Curs de Python – Telecogresca
Aplicació de frases de benvinguda

Alex Barceló

18 de juliol de 2016

0 Preparació del `virtualenv`

Començarem per preparar un `virtualenv`. Això ho farem per assegurar-nos de no "contaminar" el sistema amb els paquets (i les versions) que desitgem instal·lar.

En el següents exemples suposaré que la carpeta de l'usuari és el directori de treball i el lloc desitjat on treballar.

Si encara no tenim el mòdul `virtualenv` el podem aconseguir:

- En **Ubuntu** a través del paquet `virtualenv`.
- En general, podem instal·lar el mòdul a tot el sistema mitjançant la comanda `sudo pip install virtualenv`

Per a crear la carpeta que serà el nostre *virtual environment* només cal executar:

```
~ $ virtualenv tgkenv
~ $ source tgkenv/bin/activate
(tgkenv) ~ $ echo "Look mom, I am in a virtualenv!"
```

La segona comanda s'encarrega d'assegurar que l'intèrpret de Python "actiu" serà el de l'entorn virtual, i no el de sistema.

Ja podem obtenir el Django simplement utilitzant la comanda `pip` (sense `sudo`! aquesta és una de les avantatges del `virtualenv`).

```
(tgkenv) ~ $ pip install django==1.9
```

1 Inici del projecte

[Nota: assegureu-vos de tenir el `virtualenv` actiu, si ho voleu fer així. Això implica realitzar la comanda `source tgkenv/bin/activate` o similar prèviament, i fer-ho en totes les consoles que utilitzeu]

En **Django** podem començar l'estructura d'arxius amb la següent comanda:

```
(tgkenv) ~ $ django-admin startproject tgkhandson
(tgkenv) ~ $ cd tgkhandson
(tgkenv) ~/tgkhandson $ ./manage.py startapp benvinguda
```

Després de realitzar les comandes anteriors haureu obtingut una estructura d'arxius que contindrà un projecte Django, amb una configuració inicial força raonable, i també haureu crear l'estructura d'arxius de l'aplicació de `benvinguda` que és la que programarem en aquest *hands-on*.

Per a indicar que volem utilitzar l'aplicació `benvinguda` cal obrir l'arxiu `settings.py` i modificar la variable `INSTALLED_APPS`, concretament:

```
INSTALLED_APPS = [
    'benvinguda',
    'django.contrib.admin',
    ...  # deixar la resta de linies iguals
```

2 Definició del model de dades

El primer que necessitarem és definir el model de dades que Django que s'utilitzarà per a guardar les frases de benvinguda. No cal tenir coneixements de SQL, de bases de dades, o similars; n'hi ha prou utilitzant els `model` de Django.

El model de dades serà molt senzill, en el nostre cas. Només utilitzarem un cap `missatge`, i deixarem (tot i que avui no mostrarem com utilitzar-ho) un camp `only_anonymous` que ens permetria en un futur filtrar les frases de benvinguda en funció si l'usuari està loggejat o encara no.

Els models es defineixen a l'arxiu `models.py` (carpeta `benvinguda`). El contingut del model pot ser el següent:

```
class WelcomeFrase(models.Model):
    """Sentences to greet the user.

    The application contains a list of sentences and,
    randomly, one will be used in order to greet the
    user (at the main page).
    """
    class Meta(object):
        verbose_name = "frase de benvinguda"
        verbose_name_plural = "frases de benvinguda"

    missatge = models.CharField('missatge', max_length=100)
    only_anonymous = models.BooleanField('nomes per anonims',
                                         default=True)
```

Un cop hem definit el model (que és una simple classe Python) procedim a indicar a Django que volem utilitzar aquest model de dades. El que internament fa és crear les

taules a la base de dades[1], amb els seus camps, relacions, etc.

El procediment és senzill:

```
(tgkenv) ~/tgkhandson $ ./manage.py makemigrations
(tgkenv) ~/tgkhandson $ ./manage.py migrate
```

L'operació està dividida en una preparació (`makemigrations`) que comprova la consistència dels models i que la operació sigui correcte i l'execució dels canvis (`migrate`). Per a una aplicació real, el `makemigrations` el realitzaria el desenvolupador i un cop comprovada la correctesa, l'administrador executaria `migrate` en el servidor i la base de dades de producció.

3 Introduïnt frases

Django té una interfície d'administració molt potent, i activada per defecte en les darreres versions. Per a utilitzar-la hem d'indicar que desitgem que el model de dades introduït prèviament sigui utilitzat en l'admin. Per a fer-ho hem d'afegir (deixeu les línies ja existents) el següent a l'arxiu `benvinguda/admin.py`:

```
1 from .models import WelcomeFrase
2
3 admin.site.register(WelcomeFrase)
```

Creem un usuari administrador i iniciem el servidor de proves:

```
(tgkenv) ~/tgkhandson $ ./manage.py createsuperuser admin
  <... completar les dades que se'ns demanen ...>
(tgkenv) ~/tgkhandson $ ./manage.py runserver
```

I ara ja podem accedir a la URL `http://127.0.0.1:8000/admin/` i introduir frases de benvinguda.

4 Preparant Controller i View

A continuació prepararem:

Controller escull una frase aleatòria (o seguint certs criteris)

View mostra al visitant la web, que contindrà la frase escollida

Vigileu que l'estructura clàssica Model-View-Controller en Django es conserva, però s'anomenen Model-Template-View. L'abús de la paraula *view* sovint ocasiona certes confusions.

Comencem repassant el codi del `views.py` (el típicament anomenat *Controller*):

[1] En el nostre cas, Django estarà utilitzant una base de dades **SQLite**, que és molt adecuada per fer proves i per a desenvolupament però mai s'ha d'utilitzar en producció

```python
from django.views.generic import TemplateView
from benvinguda.models import WelcomeFrase
from random import choice

class HomeView(TemplateView):
    """First page view."""
    template_name = 'home.html'

    def get(self, request, *args, **kwargs):
        """Put the 'welcome' in context, template will show it."""
        context = {
            'welcome': choice(WelcomeFrase.objects.all()).missatge,
        }
        return self.render_to_response(context)
```

Veiem una selecció aleatòria molt simple. Un cop definida la variable `context`, la feina és ara del *template* per a generar el HTML que anirà a l'usuari (típicament, la capa de *view* en un Model-View-Controller). L'arxiu s'ha d'anomenar `home.html` i el crearem a la carpeta `benvinguda/templates` amb el següent contingut:

```
<h1>{{ welcome }} </br><small>esca, esca, esca, Telecogresca!</small></h1>
```

Sou lliures de millorar aquest template, però aquí utilitzo una mostra minimalista de template HTML quasi-vàlid (els navegadors ho acceptaran, tot i que no és un document vàlid en el seu estat actual).

5 Rutes

Ja quasi hem acabat, però si us hi fixeu encara no hem indicat quina URL haurà de mostrar la benvinguda. La relació entre URL i controladors associats s'acostuma a fer mitjançant l'arxiu *urls.py*. La complexitat de les rutes pot ser molt alta, però per a l'exemple que estem realitzant, hi haurà prou modificant l'arxiu fins arribar al següent estat:

```python
from django.conf.urls import url
from django.contrib import admin
from benvinguda.views import HomeView

urlpatterns = [
    url(r'^benvinguda/', HomeView.as_view()),
    url(r'^admin/', admin.site.urls),
]
```