

Django erabiltzen

Adibide batean oinarrituta



Laborategiko instalazioa

- Zerbitzaria: 158.227.142.232
- SSH bidez konektatzeko erabili:

```
ssh -XC user##@158.227.142.232
```

non ## erabiltzaile bakoitzaren digituak diren.
- Kontuak:
 - Erabiltzaileak: user01, user02...
 - Pasahitzak: user01, user02...
- MySQL eta phpmyadmin
 - <https://158.227.142.232:12322>
 - Nork bere datu-baseak sortuko ditu.
 - Erne!!! Sortzen diren datu-base guztiak denon eskura egongo dira.
 - Beraz:
 - Beti jarri behar dira datu-basearen izenaren bukaeran norberari dagozkion ## digituak



Tutorialaren 1. partea

- <http://teknikariak.informatika.ehu.es/Django/intro/tutorial01.html>
- **Bozketen aplikazioa.** Bi zati izango ditu:
 - Alde publikoa, aukera emango duena bozketak ikusi eta aukerak bozkatzeko.
 - Kudeaketarako gunea (*admin*), aukera emango duena bozketak gehitu, aldatu eta ezabatzeko.

Lanera!



Django proiektu baten kudeaketa

- Django proiektuak kudeatzeko bi utilitate daude: *django-admin.py* eta *manage.py*
 - *django-admin.py* orokorra da
 - *manage.py* automatikoki sortzen da proiektu bat sortzen den aldiro
 - Zehaztasun gehiago nahi izatera: ikusi
 - <http://teknikariak.informatika.ehu.es/Django/ref/django-admin.html>



Django proiektu baten sorrera

- Sortu Django proiektu bat

- `$ django-admin.py startproject nireproiektua`

- Gure lan-ingurunean:

```
$ python /usr/lib/python2.7/dist-packages/django/bin/django-admin.py  
startproject nireproiektua
```

- Direktorioa

```
nireproiektua/  
    manage.py  
nireproiektua/  
    __init__.py  
    settings.py  
    urls.py  
    wsgi.py
```



Direktorioko fitxategiak

- **Kanpoko nireproiektua/** direktorioa proiektuaren edukiontzia da.
- **manage.py:** *python* fitxategia, proiektuarekin lan egiteko aukera ematen du. `django-admin.py` eta `manage.py` erabiltzen dira.
- **Barruko nireproiektua/** direktorioa da, izatez, proiektuaren benetako Python paketea. Paketearen izen hori erabili behar da baliabideak inportatzeko, adibidez: `import nireproiektua.settings`.
- **nireproiektua/__init__.py:** Fitxategi hutsa, soilik Python-i adierazteko direktorio hau paketetzat hartu behar dela.
- **nireproiektua/settings.py:** Proiektuaren ezarpenak eta konfigurazioa.
- **nireproiektua/urls.py:** Proiektuaren URL erazagupenak. Edukien taula moduko bat.
- **nireproiektua/wsgi.py:** Proiektuaren eta WSGI zerbitzariaren arteko lotura.



Egiaztapena eta martxan jartzea

- Kanpoko direktorioan:

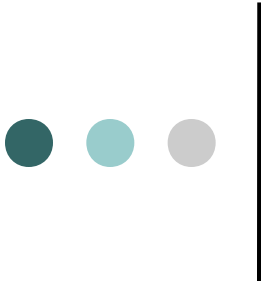
```
$ python manage.py runserver 158.227.142.232:80##
```



Proiektuaren ezarpenak. Datu-basea

- *nireproiektua/settings.py* editatu

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'djangogunea##',  
        'USER': 'user##',  
        'PASSWORD': 'user##',  
        'HOST': '',  
        'PORT': '12322',  
    }  
}
```

Proiektuaren ezarpenak

Besterik ezeko aplikazioak

Besterik ezean, `INSTALLED_APPS`en honako aplikazioak datoz:

- `django.contrib.auth` – Autentifikazio-sistema.
- `django.contrib.contenttypes` – Edukien motetarako *frameworka*.
- `django.contrib.sessions` – Saiotetarako *frameworka*.
- `django.contrib.sites` – Gune bat baino gehiago maneiatzeko instalazio berean.
- `django.contrib.messages` – Mezuak lantzeko *frameworka*.
- `django.contrib.staticfiles` – Fitxategi estatikoak maneiatzeko *frameworka*.
- Aplikazio hauek gutxienez datu-base bana erabiltzen dute. Beharrezkoa da, datu-base horiek indarrean jartzea:
 - `$ python manage.py syncdb`



Aplikazioa sortzen

- Proiektuak eta aplikazioak
 - Proiektua kontzeptu zabalagoa da.
 - Proiektu batean webgune bateko aplikazioak sartzen dira (beren konfigurazioarekin).
 - Aplikazioa (web-aplikazioa) lotuago dago erabilpen jakin bati, web-aplikazioaren ideiarekin.



Aplikazioa sortzen (II)

```
$ python manage.py startapp bozketak
```

```
bozketak/
```

```
    __init__.py
```

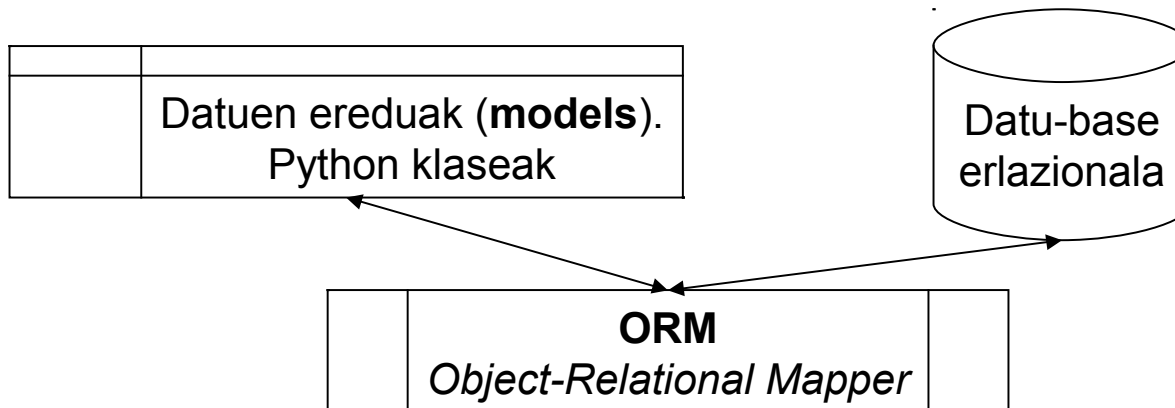
```
    models.py
```

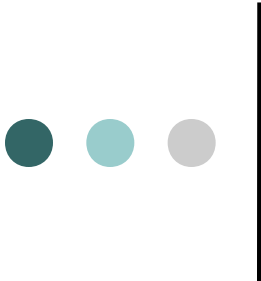
```
    tests.py
```

```
    views.py
```

Ereduak (*models*)

- Ereduetan datuen deskribapenak gordetzen dira. Hor adierazten dira gordetako datuen eremuak, erlazioak eta portaera.
- *DRY* printzipioa: *Don't Repeat Yourself*
- *ORM*





Ereduak (*models*)

- Eredu bakoitza klase batean errepresentatzen da
 - Klase horiek `django.db.models.Model` gainklasetik eratortzen dira.
 - Klase-aldagaiek datu-baseko eremuak adierazten dituzte.
 - <http://teknikariak.informatika.ehu.es/Django/topics/db/models.html>
- Editatu ***bozketak/models.py***



Aktibatu ereduak

- Ereduetatik informazio asko eta aberatsa ateratzen da.
- Django gai da ereduaren informaziotik datu-baseak sortzeko.
- Beraz, django proiektuko datuen modelizazioa eta datu-basearen diseinua bat etorriko dira ezinbestez.
- Lehenbizi, **bozketak** aplikazioa instalatu behar dugu datu-basearen sorrera eta erabilera aktibatzeko.
 - Editatu *settings.py* fitxategia, eta aldatu `INSTALLED_APPS` ezarpena '**bozketak**' txertatzeko
 - `python manage.py sql bozketak`
 - `python manage.py syncdb`
 - `syncdb` aukerak datu-ereduetatik datu-basea sortzen du, baina alderantzizko bidea ere egin daiteke `inspectdb` aukera erabiliz. Ikusi, adibidez:
 - <http://teknikariak.informatika.ehu.es/Django/ref/django-admin.html#inspectdb>
 - Phpmyadmin interfazeaz aztertu nola geratu den
 - Python-en *shell* interaktiboa erabiltzea interesgarria da eta bertan **database APIa** aztertzea.
 - Objektuak sortu, kontsultatu, metodoak sortu...
 - Datu-basean erregistroak sartu...