django-simon Documentation

Release 0.1.0

Andy Dirnberger

January 02, 2014

Contents

Simon is a library to help make working with MongoDB easier. django-imon was created to make it even easier to use Simon with your Django applications.

CHAPTER 1

Installation

To install the latest stable version of django-simon:

```
$ pip install django-simon
```

or, if you must:

```
$ easy_install django-simon
```

To install the latest development version:

```
$ git clone git@github.com:dirn/django-simon.git
$ cd django-simon
$ python setup.py install
```

In addition to django-simon, this will also install:

- Django (1.2 or later)
- PyMongo (2.1 or later)
- Simon

CHAPTER 2

Quickstart

After installing django-simon, import it somewhere in your Django project, like in your `models.py` file.

```
from django_simon import simonize()

simonize()
```

`simonize()` will establish a connection to the database that will be used as the default database for any `Model` classes that you define.

CHAPTER 3

Configuration

`simonize()` looks for the following in your Django project's settings:

`MONGO_URI`	A MongoDB URI connection string specifying the database connection.
`MONGO_HOST`	The hostname or IP address of the MongoDB server. default: 'localhost'
`MONGO_PORT`	The port of the MongoDB server. default: 27017
`MONGO_DNAME`	The name of the database on `MONGO_HOST`. Default: `app.name`
`MONGO_USERNAME`	The username for authentication.
`MONGO_PASSWORD`	The password for authentication.
`MONGO_REPLICA_SET`	The name of the replica set.

The `MONGO_URI` setting will be used before checking any other settings. If it's not present, the others will be used.

By default, `simonize()` will use `MONGO` as the prefix for all settings. This can be overridden by using the `prefix` argument.

Specifying a value for `prefix` will allow for the use of multiple databases.

```
# settings.py

MONGO_URI = 'mongodb://localhost/mongo'
SIMON_URI = 'mongodb://localhost/simon'

# models.py

simonize()
simonize(prefix='SIMON')
```

This will allow for the use of the `mongo` and `simon` databases on `localhost`. `mongo` will be available to models through the aliases `default` and `mongo`. `simon` will be available through the alias `simon`. This alias can be changed by using the `alias` argument.

```
simonize(prefix='SIMON', alias='other-database')
```

API

django_simon.**simonize**(*prefix='MONGO'*, *alias=None*)
Automatically creates a connection for Simon models.

Parameters

- **prefix** (*str*) – (optional) the prefix of the settings
- **alias** (*str*) – (optional) the alias to use for the database connection

New in version 0.1.0.

django_simon.**get_list_or_404**(*model*, **qs*, ***fields*)
Finds and returns a single document, or raises a 404 exception.

This method will find documents within the specified model. If the specified query matches no documents, a `404 Not Found` exception will be raised.

Parameters

- **model** (`simon.Model`) – the model class.
- ***qs** (`simon.query.Q`) – logical queries.
- ****fields** (*kwargs*) – keyword arguments specifying the query.

Returns `QuerySet` – a query set of model instances.

django_simon.**get_object_or_404**(*model*, **qs*, ***fields*)
Finds and returns a single document, or raises a 404 exception.

This method will find a single document within the specified model. If the specified query matches zero or multiple documents, a `404 Not Found` exception will be raised.

Parameters

- **model** (`simon.Model`) – the model class.
- ***qs** (`simon.query.Q`) – logical queries.
- ****fields** (*kwargs*) – keyword arguments specifying the query.

Returns `Model` – an instance of a model.

Full details of how to query using `get_list_or_404()` and `get_object_or_404()` can be found in the `Simon API`.

CHAPTER 5

Further Reading

For more information, check out the Simon docs and the MongoDB docs.

Python Module Index

d