MetPetDB documentation

Contents

1. Software requirements and Installation

1.1 Software requirements

Source code	: https://github.com/metpetdb/metpetdb-py
Programming language	: Python 2.7
Web framework	: Django 1.4
Web server	: Apache 2.2
Database	: PostgreSQL

Libraries used

- Psycopg2
- PostGIS
- Mod-wsgi
- JSON

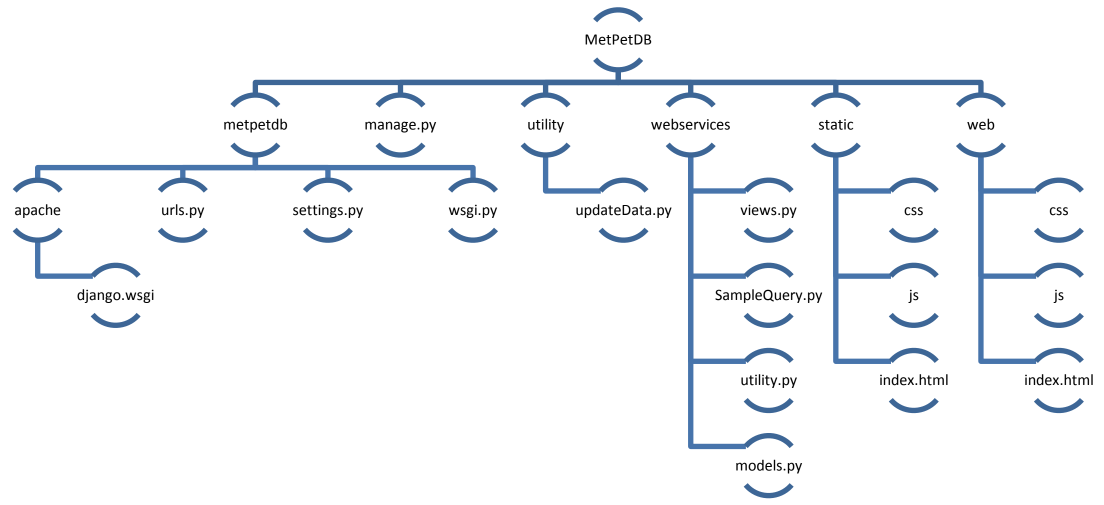
1.2 Installation: Web service / Exhibit

Please refer to the installation notes.

2. Architecture

2.1 Overview

This section describes the main components of application.



2.2 Component description

2.2.1 urls.py

This file is used to associate a URL with its ‘view’.

In our program we set the web service URL at ‘api/metpetdb/’ to a view function defined at ‘webservices.views.metpetdb’.

urlpatterns+=staticfiles_urlpatterns()

Is added at the end of the file to append files in the static directory. The location of the static directory is defined in the settings.py file.

2.2.2 settings.py

This file defines the Django settings for the project
The following parameters are set for the MetPetDB project.

- DATABASES: used to specify the credentials used to access the PostgreSQl database.
- STATIC_ROOT: used to define the location of the static files
- STATIC_URL: used to define the URL for the static files
- STATICFILES_DIRS: used to define the location of the static files from which they are copied to the STATIC_ROOT directory.
    - *Django-admin.py collectstatic* Is used to collect the static files and copy them into the static_root directory.
- INSTALLED_APPS: used to specify applications used by Django. The application used for metpetdb is called ‘webservices’.

2.2.3 views.py

Contains functions for different views in metpetdb.

- metpetdb
    - New metpetdb web service.
    - It parses the data from the parameters, constructs a query and generates the output.
    - General execution pattern can be visualized as,
      GET parameters → SampleQuery.py (construct query) → utility.py (generate output
- samples
    - Samples web service for ‘Exhibit’.
- chemical_analyses
    - Chemical analyses web service for ‘Exhibit’.

2.2.4 utility.py

This file contains helper functions to read data from the database and generate output for the views in JSON or HTML format. This file mainly contains three helper functions.

- getFacetJSON(query): This function executes the query passed to it against the underlying database and returns facet data as a JSON array.
- getAllJSON(query): This function executes the query passed to it against the underlying database and returns all the data as a JSON array.
- getSampleResults(query): This function executes the query passed to it against the underlying database and returns the results in an HTML table format.

2.2.5 SampleQuery.py

This file is used to construct an SQL query with the parameters selected by the user. The resulting query is used by the utility.py file to generate the output.

2.2.6 django.py / wsgi.py

Contains configuration information to connect to the Apache webserver.