Contents

[Introduction 2](#_Toc14952266)

[Setup 2](#_Toc14952267)

[Django project 2](#_Toc14952268)

[Create polls app in mysite 2](#_Toc14952269)

[Explanation: 3](#_Toc14952270)

[Installing Django server on cloudjiffy 3](#_Toc14952271)

15-07-2019

Url’s: <https://docs.djangoproject.com/en/2.2/intro/tutorial01/>

# Introduction

Django is high level Python web framework. It also contains light weight web server.

# Setup

Setup consists of installing:

1. Python: Download and install python from windows installer
2. Django

Install through virtual env.

pip install virtualenvwrapper-win

mkvirtualenv myproject

//You will see myproject next to command prompt

1. Web server: automatically installed

Check:

python -m django --version

# Django project

Entire web application is a project. Project is collection or container of applications. An application is set of code files. Say a website has home, contacts, about, blogs. Then this home, contacts, about etc. are applications and all of them taken together is project. An application can be moved between projects. Each application is independent.

To setup a web server localhost:8000

* Create the project: python startproject mysite
* Change and start server: cd mysite; python manage.py runserver

Here you may be asked to do some migrations which you do as directed.

* Browse at localhost:8000. An admin site localhost:8000/admin is also created automatically.

## Create polls app in mysite

* python manage.py startapp polls
* In polls folder, modify views.py as

**from** **django.http** **import** HttpResponse

**def** index(request):

**return** HttpResponse("Hello, world. You're at the polls index.")

This is defining the function index which returns hello world. Now this views.py needs to be associated with urlPatterns. See, at top mysite receives the request. Its urls.py will transfer the request to polls urls.py, which will call the above view. For this urlPatterns and include is done

* Create a file urls.py in polls/urls.py

**from** **django.urls** **import** path

**from** **.** **import** views

urlpatterns = [

path('', views.index, name='index'),

]

* In top-level site mysite modify the urls.py

**from** **django.contrib** **import** admin

**from** **django.urls** **import** include, path

urlpatterns = [

path('polls/', include('polls.urls')),

path('admin/', admin.site.urls),

]

* python manage.py runserver
* browse at localhost:8000/polls

### Explanation:

Urls.py is called urlconf. Include function adds other urlconfs thereby doing the complete url match.

Path function’s arguments. There are 4 arguments

1. route: This is string which contains url patter. The patters are not concerned with http:// or https:// or any url parameters
2. view: If urlPattern is matched then corresponding view function is called. The firest argument of this function is HttpRequest object and other arguments are keyword arguments explained in a bit.
3. kwargs: Arbitrary keyword arguments can be passed as dictionary to the targetted views
4. name: This is how you name the url. Once you name the url you can call it by giving the name from anywhere in the Djanjo, especially from templates.

# Installing Django server on cloudjiffy

<https://jelastic.com/blog/django-cms-installation-python-cloud-hosting/>

This link installs Django-cms software. A custom Django server will be same except that you need to manually copy project files to ROOT folder and do some manual settings as explained in above link. I followed this process:

* **virtualenv virtenv**

virtualenv and python are preinstalled. You need to use the word ‘virtenv’ as it is, since the same word is hard code in apache server wsgi configurations.

* **source virtenv/bin/activate**

To activate virtenv. You will see the word virtenv in command prompt. Now on all installations will be done in virtenv virtual environment

* **pip install Django==2.1**

After this, You will find in wwwroot a folder virtenv. We need to install Django 2.1, because higher virson has some problem with sqlite.

* **django-admin startproject myproject**

You will see a new folder myproject with project files in it.

* **cd myproject**
* **python manage.py migrate**

This will create some files with sqlite3.db. This step is important

Now you need to port the files in myproject to ROOT folder and modify the wsgi.py and settings files.

* **zip -r myproject.zip myproject**

Recursively zip all files in folder myproject. Output file is myproject.zip. Download this file and extract in your local machine.

* From local machine upload two files to wwwroot folder which is container of ROOT folder. These files are in myproject outer folder.

manage.py

db.sqlite3

* All the project files in inner myproject folder except the cache files, zip the, upload to ROOT folder and unzip them in cloud.
* Replace wsgi.py with following

import os,sys

virtenv = os.path.expanduser('~') + '/virtenv/'

virtualenv = os.path.join(virtenv, 'bin/activate\_this.py')

try:

if sys.version.split(' ')[0].split('.')[0] == '3':

exec(compile(open(virtualenv, "rb").read(), virtualenv, 'exec'), dict(\_\_file\_\_=virtualenv))

else:

execfile(virtualenv, dict(\_\_file\_\_=virtualenv))

except IOError:

pass

sys.path.append(os.path.expanduser('~'))

sys.path.append(os.path.expanduser('~') + '/ROOT/')

os.environ['DJANGO\_SETTINGS\_MODULE'] = 'ROOT.settings'

from django.core.wsgi import get\_wsgi\_application

application = get\_wsgi\_application()

* Modify settings.py to have following lines

gettext = lambda s: s

DATA\_DIR = os.path.dirname(os.path.dirname(\_\_file\_\_))

ALLOWED\_HOSTS = ['\*']

ROOT\_URLCONF = 'ROOT.urls'

STATIC\_URL = '/static/'

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(DATA\_DIR, 'ROOT', 'media')

STATIC\_ROOT = os.path.join(DATA\_DIR, 'ROOT', 'static')

STATICFILES\_DIRS = (

os.path.join(BASE\_DIR, 'ROOT', 'static\_local'),

)

Restart the server and browse. All apache calls will be redirected to Django server.

# Virtualenv for Django

Here I explain how to create a virtualenv for Django in Windows

Virtualenv is setup which has got all installations. You can have multiple virtual environments applicable and activate as per requirement.

pip install virtualenvwrapper-win

mkvirtualenv virtenv

workon virtenv

To start virtenv in linux

source virtenv/bin/activae and to deactivate in linux

deactivate

The above code will create and activate virtual environment by name ‘virtenv’. When it is active whatever you install with pip will be installed in the virtenv. To deactivate it call deactivate()

With virtenv active:

pip install Django==2.1

django-admin createproject myproject

cd myproject

python manage.py migrate

python manage.py runserver

Above lines create myproject with folder hierarchy and starts the Django server at port 8080.

python manage.py startapp polls will create polls app.

# Cloudjiffy create an app polls in ROOT project

source virtenv/bin/activate

cd ROOT

python manage.py startapp polls

change Root/urls.py to include polls urls.py

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('polls/', include('polls.urls'))

]

create ROOT/polls/urls.py

from django.urls import path, include

from . import views

urlpatterns = [

path('', views.index, name='index' )

]

change view in ROOT/polls/views.py for output

from django.http import HttpResponse

def index(request):

return HttpResponse('This is content from views')

The above works fine in cloudjiffy

# Import statements in Python /Django

The import does not work as in node.js. Import first searches in current directory then in python path, sys.path and python build in modules. There is no way to go to peer or parent from current directory.

If you want to import from peer or parent directory you should start from root directory folder.

Say polls and utils are peers. You want to import from utils into polls. You should do something like ‘from utils import getcomments’

Here you cannot do like from ‘../utils’ import getcomments as you would do in node.js.