sandman Documentation

Release 0.1

Jeff Knupp

Jul 26, 2018

Contents

Contents:

CHAPTER 1

Installation

Simply run:

```
pip install sandman
```

The requirements aren't set up on PyPI yet but that's easily taken care of below.

Requirements

You'll need to install Flask-SQLAlchemy:

```
pip install Flask-SQLAlchemy
```

OK, that takes care of the requirements...

CHAPTER 2

Quickstart

Create one file with the following contents (which I call `runserver.py`):

```
from sandman.model import register, Model

# Insert Models here
# Register models here
# register((Model1, Model2, Model3))

from sandman import app, db
app.config['SQLALCHEMY_DATABASE_URI'] = '<your database connection string (using␣
↪SQLAlchemy)'
app.run()
```

Then simply run:

```
python runserver.py
```

and try curling your new REST API service!

In a "real" project, you should divide the code into at least two files: one with the "Model" definitions (`models.py`) and the other with the configuration and `app.run()` call (`runserver.py`).

Or you can come up with your own scheme. Whatever.

CHAPTER 3

sandman Package

3.1 sandman Package

Sandman automatically generates a REST API from your existing database, without you needing to tediously define the tables in typical ORM style. Simply create a class for each table you want to expose in the API, give it's associated database table, and off you go! The generated API essentially exposes a completely new system based on your existing data, using HATEOAS.

3.2 exception Module

Exception specifications for Sandman

exception sandman.exception.**JSONException**(*description=None*, *code=400*, *response=None*)

Bases: werkzeug.exceptions.HTTPException

Excecption returned with JSON data rather than HTML

get_body(*environ=None*)

Get the HTML body.

get_headers(*environ=None*)

Get a list of headers.

3.3 model Module

The model module is repsonsible exposes the Model class, from whic user models should derive. It also makes the 'register' function available, which maps endpoints to their associated classes.

class sandman.model.**DatabaseColumnDictMixin**

Bases: object

Set an instances database-relevant attributes from a dict or return a dict containing only database-column attributes of the instance.

as_dict()
Return a dictionary of each of the instance's database columns and their associted values.

from_dict(*dictionary*)
Set the instance's attributes based on a dictionary of instance's database columns.

class sandman.model.**Resource**
Bases: *sandman.model.DatabaseColumnDictMixin*

A RESTful resource

classmethod endpoint()

links()
Return a list of links for endpoints related to the resource.

primary_key()

resource_uri()
Return the URI at which the resource can be found.

sandman.model.**register**(*cls*)
Register an endpoint with the Model class that represents it

3.4 sandman Module

Sandman REST API creator for Flask and SQLAlchemy

sandman.sandman.**add_resource**(*collection*)
Return response for adding a resource

sandman.sandman.**collection_handler**(*collection*)
Handler for a collection of resources

sandman.sandman.**created_response**(*resource*)
Return response for created resource

sandman.sandman.**delete_resource**(*collection*, *lookup_id*)
Return response for deleting a resource

sandman.sandman.**get_session**()
Return a database session

sandman.sandman.**no_content_response**()
Return response when no resource is returned in body

sandman.sandman.**patch_resource**(*collection*, *lookup_id*)
Return response for patching a resource

sandman.sandman.**resource_handler**(*collection*, *lookup_id*)
Handler for single resource

sandman.sandman.**unsupported_method_response**()
Return response when no resource is returned in body

sandman.sandman.**validate**(*cls*, *method*, *resource=None*)

3.5 Subpackages

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

S

Index