Frink Documentation

Release 0.0.1

Hactar

Oct 28, 2017

Contents

CHAPTER 1

About

Frink is a super basic ORM-like thing for using RethinkDB in Flask, built on top of Schematics. Flask-Frink adds Flask and *Flask-Security* compatibility.

Warning: Frink is currently pre-alpha and extremely likely to change. However, it is stable enough in the Flask-Security datastore that you could probably use it to enable Flask-Security to work with RethinkDB.

Install

```
pip install flask_frink
```

If you are contributing to the development of Frink, install it as an editable package from your own clone of the git repo. That way you can commit and push your edits.

```
git clone git@github.com:hactar-is/flask-frink.git lib/flask_frink
pip install -e lib/flask_frink
```

Quickstart

Flask

Flask-Frink is designed to be used with the Application Factory pattern in Flask.

```
from flask_frink.connection import RethinkFlask

db = RethinkFlask()
```

Then in your application factory, call init_app on the RethinkFlask instance.

```
def create_app():
    ...
    db.init_app(app)
```

Flask-Security

Frink includes `FrinkDatastore` and `FrinkUserDatastore` for Flask-Security compatibility.

Define your `User` and `Role` models.

```
import datetime
from schematics.types.base import (
    StringType, BooleanType, DateTimeType, IntType
)

from schematics.types.compound import (
    ListType, ModelType
)

from flask.ext.security import UserMixin, RoleMixin

from frink.base import BaseModel
from frink.orm import ORMMeta


class Role(with_metaclass(ORMMeta, BaseModel, RoleMixin)):

    name = StringType()
    description = StringType()


class User(with_metaclass(ORMMeta, BaseModel, UserMixin)):

    _uniques = ['email']

    email = StringType()
    password = StringType()
    active = BooleanType(default=True)
    confirmed_at = DateTimeType()
    last_login_at = DateTimeType(default=datetime.datetime.now)
    current_login_at = DateTimeType(default=datetime.datetime.now)
    registered_at = DateTimeType()
    last_login_ip = StringType()
    current_login_ip = StringType()
    login_count = IntType()

    roles = ListType(ModelType(Role))
```

Then in your application factory, initialise this...

```
from flask_frink.datastore import FrinkUserDatastore
from .users.models import User, Role

def create_app():
    ...
    user_datastore = FrinkUserDatastore(db, User, Role)
```

```
    security.init_app(app, user_datastore)
    app.user_datastore = user_datastore
```

API

Datastores

Change Log

0.1.0

CHAPTER 2

Contributing

If you think Frink could be useful for you, you could help get it to some kind of stability by getting involved. Contributors welcome.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search