Case : Integrating third parties

Files

The main directory of this assignment is the Django project folder, called *maykin*. Along the usual files and folders inside of it, the directory *hotels* contains all the code for the "hotels" app.

Inside this *hotels* folder there are usual python scripts for Django apps, such as :
— *models.py* : contains the data structures to store city and hotel data.
— *forms.py* : contains form shape for POST requests (only cities names)
— *urls.py* : contains paths to redirect urls.
— *views.py* : contains functions that define what will see the end user.

Inside this folder, there are also two *.csv* files called *city.csv* and *hotel.csv*, that contain cities and hotels data locally. These files are stored only as a "back up plan", if it's impossible to download the files from internet.

The python script responsible of downloading/loading the files in the database, is stored in the directory *hotels/management/commands/* and is called *loadfiles*.

The view functions use two *.html* templates stored in the folder *templates* of the base dir : *hotels.html* and *hotels_complete.html*. The first one allows the user to select a city with a pull down, while the second works with an auto-complete field.

How to use

This assignement was developed and tested using Ubuntu 16.04.9, Django 2.0.3 and Python 3.5.2.

Export the *maykin* folder to your pc.

Import the data

To load the *.csv* files to the database, use the following command in a terminal inside the *maykin* base folder :

```
python3 manage.py loadfiles
http://rachel.maykinmedia.nl/djangocase/city.csv
http://rachel.maykinmedia.nl/djangocase/hotel.csv internet
```

If it doesn't work (**Couldn't download file <url>**), use the following command, that will load from local files (*city.csv* and *hotel.csv* from the *hotels* directory, for example) :

```
python3 manage.py loadfiles path_to_city_file path_to_hotel_file local
```

This local solution was crucial for me because after some time I couldn't download the *.csv* files from the internet (403 errors).

Test the *hotels* app

Inside the *maykin* base folder, type the following command to run the server :

```
python3 manage.py runserver
```

You can now access the Django development server at your local IP address, with the port defined in your Django configuration (for example : http ://127.0.0.1 :8000/).

You can use the pull down to choose a city, and use the refresh button to update the list of available hotels for the selected city.

Other fonctionnalities

Auto-complete : You can activate the auto-complete mode by setting the *autocomplete* variable of *maykin/hotels/views.py* to True. This configuration will show on the view an auto-complete search field (powered by *jQuery Autocomplete*) instead of the pull-down.

cronjob : If you want the *.csv* files to be stored automaticaly, every day, at midnight on the database, you can use the following crontab configuration :

```
0 0 * * * python3 path_of_maykin_project/manage.py loadfiles
http://rachel.maykinmedia.nl/djangocase/city.csv
http://rachel.maykinmedia.nl/djangocase/hotel.csv
```

This configuration was tested on the */bin/nano* file.