Indice

Pubblicare un progetto Django su pythonanywhere.com

I passaggi necessari ad ottenere il risultato voluto, sono i seguenti:

1. carica il tuo codice su PythonAnywhere
2. imposta un virtualenv e installa Django e tutti gli altri requisiti
3. configura la tua app web usando l'opzione ***manual config***

1. Carica il tuo codice su PythonAnywhere

La soluzione più immediata sarebbe quella di clonare il repository presente su GitLab o GitHub direttamente all'interno della propria virtual machine, ma avendo registrato un account gratuito, pythonanywhere permette connessioni https ed ssh solamente a siti o servizi presenti in una loro **whitelist**, che non include il server ***dw.gnet.it***.

Perciò l'unica alternativa è caricare manualmente il progetto django compresso. E' consigliabile utilizzare il formato **.tar.gz** in modo da poterlo comprimere direttanente dalla **bash** della vm con il comando:

```
tar -xvzf cruscanalyzer_base_folder.tar.gz.
```

2. Imposta un virtualenv e installa Django e tutti gli altri requisiti

Fatto questo, è necessario creare un virtual environment sulla macchina per poter installare la versione più recente di Django, in particolare:

```
mkvirtualenv --python=/usr/bin/python3.6 cruscanalyzer-venv
```

Dopo di che sarà necessario installare i pacchetti richiesti dall'applicativo nel nuovo ambiente virtuale, i comandi da inserire sono i seguenti:

```
1. workon cruscanalyzer-venv
2. (cruscanalyzer-venv) pip install django
3. (cruscanalyzer-venv) pip install nltk
4. (cruscanalyzer-venv) pip install googletrans
```

3. Configura la tua app web usando l'opzione manual config

Seguendo la procedura suggerita dal sito, andare sul tab **Web** del sito, crea una nuova applicazione e impostare la **configuazione manuale (manual config)**.

Inserisci il nome del tuo virtualenv

Una volta fatto, nella sezione **Virtualenv**, inserisci il path del tuo virtualenv creato, per esempio:

```
/home/cruscanalyzer/.virtualenvs/cruscanalyzer-venv
```

Opzionale: inserisci il path alla cartella del codice sorgente

Sopra alla sezione **Virtualenv**, nella sezione **Code**, puoi inserire il path alla cartella del progetto Django per avere un link veloce ai file contenuti in essa, il path sarà il seguente:

```
/home/cruscanalyzer/cruscanalyzer_base_folder
```

Modifica il file WSGI

Clicca sul link al file WSGI, nella sezione **Code**, per modificare le impostazioni e far funzionare l'applicazione Django. Il link è il seguente:

https://www.pythonanywhere.com/user/cruscanalyzer/files/var/www/cruscanalyzer_pythonanywhere_com_wsgi.py?edit

In esso, cancella **tutto** il contenuto ed inserisci **solamente** le seguenti righe di codice:

```
# ++++++++++ DJANGO ++++++++++
# To use your own Django app use code like this:
import os
import sys

# assuming your Django settings file is at
'/home/myusername/mysite/mysite/settings.py'
path = '/home/cruscanalyzer/cruscanalyzer_base_folder'
if path not in sys.path:
    sys.path.insert(0, path)

os.environ['DJANGO_SETTINGS_MODULE'] = 'cruscanalyzer.settings'

## Uncomment the lines below depending on your Django version
###### then, for Django >=1.5:
from django.core.wsgi import get_wsgi_application
application = get_wsgi_application()
###### or, for older Django <=1.4
#import django.core.handlers.wsgi
#application = django.core.handlers.wsgi.WSGIHandler()
```

Successivamente, **salva** il file e premi il tasto **Reload** all'interno del tab Web della tua vm.

NOTA BENE:
Per far funzionare il nuovo applicativo, è importante inserire nel file **settings.py** del progetto Django la URL del servizio cloud nel nostro progetto nella riga **ALLOWED_HOST**:

```
ALLOWED_HOSTS = ['cruscanalyzer.pythonanywhere.com']
```

Configurazione dei file statici (static) per la web app

Da **BASH** attivare il proprio virtualenv:

```
workon cruscanalyzer-venv
cd cruscanalyzer_base_folder/
```

Modificare il file settings.py ed inserire la seguente variabile:

```
STATIC_ROOT = os.path.join(BASE_DIR, "static")
```

In questo modo, Django saprà dove inserire tutti i file statici (css, js, img, ecc,) che saranno ottenuti con il comando successivo. Eseguire il seguente comando per inserire nella cartella root **static/** del progetto tutti i file statici necessari per il funzionamento della web app (**inclusi i file per le pagine admin**):

```
python manage.py collectstatic
```

Fatto questo, dal tab Web recarsi nella sezione Static ed importare in **una singola riga** i due path necessari al caricamento dei file css, javascript e immagini locali. I path da inserire sono i seguenti:

```
/static/ → URL
/home/cruscanalyzer/cruscanalyzer_base_folder/static/ → Directory
```

Dopo di che sarà necessario fare nuovamente il **Reload** della Web App.

Configurazioni aggiuntive

E' possibile eliminare i file prodotti da python, pycharm e pipenv che non sono necessari al funzionamento del server, i file sono i seguenti:

- `.idea/`
- `__pycache__/`
- `Pipfile`
- `Pipfile.lock`

Visualizzare il database

Per poter visualizzare lo schema e il contenuto del database di django (in automatico viene creato un database SQLite, ma è possibile cambiarlo), è necessario seguire i seguenti passi:

- Cliccare sulla tab **database** in alto a destra di Pycharm; si aprirà una finestra in cui sarà possibile visualizzare i database presenti nel progetto.
- Premere **MAIUSC+INVIO**, o cliccare sull'icona del database con i settings *Data Source Properties.*
- Dovrebbe essere già presente in automatico il database db, collegato al file db.sqlite3. Nel caso non fosse così, aggiungere questo file.
- Aggiornare i driver se necessario (messaggio a fondo finestra).
- In *comment*, inserire il percorso per raggiungere il file **settings.py** del progetto.
- Applicare i cambiamenti e selezionare ok.

Generare la documentazione del progetto

Per generare la documentazione del progetto, è stato usato lo stile **reST** (uso di docstring per classi e funzioni) come format e **sphinx** come tool per la generazione di documentazione automatica.
Per visualizzare la documentazione del progetto, aprire da browser il file **index.html** in */doc/_build/html/*.

Passi per l'installazione di sphinx:

- Entrare nel virtual enviroment del progetto (nel nostro caso, pipenv).
- Eseguire da terminale `pip3 install sphinx` → Installazione sphinx
- Creare cartella *"doc"* nella cartella principale del progetto.
- Inizializzare sphinx eseguendo `sphinx-quickstart`
- Inserire in */doc/conf.py*

```
import os
import sys
import django
sys.path.insert(0, os.path.abspath('..'))
os.environ['DJANGO_SETTINGS_MODULE'] =
'Your_project_name.settings'
django.setup()
```

Sempre in */doc/conf.py*, aggiungere alla lista extension la variabile `'sphinx.ext.autodoc'`

Passi per la generazione automatica della documentazione:

- Eseguire dalla cartella */doc* `sphinx-apidoc -o . ..`
- Eseguire `make html`

Set up dell'ambiente python per eseguire i test dell'applicazione

Requisiti

Per eseguire i test presenti nel progetto, è necessario avere una **versione** di **Python** uguale o superiore alla **3.6**.

Installare pipenv e creare l'ambiente virtuale

Da terminale Unix o Prompt dei comandi di Windows digitare:

```
python -m pip install pipenv
```

Dopo di che è necessario recarsi nella cartella:

```
team-8/code/cruscanalyzer_base_folder/
```

A questo punto il comando da dare dipende dalla **versione** di Python **installata** nel SO:

→ Python 3.6:

```
pipenv install
```

→ Python maggiore di 3.6:

```
pipenv install --python path/to/python
```

Al termine dell'esecuzione di uno dei due comandi, verrà creato l'ambiente virtuale (con i necessari pacchetti python) necessario all'esecuzione di tutto il codice presente nella cartella cruscanalyzer_base_folder/.

Per attivare l'ambiente appena creato, sempre all'interno della cartella sopra citata, bisogna dare il seguente comando:

```
pipenv shell
```

Esecuzione dei test

A questo punto è possibile eseguire i seguenti test (**da terminale**):

1. **`$../cruscanalyzer_base_folder/`** `python manage.py test`
   `per eseguire tutti i test del progetto.`

Per maggiori informazioni sulle altre tipologie di test da poter eseguire, si rimanda al file ***team-8/test/test_strategy***.