
Flask Template Documentation

Release 1.0

Eric Daoud Attoyan

Feb 15, 2021

Contents:

1	Install	3
2	Run	5
3	Docs	7
4	Clone the template to another directory	9
4.1	application package	9
5	Indices and tables	11
	Python Module Index	13
	Index	15

This is a template for a basic Flask web application that responds an HTML page on `localhost:8080`.

This template comes with support for:

- [SQLite database](#)
- [Sphinx documentation](#)
- [Twitter Bootstrap](#)
- [Font Awesome](#)
- [Docker and docker-compose](#)

Don't hesitate to contribute!

CHAPTER 1

Install

Create a virtual environment and install the requirements.

```
make venv/bin/activate
```

Init the database by creating all tables.

```
export DATABASE_URL=sqlite:///../db.sqlite3
export FLASK_APP="application.app"
flask init-db
```


CHAPTER 2

Run

Run the application debug mode.

```
FLASK_DEBUG=True flask run
```

If you wish to use Docker for deploying the app, run the following:

```
docker-compose up -d
```


CHAPTER 3

Docs

Automatically create and build the code documentation using Sphinx. You can use [Read the Docs](#) to build and host the documentation, like I did [here](#).

```
make docs
```

Clone the template to another directory

```
./install.sh /path/to/your/dir
```

4.1 application package

4.1.1 Subpackages

`application.blueprints` package

Submodules

`application.blueprints.home` module

`application.blueprints.home.index()`

4.1.2 Submodules

4.1.3 `application.admin` module

`application.admin.init_db()`

4.1.4 `application.app` module

`application.app.create_app(config=None)`

Flask app factory that creates and configure the app.

Parameters `test_config` (*str*) – python configuration filepath

Returns: Flask application

4.1.5 application.cli module

4.1.6 application.config module

4.1.7 application.errors module

`application.errors.page_not_found(e)`

Renders template for 404 error

Parameters `e` – Error

Returns: 404 error template

4.1.8 application.model module

```
class application.model.User(**kwargs)
    Bases: sqlalchemy.ext.declarative.api.Model

    created_at
    email
    username
    uuid
```

4.1.9 application.schema module

```
class application.schema.UserSchema(*args, **kwargs)
    Bases: marshmallow_sqlalchemy.schema.ModelSchema

    class Meta
        Bases: object

        fields = ('username', 'email')

        model
            alias of application.model.User

        sqla_session = <sqlalchemy.orm.scoping.scoped_session object>

        opts = <marshmallow_sqlalchemy.schema.ModelSchemaOpts object>
```

4.1.10 application.wsgi module

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

a

- `application`, 9
- `application.admin`, 9
- `application.app`, 9
- `application.blueprints`, 9
- `application.blueprints.home`, 9
- `application.cli`, 10
- `application.config`, 10
- `application.errors`, 10
- `application.model`, 10
- `application.schema`, 10
- `application.wsgi`, 10

A

`application` (*module*), 9
`application.admin` (*module*), 9
`application.app` (*module*), 9
`application.blueprints` (*module*), 9
`application.blueprints.home` (*module*), 9
`application.cli` (*module*), 10
`application.config` (*module*), 10
`application.errors` (*module*), 10
`application.model` (*module*), 10
`application.schema` (*module*), 10
`application.wsgi` (*module*), 10

C

`create_app()` (*in module application.app*), 9
`created_at` (*application.model.User attribute*), 10

E

`email` (*application.model.User attribute*), 10

F

`fields` (*application.schema.UserSchema.Meta attribute*), 10

I

`index()` (*in module application.blueprints.home*), 9
`init_db()` (*in module application.admin*), 9

M

`model` (*application.schema.UserSchema.Meta attribute*), 10

O

`opts` (*application.schema.UserSchema attribute*), 10

P

`page_not_found()` (*in module application.errors*), 10

S

`sqla_session` (*application.schema.UserSchema.Meta attribute*), 10

U

`User` (*class in application.model*), 10
`username` (*application.model.User attribute*), 10
`UserSchema` (*class in application.schema*), 10
`UserSchema.Meta` (*class in application.schema*), 10
`uuid` (*application.model.User attribute*), 10