

1) Download VS code

Create new folder > new file with .py > it will ask to install python extension

OR Extensions in left panel> type python > select first python intellisense extension

To stop the server CTRL+C

2) Go to Terminal > pip install django

You may need to set path or Edit the EV as you did in Java

CREATE PROGRAM

1) `django-admin startproject projectnamehere`

`cd yourProjectFolderName`

`python manage.py runserver`

CTRL + C to stop the server

2) (you are in your projectFolder)

`django-admin startapp appName`

3) Go to project's setting file>add your app in
installed_apps

4) Then go to project> urls file > import include >
path("",include(appName.urls))

5) copy project urls file to app>urls file>from . views import
yourViewFunctionName

in urlpatterns,

path('urlName',viewFunctionName,name='viewFunctionN
ame')

python manag.py runserver

STEPS:

1)create project using command

2)create app using command

3) register app in project>urls>installed_apps

4)include url to your app , go to project>urls>include url to
your app

5)include url to view function, go to app>urls>include url to
view function

6)run server

To have more urls/view functions in one view.py file, just update app>urls like

```
path('urlName',viewFunction,name='viewFunction')
```

DATABASE:

- 1) go to Terminal>pip install mysqlclient
- 2) update project>settings.py, change your database connection details
- 3) update app>models.py > from django.db import models
- 4) create class <modelName> with class Meta:
- 5) Now in terminal, go to your project folder
- 6) go to terminal>python manage.py makemigrations appName
- 7) go to terminal>python manage.py migrate appName

MODEL in Django

A model is a class that represents table in our DB, and where every attribute of the class is a field of the table. Models are defined in the app/models.py

Creating a Model

Following is a Dreamreal model created as an example –

```
from django.db import models

class Dreamreal(models.Model):

    website = models.CharField(max_length = 50)
    mail = models.CharField(max_length = 50)
    name = models.CharField(max_length = 50)
```

```
phonenumber = models.IntegerField()

class Meta:
    db_table = "dreamreal"
```

Every model inherits from `django.db.models.Model`.

Our class has 4 attributes (3 CharField and 1 Integer), those will be the table fields.

The Meta class with the `db_table` attribute lets us define the actual table name. Django names the table automatically: `myapp_modelName`. This class will let you force the name of the table to what you like.

Manipulating Data (CRUD)

Let's create a "crudops" view to see how we can do CRUD operations on models. Our `myapp/views.py` will then look like –

`myapp/views.py`

```
from myapp.models import Dreamreal
from django.http import HttpResponse

def crudops(request):
    #Creating an entry or Row

    dreamreal = Dreamreal(
        website = "www.polo.com", mail =
"sorex@polo.com",
        name = "sorex", phonenumber = "002376970"
    )

    dreamreal.save()

    #Read ALL entries/ rows
    objects = Dreamreal.objects.all()
    #returns all rows

    res = 'Printing all Dreamreal entries in the DB :
<br>'

    for elt in objects: #elt is one row in a table
```

```

        res += elt.name+"<br>"

#Read a specific entry or row
sorex = Dreamreal.objects.get(name = "sorex")

res += 'Printing One entry <br>'
res += sorex.name

#Delete an entry
res += '<br> Deleting an entry <br>'
sorex.delete()

#Update row
dreamreal = Dreamreal(
    website = "www.polo.com", mail =
"sorex@polo.com",
    name = "sorex", phonenumber = "002376970"
)

dreamreal.save()
res += 'Updating entry<br>'

dreamreal = Dreamreal.objects.get(name = 'sorex')
dreamreal.name = 'thierry'
dreamreal.save()

return HttpResponse(res)

```

Admin

Go to outer project folder> Run server python manage.py
 runserver > go to browser url localhost:8000/admin/

- 1) **python manage.py migrate**
- 2) **py manage.py createsuperuser**

- 3) Register your database table or Model in Admin.py,
- a) `from .models import ModelName`
 - b) App folder>admin.py>
`admin.site.register(ModelName)`
`#here modelName is not a string`

Now you can read, insert, update, delete rows in your database table from the Admin Console