Django Architecture Summary

G1391 P6
Daniel Mohedano
Silvia Sopeña

October 26, 2020

Django projects are formed by different Django applications. This can have two reasons:

- To distribute the workload of the whole project in small, more specific applications.
- Because very different functionalities are needed and thus it only make sense to implement them in different applications.

Either way, they all work together in the same project.

Django applications (and also in general the Django project related with those applications) apply the MVC pattern: Model, Controller and View. This allows a more organized structure.

Starting with the Model, it encapsulates everything related to data structures and the data that needs to be stored. The project holds and takes responsibility for the whole database where all the data is stored. Meanwhile, each application is responsible for defining their data structures, relations between them, etc. in *models.py*.

The View refers to how the contents are viewed. For the web projects this is all neatly contained in the HTML templates. Each application can have its own templates that define how the contents will be organized and displayed.

The Controller is implemented by two different modules: *views.py* and *urls.py*. The views are the main source of control, where all the code is stored in regards to processing requests of pages, processing uploaded forms, etc. The views also render the HTML templates so that they can be viewed. In order to make all work, these views are then associated with certain URLs through the *urls.py* module. This module can appear both in the application and the project in general, allowing for example to define a certain URL pattern for each application (in the project's module) and then later defining the specific associations between views and URLs in the application's module. This allows for a cleaner and more structured way of defining the URLs and its relations instead of having all URLs of all current applications of the project stored in the same place.

```
(psi) eps@labvirteps:~/Documentos/psi_1391_6_p2/tango_with_django_project$ cover
age report -m -i
Name                                                Stmts   Miss  Cover   Missing
---------------------------------------------------------------------------------
rango/__init__.py                                       0      0   100%
rango/admin.py                                          9      0   100%
rango/apps.py                                           3      3     0%   1-5
rango/forms.py                                         34      3    91%   36-39
rango/migrations/0001_initial.py                        6      0   100%
rango/migrations/0002_auto_20201011_1519.py             4      0   100%
rango/migrations/0003_category_slug.py                  4      0   100%
rango/migrations/0004_auto_20201011_1721.py             6      0   100%
rango/migrations/0005_auto_20201011_1749.py             6      0   100%
rango/migrations/0006_auto_20201020_0843.py             8      0   100%
rango/migrations/0007_auto_20201022_1426.py             6      0   100%
rango/migrations/__init__.py                            0      0   100%
rango/models.py                                        35      1    97%   52
rango/templatetags/__init__.py                          0      0   100%
rango/templatetags/rango_template_tags.py               6      0   100%
rango/urls.py                                           4      0   100%
rango/views.py                                        102     19    81%   44-45, 53, 7
5-77, 84-85, 101-104, 129-132, 152-155, 163, 168-169
---------------------------------------------------------------------------------
TOTAL                                                 233     26    89%
(psi) eps@labvirteps:~/Documentos/psi_1391_6_p2/tango_with_django_project$
```

Figure 1: Coverage

Side note: we originally made the project inside a tango_with_django_project folder. So originally our git repository only had that folder with the project inside. After reaching the Heroku part of the practice we found ourselves needing to create a second git repository, inside the project folder, in order to upload the application to the servers. That is the reason why there are two .git folders in the deliverable.