David Nevado Catalán
Luis C. Jariego Pérez

REPORT

DJANGO ARCHITECTURE

Django is an open-source Python Web framework which follows the model-view-controller architectural pattern. It allows the modularization of websites, providing reusability and simplification. This is how it works:

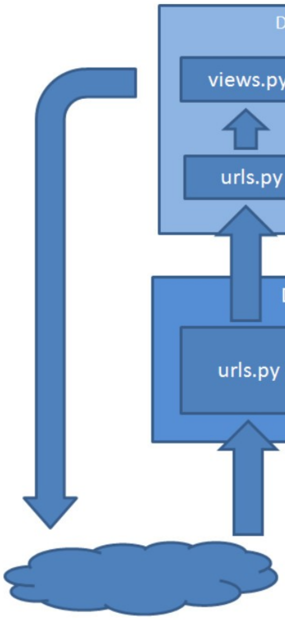
Our web server contains a Django Project, to which one or multiple Django applications can be plugged in. The Django project holds the common elements for its applications: the relational database table, the root URL parser and the base HTML template. The applications have their own (more specific) URL parser and HTML templates that will complement the project's. When an application is plugged into a project, the database needs to be synchronized with the application's data model.

In the process of handling a request there are several files involved from both the project and the application.

When a request for a URL arrives, the *urls.py* file in the Django project will take control and parse it, partially, and will decide which application should take it from there. Once it is chosen, the URL will be sent to the application.

The Django application contains 3 main python files: *urls.py, views.py* and *models.py,* that will handle the rest. The *urls.py* file is in charge of analyzing the URL (passed by the Django Project) and deciding which *views.py* function is called in order to load an HTML template. The *models.py* file contains the classes (data models) that describe the data in the project's database, to be able to access it and be able to show the complete web page to the client.

A simple diagram that shows the Django architecture we have been describing:


Django Application
views.py
HTML Templates
urls.py
models.py
Django Project
urls.py
DB