

## Conda

Όταν ασχολείστε με πολλά projects σε python είναι βασικό κάθε project να είναι απομονωμένο. Αυτό σημαίνει ότι κάθε project έχει τις δικές του βιβλιοθήκες, πακέτα python κτλ. Σε τεχνική ορολογία αυτό ονομάζεται virtualization ή αλλιώς ότι κάθε project "τρέχει" στο δικό του απομονωμένο περιβάλλον.

Υπάρχουν πολλά πακέτα σε python που προσφέρουν αυτή τη λειτουργικότητα. Για παράδειγμα:

- [conda \(https://docs.conda.io/en/latest/\)](https://docs.conda.io/en/latest/)
- [virtualenv \(https://virtualenv.pypa.io/en/latest/\)](https://virtualenv.pypa.io/en/latest/)
- [pipenv \(https://pipenv-fork.readthedocs.io/en/latest/\)](https://pipenv-fork.readthedocs.io/en/latest/)

Εδώ θα ασχοληθούμε με το conda. Το conda "έρχεται" (δηλαδή εγκαθιστάται αυτόματα) όταν εγκαθιστάτε την anaconda. Για να τρέξετε conda θα πρέπει να βεβαιωθείτε ότι είναι εγκατεστημένο:

```
In [6]: !/Users/alexandroskanterakis/anaconda3/bin/conda -V
conda 4.5.11
```

Για να δείτε τα εικονικά περιβάλλοντα που υπάρχουν ήδη στον υπολογιστή σας τρέξτε:

```
In [10]: !/Users/alexandroskanterakis/anaconda3/bin/conda info --envs

# conda environments:
#
base                                /Users/alexandroskanterakis/anaconda3
HLA_manos                          /Users/alexandroskanterakis/anaconda3/envs/HLA_manos
MutationInfo2                      /Users/alexandroskanterakis/anaconda3/envs/MutationIn
fo2
airflow_python                    /Users/alexandroskanterakis/anaconda3/envs/airflow_py
thon
findbase_python                   /Users/alexandroskanterakis/anaconda3/envs/findbase_p
ython
jacob_python                      /Users/alexandroskanterakis/anaconda3/envs/jacob_pyth
on
new_project                        * /Users/alexandroskanterakis/anaconda3/envs/new_projec
t
obc_python                        /Users/alexandroskanterakis/anaconda3/envs/obc_python
obc_validator                     /Users/alexandroskanterakis/anaconda3/envs/obc_valida
tor
python_lesson                     /Users/alexandroskanterakis/anaconda3/envs/python_les
son
python_project                   /Users/alexandroskanterakis/anaconda3/envs/python_pro
ject
tsatsara_python                 /Users/alexandroskanterakis/anaconda3/envs/tsatsara_p
ython
zazz_python                      /Users/alexandroskanterakis/anaconda3/envs/zazz_pytho
n
```

**ΠΡΟΣΟΧΗ!** Τις παρακάτω εντολές τρέξετε "έξω" από το jupyter! Το jupyter "τρέχει" ήδη σε κάποια python του συστήματός σας!

Μπορείτε να φτιάξετε ένα καινούργιο περιβάλλον με την εντολή:

```
In [ ]: !/Users/alexandroskanterakis/anaconda3/bin/conda create -n my_fabulous_project
python=3.7
```

Μπορείτε να αλλάξετε το όνομα (my\_fabulous\_project) ή την έκδοση της python (3.7) στην οποία θα βασίζεται

Για να "μπείτε" ή αλλιώς να ενεργοποιήσετε το εικονικό περιβάλλον μπορείτε να τρέξετε:

```
In [ ]: !source /Users/alexandroskanterakis/anaconda3/bin/activate my_fabulous_project
```

Από τη στιγμή που μπείτε έχετε μία "ανεξάρτητη" έκδοση της python. Εκεί για παράδειγμα μπορείτε να βάλετε "πακέτα" τα οποία θα υπάρχουν μόνο σε αυτό το περιβάλλον. Για παράδειγμα μπορείτε να τρέξετε:

```
In [ ]: !pip install requests
!pip install jupyter
```

Αυτά τα πακέτα τώρα θα υπάρχουν μόνο στο περιβάλλον με τον όνομα my\_fabuloys\_project.

Για να "βγείτε" από αυτό το περιβάλλον τρέξετε:

```
In [ ]: !source /Users/alexandroskanterakis/anaconda3/bin/deactivate
```

Τέλος, για να διαγράψετε τελείως ένα περιβάλλον τρέξετε:

```
In [ ]: !/Users/alexandroskanterakis/anaconda3/bin/conda remove --name my_fabuloys_project --all
```

Διαβάστε περισσότερα:

- <https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html> (<https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html>)
- [https://en.wikipedia.org/wiki/Conda\\_\(package\\_manager\)](https://en.wikipedia.org/wiki/Conda_(package_manager)) ([https://en.wikipedia.org/wiki/Conda\\_\(package\\_manager\)](https://en.wikipedia.org/wiki/Conda_(package_manager)))

## Django

Το [Django](https://www.djangoproject.com/) (<https://www.djangoproject.com/>) ([wiki](https://en.wikipedia.org/wiki/Django_(web_framework)) ([https://en.wikipedia.org/wiki/Django\\_\(web\\_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)))) είναι ένα περιβάλλον σε python για ανάπτυξη web εφαρμογών το οποίο βασίζεται στο μοντέλο [Model View Controller](https://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller) (<https://en.wikipedia.org/wiki/Model%E2%80%93View%E2%80%93Controller>). Για να το εγκαταστήσετε:

```
In [13]: !pip install django
```

```
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
Requirement already satisfied: django in /Users/alexandroskanterakis/anaconda3/envs/new_project/lib/python3.7/site-packages (3.0.2)
Requirement already satisfied: asgiref<=3.2 in /Users/alexandroskanterakis/anaconda3/envs/new_project/lib/python3.7/site-packages (from django) (3.2.3)
Requirement already satisfied: pytz in /Users/alexandroskanterakis/anaconda3/envs/new_project/lib/python3.7/site-packages (from django) (2019.3)
Requirement already satisfied: sqlparse>=0.2.2 in /Users/alexandroskanterakis/anaconda3/envs/new_project/lib/python3.7/site-packages (from django) (0.3.0)
```

Επιβαβαιώνουμε ότι έχει εγκατασταθεί σωστά:

```
In [34]: !python -m django --version  
3.0.2
```

Ας φτιάξουμε ένα νέα site!

Το site μας θα λέγεται my\_fabulous\_site. Αρχικά φτιάχνουμε την αρχική δομή του project:

```
In [19]: !django-admin startproject my_fabulous_site
```

Ένα site φτιαγμένο σε django έχει πολλά applications. Κάθε application αναπαριστάει ένα δομικό στοιχείο του site και έχει τη δική του βάση δεδομένων. Ας φτιάξουμε ένα application με όνομα my\_fabulous\_app:

```
In [20]: %%bash  
cd my_fabulous_site  
python manage.py startapp my_fabulous_app
```

Το αρχείο my\_fabulous\_site/settings.py περιέχει τις ρυθμίσεις όλου του site

```
In [8]: !cat my_fabulous_site/my_fabulous_site/settings.py
```

```
"""
Django settings for my_fabulous_site project.

Generated by 'django-admin startproject' using Django 3.0.2.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '!c1^)^8k+pp)e)4+==p&29v5v=dao*!)z!@d4s!m-jn@drsw_tj'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'my_fabulous_site.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Στο κομμάτι

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

περιέχει όλα τα applications που έχει το site. Ας προσθέσουμε και το my\_fabulous\_app:

```
In [21]: %%writefile my_fabulous_site/my_fabulous_site/settings.py

"""
Django settings for my_fabulous_site project.

Generated by 'django-admin startproject' using Django 3.0.2.

For more information on this file, see
https://docs.djangoproject.com/en/3.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.0/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '!c1^)8k+pp)e)4+==p&29v5v=dao*!)z!@d4s!m-jn@drsw_tj'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'my_fabulous_app',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'my_fabulous_site.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ]
        }
    }
]
```

Overwriting my\_fabulous\_site/my\_fabulous\_site/settings.py

Πριν συνεχίσουμε θα πρέπει να αρχικοποιήσουμε τη βάση δεδομένων του site:

```
In [39]: %%bash
cd my_fabulous_site
python manage.py migrate

Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
```

Στο directory my\_fabulous\_site/my\_fabulous\_app/templates/my\_fabulous\_app Βάζουμε τις html σελίδες του site. Για αρχή θα δημιουργήσουμε αυτόν τον κατάλογο:

```
In [22]: !mkdir -p my_fabulous_site/my_fabulous_app/templates/my_fabulous_app
```

Ας φτιάξουμε και μία html σελίδα την οποία θα ονομάσουμε first.html

```
In [23]: %%writefile my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/first.html
<html>
  <head>
    <title>My first page!</title>
  </head>
  <body>
    <h1>Welcome!</h1>
  </body>
</html>
```

Writing my\_fabulous\_site/my\_fabulous\_app/templates/my\_fabulous\_app/first.html

Τώρα πρέπει να πούμε ότι όταν μπαίνουμε στο site μας στο url /fabulous\_app , τότε θα πρέπει να μπαίνουμε στο application με το όνομα my\_fabulous\_app. Για να το κάνουμε αυτό πρέπει να κάνουμε edit το αρχείο my\_fabulous\_site/my\_fabulous\_site/urls.py :



```
In [13]: !cat my_fabulous_site/my_fabulous_site/urls.py

"""my_fabulous_site URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

```
In [38]: %%writefile my_fabulous_site/my_fabulous_site/urls.py

"""my_fabulous_site URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('fabulous_app/', include('my_fabulous_app.urls')),
]
```

Overwriting my\_fabulous\_site/my\_fabulous\_site/urls.py

Τώρα θα πρέπει να πούμε ότι όταν μπαίνουμε στο application my\_fabulous\_app στο URL first\_try (Δηλαδή: /fabulous\_app/first\_try) , θα πρέπει να τρέχει ο κώδικας που υπάρχει στο αρχείο views.py και συγκεκριμένα να τρέχει τη συνάρτηση first\_view:

```
In [25]: %%writefile my_fabulous_site/my_fabulous_app/urls.py

from django.urls import path, re_path, include

from . import views

urlpatterns = [
    path('first_try/', views.first_view),
]

Writing my_fabulous_site/my_fabulous_app/urls.py
```

Τώρα θα πρέπει να υλοποιήσουμε τη συνάρτηση `first_view` στο αρχείο `views.py`:

```
In [30]: !pwd

/Users/alexandroskanterakis/del

In [41]: %%writefile my_fabulous_site/my_fabulous_app/views.py

from django.shortcuts import render

# Create your views here.
def first_view(request):

    context = {}
    return render(request, 'my_fabulous_app/first.html', context)

Overwriting my_fabulous_site/my_fabulous_app/views.py
```

Προς το παρών αυτή η συνάρτηση δεν κάνει τίποτα σπουδαίο. Απλά λέει ότι το view `first_view` θα επιστρέφει τη σελίδα `first.html`

Ας το τρέξουμε!

Από ένα command line (έξω από το jupyter) τρέχουμε:

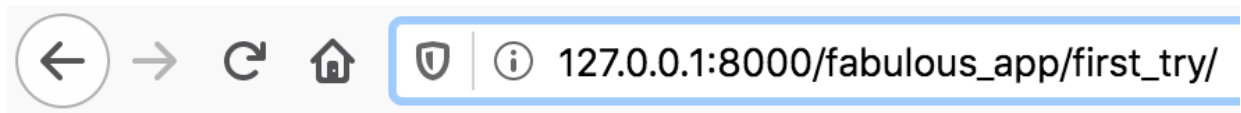
```
cd my_fabulous_site
python manage.py runserver
```

Θα πρέπει να δούμε ένα μήνυμα όπως:

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 28, 2020 - 12:30:36
Django version 3.0.2, using settings 'my_fabulous_site.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Τώρα μπορούμε να πάμε με τον browser μας στο: [http://127.0.0.1:8000/fabulous\\_app/first\\_try/](http://127.0.0.1:8000/fabulous_app/first_try/) ([http://127.0.0.1:8000/fabulous\\_app/first\\_try/](http://127.0.0.1:8000/fabulous_app/first_try/)) και να δούμε το site μας να "τρέχει":



# Welcome!

Ωραία. Ας κάνουμε το σάιτ να δείχνει την ώρα:

```
In [43]: %%writefile my_fabulous_site/my_fabulous_app/views.py

from django.shortcuts import render

import datetime

# Create your views here.
def first_view(request):

    context = {
        'now': str(datetime.datetime.now()),
    }
    return render(request, 'my_fabulous_app/first.html', context)
```

Overwriting `my_fabulous_site/my_fabulous_app/views.py`

```
In [44]: %%writefile my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/first.html
<html>
  <head>
    <title>My first page!</title>
  </head>
  <body>
    <h1>Welcome!</h1>
    <h1>Time now is: {{now}} </ht>
  </body>
</html>
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/templates/my\_fabulous\_app/first.html



# Welcome!

## Time now is: 2020-01-28 12:41:35.554121

Ας προσθέσουμε στη βάση δεδομένων έναν πίνακα ο οποίος αναπαριστάει ανθρώπους με όνομα, επώνυμο και ηλικία:

```
In [46]: %%writefile my_fabulous_site/my_fabulous_app/models.py

from django.db import models

# Create your models here.

class Person(models.Model):
    name = models.CharField(max_length=100)
    second_name = models.CharField(max_length=100)
    age = models.IntegerField()
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/models.py

Κάθε φορά που κάνουμε κάποια αλλαγή στο [σχήμα της βάσης \(https://en.wikipedia.org/wiki/Database\\_schema\)](https://en.wikipedia.org/wiki/Database_schema) πρέπει να τρέχουμε:

```
In [49]: %%bash
cd my_fabulous_site/
python manage.py makemigrations
python manage.py migrate
```

```
Migrations for 'my_fabulous_app':
  my_fabulous_app/migrations/0001_initial.py
    - Create model Person
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, my_fabulous_app, sessions
Running migrations:
  Applying my_fabulous_app.0001_initial... OK
```

Ας φτιάξουμε μία φόρμα όπου ο χρήστης θα μπορεί να εισάγει στοιχεία:

```
In [56]: %%writefile my_fabulous_site/my_fabulous_app/forms.py

from django import forms

class PeopleForm(forms.Form):
    name = forms.CharField(label='Your name', max_length=100)
    second_name = forms.CharField(label='Your surname', max_length=100)
    age = forms.IntegerField(label='Your age')

Overwriting my_fabulous_site/my_fabulous_app/forms.py
```

Πρέπει να φτιάξουμε και την αντίστοιχη html σελίδα:

```
In [68]: %%writefile my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/second.html
<html>
<head>
  <title>My first page!</title>
</head>
<body>
  <form action="/fabulous_app/insert_name/" method="post">
    {% csrf_token %}
    <table>
      {{form}}
    </table>
    <input type="submit" value="Submit">
  </form>
</body>
</html>

Overwriting my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/second.html
```

[Τι είναι αυτό το csrf\\_token; \(https://docs.djangoproject.com/en/3.0/ref/csrf/\)](https://docs.djangoproject.com/en/3.0/ref/csrf/)

Πρέπει να προσφέρουμε και ένα view για αυτό:

```
In [55]: %%writefile my_fabulous_site/my_fabulous_app/views.py

from django.shortcuts import render

from .forms import PeopleForm

import datetime

# Create your views here.
def first_view(request):

    context = {
        'now': str(datetime.datetime.now()),
    }
    return render(request, 'my_fabulous_app/first.html', context)

def new_people_view(request):
    context = {
        'form' : PeopleForm(),
    }
    return render(request, 'my_fabulous_app/second.html', context)

Overwriting my_fabulous_site/my_fabulous_app/views.py
```

Και επίσης μία καινούργια εγγραφή στα URLs:

```
In [58]: %%writefile my_fabulous_site/my_fabulous_app/urls.py

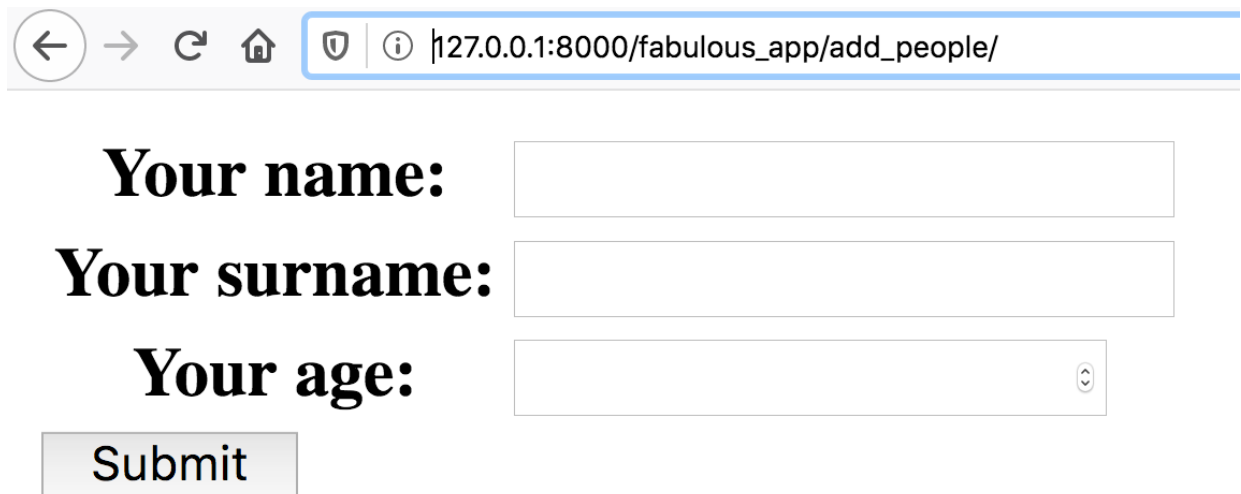
from django.urls import path, include

from . import views

urlpatterns = [
    path('first_try/', views.first_view),
    path('add_people/', views.new_people_view),
]
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/urls.py

Μπορούμε να δούμε τη φόρμα τώρα εδώ: [http://127.0.0.1:8000/fabulous\\_app/add\\_people/](http://127.0.0.1:8000/fabulous_app/add_people/) ([http://127.0.0.1:8000/fabulous\\_app/add\\_people/](http://127.0.0.1:8000/fabulous_app/add_people/))



← → ↻ 🏠 🔒 ⓘ http://127.0.0.1:8000/fabulous\_app/add\_people/

**Your name:**

**Your surname:**

**Your age:**

Όταν πατήσουμε το submit, σε αυτή τη σελίδα θα μας "πάει" στη σελίδα "insert\_name/" οπότε πρέπει να μπει άλλο ένα view για αυτό. Επίσης σε αυτό το view θα γίνεται και η καταχώρηση των στοιχείων της φόρμας στη βάση:

```
In [74]: %%writefile my_fabulous_site/my_fabulous_app/views.py

from django.shortcuts import render

from my_fabulous_app.models import Person

from .forms import PeopleForm

import datetime

# Create your views here.
def first_view(request):

    context = {
        'now': str(datetime.datetime.now()),
    }
    return render(request, 'my_fabulous_app/first.html', context)

def new_people_view(request):
    context = {
        'form' : PeopleForm(),
    }
    return render(request, 'my_fabulous_app/second.html', context)

def insert_name_view(request):
    form = PeopleForm(request.POST)
    name = form.data['name']
    second_name = form.data['second_name']
    age = form.data['age']

    # Store in database
    new_person = Person(name=name, second_name=second_name, age=age)
    return render(request, 'my_fabulous_app/third.html', {})
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/views.py

Και επίσης πρέπει να ανανεώσουμε και το urls.py:

```
In [71]: %%writefile my_fabulous_site/my_fabulous_app/urls.py

from django.urls import path, include

from . import views

urlpatterns = [
    path('first_try/', views.first_view),
    path('add_people/', views.new_people_view),
    path('insert_name/', views.insert_name_view),
]
```

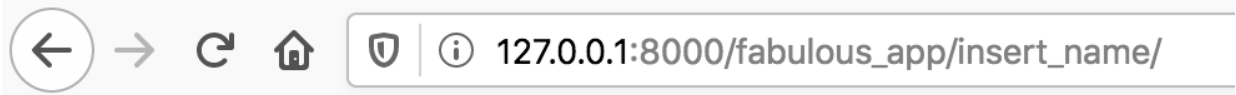
Overwriting my\_fabulous\_site/my\_fabulous\_app/urls.py

Καθώς επίσης και να φτιάξουμε τη σελίδα success.txt:

```
In [75]: %%writefile my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/third.html
<html>
  <head>
    <title>My first page!</title>
  </head>
  <body>
    <h1>Success!</h1>
  </body>
</html>
```

Writing my\_fabulous\_site/my\_fabulous\_app/templates/my\_fabulous\_app/third.html

Τώρα, πατήσουμε το submit εμφανίζεται ένα μήνυμα Success:



# Success!

Τέλος μπορούμε να ζητήσουμε να τυπώσει μία λίστα με όλα τα πρόσωπα:

```
In [80]: %%writefile my_fabulous_site/my_fabulous_app/templates/my_fabulous_app/forth.html
<html>
  <head>
    <title>My first page!</title>
  </head>
  <body>
    <h1>List with all people</h1>
    <ul>
      {% for person in person_list %}
        <li>{{ person.name }} {{person.second_name}} {{person.age}} </li>
      {% endfor %}
    </ul>
  </body>
</html>
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/templates/my\_fabulous\_app/forth.html

Παρατηρήστε ότι μπορούμε να κάνουμε iterate μέσα στη html! [Διαβάστε εδώ για περισσότερα](https://docs.djangoproject.com/en/3.0/ref/templates/language/)



```
In [83]: %%writefile my_fabulous_site/my_fabulous_app/views.py

from django.shortcuts import render

from my_fabulous_app.models import Person

from .forms import PeopleForm

import datetime

# Create your views here.
def first_view(request):

    context = {
        'now': str(datetime.datetime.now()),
    }
    return render(request, 'my_fabulous_app/first.html', context)

def new_people_view(request):
    context = {
        'form' : PeopleForm(),
    }
    return render(request, 'my_fabulous_app/second.html', context)

def insert_name_view(request):
    form = PeopleForm(request.POST)
    name = form.data['name']
    second_name = form.data['second_name']
    age = form.data['age']

    # Store in database
    new_person = Person(name=name, second_name=second_name, age=age)
    new_person.save()

    return render(request, 'my_fabulous_app/third.html', {})

def all_names_view(request):
    person_list = Person.objects.all()
    context = {
        'person_list': person_list,
    }

    print (person_list)
    return render(request, 'my_fabulous_app/forth.html', context)
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/views.py

```
In [79]: %%writefile my_fabulous_site/my_fabulous_app/urls.py

from django.urls import path, include

from . import views

urlpatterns = [
    path('first_try/', views.first_view),
    path('add_people/', views.new_people_view),
    path('insert_name/', views.insert_name_view),
    path('all_names/', views.all_names_view),
]
```

Overwriting my\_fabulous\_site/my\_fabulous\_app/urls.py

Τώρα αν προσθέσουμε μερικά άτομα και πάμε στη σελίδα: [http://127.0.0.1:8000/fabulous\\_app/all\\_names/](http://127.0.0.1:8000/fabulous_app/all_names/) ([http://127.0.0.1:8000/fabulous\\_app/all\\_names/](http://127.0.0.1:8000/fabulous_app/all_names/)) θα δούμε μία λίστα με όλα τα άτομα που προσθέσαμε:



# List with all people

- maria qwerty 105
- kostas mpigalis 60
- teris xrisos 30

Αυτή ήταν μία πάρα πολύ συνοπτική παρουσίαση του django. Αν ενδιαφέρεστε μπορείτε να διαβάσετε περισσότερα:

- [Getting started and tutorials \(https://docs.djangoproject.com/en/3.0/intro/\)](https://docs.djangoproject.com/en/3.0/intro/)
- [Main Django concepts and topics \(https://docs.djangoproject.com/en/3.0/topics/\)](https://docs.djangoproject.com/en/3.0/topics/)
- [Official documentation \(https://docs.djangoproject.com/en/3.0/\)](https://docs.djangoproject.com/en/3.0/)