RoleBaseResource* attribute), 43, 50  
 role\_serializer (*app.blueprints.roles.RoleResource* attribute), 45, 51  
 role\_serializer (*app.blueprints.roles.RolesSearchResource* attribute), 48, 51  
 RoleBaseResource (class in *app.blueprints.roles*), 42, 50  
 RoleResource (class in *app.blueprints.roles*), 44, 50  
 roles (*app.models.role.Role* attribute), 149, 158  
 roles (*app.models.user.User* attribute), 161, 173  
 RoleSchema (class in *app.utils.marshmallow\_schema*), 194, 229  
 RoleSchema.Meta (class in *app.utils.marshmallow\_schema*), 229  
 RoleSeeder (class in *database.seeds.role\_seeder*), 285, 286  
 RolesSearchResource (class in *app.blueprints.roles*), 47, 51  
 rollback\_actions () (in module *database.migrations*), 282, 284  
 root () (*app.utils.marshmallow\_schema.Timestamp* property), 205, 241  
 ROOT\_DIRECTORY (*config.Config* attribute), 305, 328  
 ROOT\_DIRECTORY (*config.DevConfig* attribute), 311, 329  
 ROOT\_DIRECTORY (*config.ProdConfig* attribute), 318, 331  
 ROOT\_DIRECTORY (*config.TestConfig* attribute), 324, 332  
 run () (*app.celery.ContextTask* method), 93, 117  
 runner () (in module *tests.conftest*), 300
- ## S
- s () (*app.celery.ContextTask* method), 93, 117  
 save () (*app.models.base.Base* method), 136, 137  
 save () (*app.models.document.Document* method), 145, 147  
 save () (*app.models.role.Role* method), 155, 158  
 save () (*app.models.user.User* method), 168, 173  
 save () (*database.factories.Factory* method), 262, 264, 265  
 save () (*database.migrations.aad\_create\_user\_roles\_table.\_OldUser* method), 272  
 save () (*database.migrations.aaf\_remove\_role\_slug\_column.\_OldRole* method), 275  
 save () (*database.migrations.Migration* method), 281, 284  
 save\_bytes () (*app.utils.file\_storage.FileStorage* method), 176  
 SearchDocumentResource (class in *app.blueprints.documents*), 33, 38  
 SearchSchema (class in *app.utils.marshmallow\_schema*), 199, 234  
 SearchSchema.Meta (class in *app.utils.marshmallow\_schema*), 234  
 SECRET\_KEY (*config.Config* attribute), 305, 328  
 SECRET\_KEY (*config.DevConfig* attribute), 311, 329  
 SECRET\_KEY (*config.ProdConfig* attribute), 318, 331  
 SECRET\_KEY (*config.TestConfig* attribute), 324, 332  
 SECURITY\_PASSWORD\_HASH (*config.Config* attribute), 305, 328  
 SECURITY\_PASSWORD\_HASH (*config.DevConfig* attribute), 312, 329

SECURITY\_PASSWORD\_HASH (*config.ProdConfig* attribute), 318, 331  
 SECURITY\_PASSWORD\_HASH (*config.TestConfig* attribute), 325, 332  
 SECURITY\_PASSWORD\_LENGTH\_MIN (*config.Config* attribute), 306, 328  
 SECURITY\_PASSWORD\_LENGTH\_MIN (*config.DevConfig* attribute), 312, 329  
 SECURITY\_PASSWORD\_LENGTH\_MIN (*config.ProdConfig* attribute), 319, 331  
 SECURITY\_PASSWORD\_LENGTH\_MIN (*config.TestConfig* attribute), 325, 332  
 SECURITY\_PASSWORD\_SALT (*config.Config* attribute), 306, 328  
 SECURITY\_PASSWORD\_SALT (*config.DevConfig* attribute), 312, 329  
 SECURITY\_PASSWORD\_SALT (*config.ProdConfig* attribute), 319, 331  
 SECURITY\_PASSWORD\_SALT (*config.TestConfig* attribute), 325, 332  
 SECURITY\_TOKEN\_AUTHENTICATION\_HEADER (*config.Config* attribute), 306, 328  
 SECURITY\_TOKEN\_AUTHENTICATION\_HEADER (*config.DevConfig* attribute), 312, 329  
 SECURITY\_TOKEN\_AUTHENTICATION\_HEADER (*config.ProdConfig* attribute), 319, 331  
 SECURITY\_TOKEN\_AUTHENTICATION\_HEADER (*config.TestConfig* attribute), 325, 332  
 SECURITY\_TOKEN\_MAX\_AGE (*config.Config* attribute), 306, 328  
 SECURITY\_TOKEN\_MAX\_AGE (*config.DevConfig* attribute), 312, 329  
 SECURITY\_TOKEN\_MAX\_AGE (*config.ProdConfig* attribute), 319, 331  
 SECURITY\_TOKEN\_MAX\_AGE (*config.TestConfig* attribute), 325, 332  
 seed\_actions() (in module *database*), 288  
 select() (*app.models.base.Base* class method), 136, 138  
 select() (*app.models.document.Document* class method), 145, 147  
 select() (*app.models.role.Role* class method), 155, 158  
 select() (*app.models.user.User* class method), 168, 173  
 select() (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 272  
 select() (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 275  
 select() (*database.migrations.Migration* class method), 281, 284  
 select\_queues() (*app.celery.MyCelery* method), 109, 126  
 send\_email\_with\_attachments() (in module *app.celery.tasks*), 81  
 send\_event() (*app.celery.ContextTask* method), 93, 118  
 send\_events (*app.celery.ContextTask* attribute), 85, 118  
 send\_task() (*app.celery.MyCelery* method), 109, 126  
 serialize() (*app.utils.marshmallow\_schema.Timestamp* method), 207, 241  
 serializer (*app.celery.ContextTask* attribute), 85, 118  
 SERVER\_NAME (*config.Config* attribute), 306, 328  
 SERVER\_NAME (*config.DevConfig* attribute), 312, 329  
 SERVER\_NAME (*config.ProdConfig* attribute), 319, 331  
 SERVER\_NAME (*config.TestConfig* attribute), 325, 332  
 set\_by\_id() (*app.models.base.Base* class method), 136, 138  
 set\_by\_id() (*app.models.document.Document* class method), 145, 147  
 set\_by\_id() (*app.models.role.Role* class method), 155, 158  
 set\_by\_id() (*app.models.user.User* class method), 168, 173  
 set\_by\_id() (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 272  
 set\_by\_id() (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 275  
 set\_by\_id() (*database.migrations.Migration* class method), 281, 284  
 set\_class() (*app.utils.marshmallow\_schema.\_SearchOrderSchema* property), 251  
 set\_class() (*app.utils.marshmallow\_schema.\_SearchValueSchema* property), 257  
 set\_class() (*app.utils.marshmallow\_schema.DocumentSchema* property), 179, 217  
 set\_class() (*app.utils.marshmallow\_schema.ExportWordInputSchema* property), 184, 223  
 set\_class() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* property), 190, 228  
 set\_class() (*app.utils.marshmallow\_schema.RoleSchema* property), 195, 233  
 set\_class() (*app.utils.marshmallow\_schema.SearchSchema* property), 200, 239  
 set\_class() (*app.utils.marshmallow\_schema.UserSchema* property), 208, 245  
 set\_current() (*app.celery.MyCelery* method), 109, 127  
 set\_default() (*app.celery.MyCelery* method), 109, 127  
 setup\_security() (*app.celery.MyCelery* method), 110, 127  
 shadow\_name() (*app.celery.ContextTask* method), 93, 118  
 si() (*app.celery.ContextTask* method), 94, 118  
 signature() (*app.celery.ContextTask* method), 94, 118

118  
signature() (*app.celery.MyCelery* method), 110, 127  
signature\_from\_request() (*app.celery.ContextTask* method), 94, 118  
size (*app.models.document.Document* attribute), 140, 147  
slug (*database.migrations.aaf\_remove\_role\_slug\_column.table* attribute), 275  
soft\_time\_limit (*app.celery.ContextTask* attribute), 85, 118  
starmap() (*app.celery.ContextTask* method), 94, 119  
start\_strategy() (*app.celery.ContextTask* method), 94, 119  
steps (*app.celery.MyCelery* attribute), 101, 127  
STORAGE\_DIRECTORY (*config.Config* attribute), 306, 328  
STORAGE\_DIRECTORY (*config.DevConfig* attribute), 312, 329  
STORAGE\_DIRECTORY (*config.ProdConfig* attribute), 319, 331  
STORAGE\_DIRECTORY (*config.TestConfig* attribute), 325, 332  
store\_errors\_even\_if\_ignored (*app.celery.ContextTask* attribute), 85, 119  
Strategy (*app.celery.ContextTask* attribute), 83, 112  
strerror (*app.utils.FileEmptyError* attribute), 259  
STRING\_QUERY\_OPERATORS (in module *app.utils*), 259  
subclass\_with\_self() (*app.celery.MyCelery* method), 110, 127  
subtask() (*app.celery.ContextTask* method), 94, 119  
subtask\_from\_request() (*app.celery.ContextTask* method), 95, 119  
SWAGGER\_API\_URL (*config.Config* attribute), 306, 328  
SWAGGER\_API\_URL (*config.DevConfig* attribute), 312, 329  
SWAGGER\_API\_URL (*config.ProdConfig* attribute), 319, 331  
SWAGGER\_API\_URL (*config.TestConfig* attribute), 325, 332  
SWAGGER\_URL (*config.Config* attribute), 306, 328  
SWAGGER\_URL (*config.DevConfig* attribute), 312, 329  
SWAGGER\_URL (*config.ProdConfig* attribute), 319, 331  
SWAGGER\_URL (*config.TestConfig* attribute), 325, 332  
SYSTEM (*app.celery.MyCelery* attribute), 97, 120

## T

table\_exists() (*app.models.base.Base* class method), 136, 138  
table\_exists() (*app.models.document.Document* class method), 145, 147  
table\_exists() (*app.models.role.Role* class method), 156, 158  
table\_exists() (*app.models.user.User* class method), 169, 173  
table\_exists() (*database.migrations.aad\_create\_user\_roles\_table.table* class method), 272  
table\_exists() (*database.migrations.aaf\_remove\_role\_slug\_column.table* class method), 275  
table\_exists() (*database.migrations.ald\_remove\_role\_slug\_column.table* class method), 281, 284  
Task (*app.celery.MyCelery* attribute), 97, 120  
task() (*app.celery.MyCelery* method), 111, 127  
task\_cls (*app.celery.MyCelery* attribute), 101, 128  
task\_default\_rate\_limit (*config.Config* attribute), 307, 328  
task\_default\_rate\_limit (*config.DevConfig* attribute), 314, 330  
task\_default\_rate\_limit (*config.ProdConfig* attribute), 320, 331  
task\_default\_rate\_limit (*config.TestConfig* attribute), 327, 333  
task\_serializer (*config.Config* attribute), 308, 328  
task\_serializer (*config.DevConfig* attribute), 314, 330  
task\_serializer (*config.ProdConfig* attribute), 321, 331  
task\_serializer (*config.TestConfig* attribute), 327, 333  
task\_track\_started (*config.Config* attribute), 308, 328  
task\_track\_started (*config.DevConfig* attribute), 314, 330  
task\_track\_started (*config.ProdConfig* attribute), 321, 331  
task\_track\_started (*config.TestConfig* attribute), 327, 333  
TaskFailure, 112, 128  
TaskResource (class in *app.blueprints.tasks*), 52, 56  
tasks (*app.celery.MyCelery* attribute), 101, 128  
TaskStatusResource (class in *app.blueprints.tasks*), 54, 56  
test\_config() (in module *tests.test\_config*), 300  
test\_create\_user\_email\_task() (in module *tests.celery.test\_tasks*), 298  
test\_create\_word\_and\_excel\_documents() (in module *tests.celery.test\_tasks*), 298  
test\_delete\_document() (in module *tests.blueprints.test\_documents*), 292, 293  
test\_delete\_role\_endpoint() (in module *tests.blueprints.test\_roles*), 294, 295  
test\_delete\_user\_endpoint() (in module *tests.blueprints.test\_users*), 296  
test\_export\_excel\_endpoint() (in module *tests.blueprints.test\_users*), 296  
test\_export\_excel\_task() (in module *tests.celery.test\_excel*), 298

`test_export_word_endpoint()` (in module `tests.blueprints.test_users`), 296

`test_export_word_task()` (in module `tests.celery.test_word`), 299

`test_get_close_db()` (in module `tests.test_db`), 301

`test_get_document_data()` (in module `tests.blueprints.test_documents`), 292, 293

`test_get_document_file()` (in module `tests.blueprints.test_documents`), 293

`test_get_role_endpoint()` (in module `tests.blueprints.test_roles`), 294, 295

`test_get_user_endpoint()` (in module `tests.blueprints.test_users`), 296, 297

`test_mail_record_messages()` (in module `tests.test_mail`), 301, 302

`test_request_reset_password()` (in module `tests.blueprints.test_auth`), 290, 291

`test_reset_password()` (in module `tests.blueprints.test_auth`), 291

`test_reset_password_email_task()` (in module `tests.celery.test_tasks`), 298

`test_save_document()` (in module `tests.blueprints.test_documents`), 293

`test_save_role_endpoint()` (in module `tests.blueprints.test_roles`), 294, 295

`test_save_user_endpoint()` (in module `tests.blueprints.test_users`), 296, 297

`test_search_document()` (in module `tests.blueprints.test_documents`), 293

`test_search_roles_endpoint()` (in module `tests.blueprints.test_roles`), 294, 295

`test_search_users_endpoint()` (in module `tests.blueprints.test_users`), 296, 297

`test_update_document()` (in module `tests.blueprints.test_documents`), 293

`test_update_role_endpoint()` (in module `tests.blueprints.test_roles`), 295

`test_update_user_endpoint()` (in module `tests.blueprints.test_users`), 296, 297

`TEST_USER_EMAIL` (*config.Config* attribute), 306, 328

`TEST_USER_EMAIL` (*config.DevConfig* attribute), 313, 329

`TEST_USER_EMAIL` (*config.ProdConfig* attribute), 319, 331

`TEST_USER_EMAIL` (*config.TestConfig* attribute), 326, 332

`test_user_login()` (in module `tests.blueprints.test_auth`), 291

`test_user_logout()` (in module `tests.blueprints.test_auth`), 291

`TEST_USER_PASSWORD` (*config.Config* attribute), 307, 328

`TEST_USER_PASSWORD` (*config.DevConfig* attribute), 313, 329

`TEST_USER_PASSWORD` (*config.ProdConfig* attribute), 320, 331

`TEST_USER_PASSWORD` (*config.TestConfig* attribute), 326, 332

`test_validate_reset_password()` (in module `tests.blueprints.test_auth`), 291

`test_welcome_api()` (in module `tests.blueprints.test_base`), 292

`TestConfig` (class in *config*), 321, 331

`TESTING` (*config.Config* attribute), 306, 328

`TESTING` (*config.DevConfig* attribute), 312, 329

`TESTING` (*config.ProdConfig* attribute), 319, 331

`TESTING` (*config.TestConfig* attribute), 325, 332

`tests`

- module, 288

`tests.blueprints`

- module, 290

`tests.blueprints.test_auth`

- module, 290

`tests.blueprints.test_base`

- module, 291

`tests.blueprints.test_documents`

- module, 292

`tests.blueprints.test_roles`

- module, 294

`tests.blueprints.test_users`

- module, 295

`tests.celery`

- module, 297

`tests.celery.test_excel`

- module, 297

`tests.celery.test_tasks`

- module, 298

`tests.celery.test_word`

- module, 298

`tests.conftest`

- module, 299

`tests.test_config`

- module, 300

`tests.test_db`

- module, 301

`tests.test_mail`

- module, 301

`tf_send_security_token()`

- (*app.models.user.User* method), 169, 173

`throws` (*app.celery.ContextTask* attribute), 85, 119

`time_limit` (*app.celery.ContextTask* attribute), 85, 119

`Timestamp` (class in *app.utils.marshmallow\_schema*), 204, 239

`timezone` (*app.celery.MyCelery* attribute), 101, 128

`timezone` (*config.Config* attribute), 308, 328

`timezone` (*config.DevConfig* attribute), 314, 330



timezone (*config.ProdConfig* attribute), 321, 331  
 timezone (*config.TestConfig* attribute), 327, 333  
 to\_readable() (in module *app.utils*), 259, 260  
 token\_required() (in module *app.utils.decorators*), 175  
 track\_started (*app.celery.ContextTask* attribute), 85, 119  
 trail (*app.celery.ContextTask* attribute), 86, 119  
 truncate\_table() (*app.models.base.Base* class method), 136, 138  
 truncate\_table() (*app.models.document.Document* class method), 145, 147  
 truncate\_table() (*app.models.role.Role* class method), 156, 158  
 truncate\_table() (*app.models.user.User* class method), 169, 173  
 truncate\_table() (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 272  
 truncate\_table() (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 275  
 truncate\_table() (*database.migrations.Migration* class method), 281, 284  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.SearchQuerySchema* attribute), 247  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.SearchVideoSchema* attribute), 253  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.DocumentSchema* attribute), 178, 213  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.ExportWordInputSchema* attribute), 184, 219  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.GetDocumentDataAttributeSchema* attribute), 189, 224  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.RoleSchema* attribute), 195, 229  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.SearchSchema* attribute), 200, 234  
 TYPE\_MAPPING (*app.utils.marshmallow\_schema.UserSchema* attribute), 208, 241  
 typing (*app.celery.ContextTask* attribute), 86, 119

## U

unwrap() (*app.models.base.Base* method), 136, 138  
 unwrap() (*app.models.document.Document* method), 146, 147  
 unwrap() (*app.models.role.Role* method), 156, 158  
 unwrap() (*app.models.user.User* method), 169, 173  
 unwrap() (*database.migrations.aad\_create\_user\_roles\_table.OldUser* method), 272  
 unwrap() (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* method), 275  
 unwrap() (*database.migrations.Migration* method), 282, 284  
 up() (*database.migrations.aaa\_add\_genre\_column\_on\_user\_table.AddGenreColumnOnUserTable* method), 267  
 up() (*database.migrations.aab\_add\_created\_by\_column\_on\_user\_table.AabAddCreatedByColumnOnUserTable* method), 268  
 up() (*database.migrations.aac\_create\_documents\_table.CreateDocumentsTable* method), 269  
 up() (*database.migrations.aad\_create\_user\_roles\_table.CreateUserRolesTable* method), 270  
 up() (*database.migrations.aaf\_remove\_role\_slug\_column.RemoveRoleSlugColumn* method), 273  
 update() (*app.models.base.Base* class method), 136, 138  
 update() (*app.models.document.Document* class method), 146, 147  
 update() (*app.models.role.Role* class method), 156, 158  
 update() (*app.models.user.User* class method), 169, 173  
 update() (*database.migrations.aad\_create\_user\_roles\_table.OldUser* class method), 272  
 update() (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* class method), 275  
 update() (*database.migrations.Migration* class method), 282, 284  
 update() (*app.celery.ContextTask* method), 95, 119  
 updated\_at (*app.models.document.Document* attribute), 140, 147  
 updated\_at (*app.models.role.Role* attribute), 149, 158  
 updated\_at (*app.models.user.User* attribute), 161, 172  
 updated\_at (*database.migrations.aad\_create\_user\_roles\_table.OldUser* attribute), 272  
 updated\_at (*database.migrations.aaf\_remove\_role\_slug\_column.OldRole* attribute), 275  
 url() (*app.models.document.Document* property), 140, 148  
 us\_send\_security\_token() (*app.models.user.User* method), 169, 173  
 User (class in *app.models.user*), 159, 170  
 user\_options (*app.celery.MyCelery* attribute), 101, 128  
 user\_serializer (*app.blueprints.users.ExportUsersExcelAndWordResource* attribute), 58, 76  
 user\_serializer (*app.blueprints.users.ExportUsersExcelResource* attribute), 61, 76  
 user\_serializer (*app.blueprints.users.ExportUsersWordResource* attribute), 64, 77  
 user\_serializer (*app.blueprints.users.NewUserResource* attribute), 66, 77  
 user\_serializer (*app.blueprints.users.UserBaseResource* attribute), 69, 78  
 user\_serializer (*app.blueprints.users.UserResource* attribute), 71, 79  
 user\_serializer (*app.blueprints.users.UsersSearchResource* attribute), 74, 79

UserBaseResource (class in *app.blueprints.users*), 68, 78  
 UserResource (class in *app.blueprints.users*), 70, 78  
 userrolethrough\_set (*app.models.role.Role* attribute), 149, 158  
 userrolethrough\_set (*app.models.user.User* attribute), 161, 173  
 users (*app.models.role.Role* attribute), 150, 158  
 UserSchema (class in *app.utils.marshmallow\_schema*), 207, 241  
 UserSchema.Meta (class in *app.utils.marshmallow\_schema*), 241  
 UserSeeder (class in *database.seeds.user\_seeder*), 286, 287  
 UsersSearchResource (class in *app.blueprints.users*), 73, 79  
 uses\_utc\_timezone() (*app.celery.MyCelery* method), 111, 128

## V

valid\_request\_file() (*app.utils.marshmallow\_schema.DocumentSchema* static method), 183, 217  
 valid\_request\_role() (*app.utils.marshmallow\_schema.RoleSchema* static method), 199, 233  
 valid\_request\_user() (*app.utils.marshmallow\_schema.UserSchema* static method), 212, 245  
 validate() (*app.utils.marshmallow\_schema.\_SearchOrderSchema* method), 251  
 validate() (*app.utils.marshmallow\_schema.\_SearchValueSchema* method), 257  
 validate() (*app.utils.marshmallow\_schema.DocumentSchema* method), 183, 217  
 validate() (*app.utils.marshmallow\_schema.ExportWordInputSchema* method), 188, 223  
 validate() (*app.utils.marshmallow\_schema.GetDocumentDataInputSchema* method), 194, 228  
 validate() (*app.utils.marshmallow\_schema.RoleSchema* method), 199, 233  
 validate() (*app.utils.marshmallow\_schema.SearchSchema* method), 204, 239  
 validate() (*app.utils.marshmallow\_schema.UserSchema* method), 212, 245  
 validate\_credentials() (*app.utils.marshmallow\_schema.UserSchema* method), 212, 246  
 validate\_email() (*app.utils.marshmallow\_schema.UserSchema* static method), 212, 246  
 validate\_model() (*app.models.base.Base* class method), 136, 138  
 validate\_model() (*app.models.document.Document* class method), 146, 148  
 validate\_model() (*app.models.role.Role* class method), 156, 158  
 validate\_model() (*app.models.user.User* class method), 170, 173  
 validate\_model() (*database.migrations.aad\_create\_user\_roles\_table* class method), 272  
 validate\_model() (*database.migrations.aaf\_remove\_role\_slug\_column* class method), 275  
 validate\_model() (*database.migrations.Migration* class method), 282, 284  
 validate\_password() (*app.utils.marshmallow\_schema.UserSchema* static method), 213, 246  
 validate\_payload() (*app.blueprints.auth.AuthUserLoginResource* method), 10, 17  
 validate\_payload() (*app.blueprints.auth.AuthUserLogoutResource* method), 12, 17  
 validate\_payload() (*app.blueprints.auth.RequestResetPasswordResource* method), 14, 18  
 validate\_payload() (*app.blueprints.auth.ResetPasswordResource* method), 17, 18  
 validate\_payload() (*app.blueprints.base.BaseResource* method), 21, 23  
 validate\_payload() (*app.blueprints.base.WelcomeResource* method), 23  
 validate\_payload() (*app.blueprints.documents.DocumentBaseResource* method), 27, 36  
 validate\_payload() (*app.blueprints.documents.DocumentResource* method), 30, 37  
 validate\_payload() (*app.blueprints.documents.NewDocumentResource* method), 33, 38  
 validate\_payload() (*app.blueprints.documents.SearchDocumentResource* method), 36, 39  
 validate\_payload() (*app.blueprints.roles.NewRoleResource* method), 41, 50  
 validate\_payload() (*app.blueprints.roles.RoleBaseResource* method), 44, 50  
 validate\_payload() (*app.blueprints.roles.RoleResource* method), 47, 51  
 validate\_payload() (*app.blueprints.roles.RolesSearchResource* method), 51

*method*), 49, 51  
 validate\_payload() (app.blueprints.tasks.TaskResource *method*), 54, 56  
 validate\_payload() (app.blueprints.tasks.TaskStatusResource *method*), 56, 57  
 validate\_payload() (app.blueprints.users.ExportUsersExcelAndWordResource *method*), 60, 76  
 validate\_payload() (app.blueprints.users.ExportUsersExcelResource *method*), 63, 76  
 validate\_payload() (app.blueprints.users.ExportUsersWordResource *method*), 65, 77  
 validate\_payload() (app.blueprints.users.NewUserResource *method*), 67, 77  
 validate\_payload() (app.blueprints.users.UserBaseResource *method*), 70, 78  
 validate\_payload() (app.blueprints.users.UserResource *method*), 73, 79  
 validate\_payload() (app.blueprints.users.UsersSearchResource *method*), 75, 79  
 verify\_and\_update\_password() (app.models.user.User *method*), 170, 173  
 verify\_auth\_token() (app.models.user.User *method*), 170, 173  
 verify\_reset\_token() (app.models.user.User *static method*), 170, 174  
 worker\_log\_format (config.DevConfig *attribute*), 314, 330  
 worker\_log\_format (config.ProdConfig *attribute*), 321, 331  
 worker\_log\_format (config.TestConfig *attribute*), 327, 333  
 worker\_task\_log\_format (config.Config *attribute*), 308, 328  
 worker\_task\_log\_format (config.DevConfig *attribute*), 314, 330  
 worker\_task\_log\_format (config.ProdConfig *attribute*), 321, 331  
 worker\_task\_log\_format (config.TestConfig *attribute*), 327, 333

## W

WelcomeResource (class in app.blueprints.base), 21, 23  
 with\_traceback() (app.celery.ContextTask.MaxRetriesExceededError *method*), 112  
 with\_traceback() (app.celery.ContextTask.OperationalError *method*), 112  
 with\_traceback() (app.celery.TaskFailure *method*), 128  
 with\_traceback() (app.utils.FileEmptyError *method*), 259  
 with\_traceback() (app.utils.libreoffice.LibreOfficeError *method*), 177  
 WorkController (app.celery.MyCelery *attribute*), 97, 120  
 Worker (app.celery.MyCelery *attribute*), 97, 120  
 worker\_log\_format (config.Config *attribute*), 308, 328