# BEGIN

<https://docs.djangoproject.com/en/2.2/intro/tutorial04/>

py -m django --version

django-admin startproject mysite

py manage.py runserver

py manage.py startapp pollsNAAME

polls/views.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial01/" \l "id1" \o "Permalink to this code)**

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

**def** index(request):

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

polls/urls.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial01/" \l "id2" \o "Permalink to this code)**

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

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

urlpatterns = [

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

mysite/urls.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial01/" \l "id3" \o "Permalink to this code)**

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

urlpatterns = [

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

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

py manage.py runserver

////////////////////////////////////////////////////////////////////////////////////

part 2

settings.py -- set times zone

**...\>** py manage.py migrate

**install django**

if error

 (   WARNING: The script sqlformat.exe is installed in

    'C:\Users\py\_dev\AppData\Roaming\Python\Python37\Scripts' which is not on

    PATH.):

    This works on Windows

* 1. On Windows, with Python 2.7 go to the Python setup folder.
  2. Open Lib/site-packages.
  3. Add an example.pth empty file to this folder.
  4. Add the required path to the file, one per each line.

            Then you'll be able to see all modules within those paths from your scripts.

after installing django

py -m django --version

While you’re editing **mysite/settings.py**, set [**TIME\_ZONE**](https://docs.djangoproject.com/en/2.2/ref/settings/#std:setting-TIME_ZONE) to your time zone.

Also, note the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/2.2/ref/settings/#std:setting-INSTALLED_APPS) setting at the top of the file. That holds the names of all Django applications that are activated in this Django instance. Apps can be used in multiple projects, and you can package and distribute them for use by others in their projects.

By default, [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/2.2/ref/settings/#std:setting-INSTALLED_APPS) contains the following apps, all of which come with Django:

* [**django.contrib.admin**](https://docs.djangoproject.com/en/2.2/ref/contrib/admin/#module-django.contrib.admin) – The admin site. You’ll use it shortly.
* [**django.contrib.auth**](https://docs.djangoproject.com/en/2.2/topics/auth/#module-django.contrib.auth) – An authentication system.
* [**django.contrib.contenttypes**](https://docs.djangoproject.com/en/2.2/ref/contrib/contenttypes/#module-django.contrib.contenttypes) – A framework for content types.
* [**django.contrib.sessions**](https://docs.djangoproject.com/en/2.2/topics/http/sessions/#module-django.contrib.sessions) – A session framework.
* [**django.contrib.messages**](https://docs.djangoproject.com/en/2.2/ref/contrib/messages/#module-django.contrib.messages) – A messaging framework.
* [**django.contrib.staticfiles**](https://docs.djangoproject.com/en/2.2/ref/contrib/staticfiles/#module-django.contrib.staticfiles) – A framework for managing static files.

These applications are included by default as a convenience for the common case.

Some of these applications make use of at least one database table, though, so we need to create the tables in the database before we can use them. To do that, run the following command:

**...\>** py manage.py migrate

after this we have to add models

питон сам створить базу даних.. тре дати тыльки модель прописати в MODELS.PY

polls/models.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial02/" \l "id2" \o "Permalink to this code)**

**from** **django.db** **import** models

**class** **Question**(models.Model):

question\_text = models.CharField(max\_length=200)

pub\_date = models.DateTimeField('date published')

**class** **Choice**(models.Model):

question = models.ForeignKey(Question, on\_delete=models.CASCADE)

choice\_text = models.CharField(max\_length=200)

votes = models.IntegerField(default=0)

To include the app in our project, we need to add a reference to its configuration class in the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/2.2/ref/settings/#std:setting-INSTALLED_APPS) setting. The **PollsConfig** class is in the **polls/apps.py** file, so its dotted path is **'polls.apps.PollsConfig'**. Edit the **mysite/settings.py** file and add that dotted path to the [**INSTALLED\_APPS**](https://docs.djangoproject.com/en/2.2/ref/settings/#std:setting-INSTALLED_APPS) setting. It’ll look like this:

INSTALLED\_APPS = [

'certification82.apps.Certification82Config',

    'polls.apps.PollsConfig',

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',]

make migrations

py manage.py makemigrations polls

by me it is :

py manage.py makemigrations certification82

<https://stackoverflow.com/questions/35484263/no-module-named-polls-apps-pollsconfigdjango-django-project-tutorial-2>

by py 3.7 and dj 2 2 5 we use

in settings.py of the project datebank

*'certification82.apps.Certification82Config',*

in the installed apps[]

in the apps.py we have

*from django.apps import AppConfig*

*class Certification82Config(AppConfig):*

*name = 'certification82'*

now we need to sqlmigrate

py manage.py sqlmigrate polls 0001

by me it is

py manage.py sqlmigrate certification82 0001

py manage.py sqlmigrate certification82 0002

py manage.py sqlmigrate certification82 0003

because of changes

the question and choice classes had fields like

question\_text and choice\_text so it was

Now, run [**migrate**](https://docs.djangoproject.com/en/2.2/ref/django-admin/#django-admin-migrate) again to create those model tables in your database

cmd:

py manage.py migrate

* Change your models (in **models.py**).
* Run [**python manage.py makemigrations**](https://docs.djangoproject.com/en/2.2/ref/django-admin/#django-admin-makemigrations) to create migrations for those changes
* Run [**python manage.py migrate**](https://docs.djangoproject.com/en/2.2/ref/django-admin/#django-admin-migrate) to apply those changes to the database.

Wait a minute. **<Question: Question object (1)>** isn’t a helpful representation of this object. Let’s fix that by editing the **Question** model (in the **polls/models.py** file) and adding a [**\_\_str\_\_()**](https://docs.djangoproject.com/en/2.2/ref/models/instances/#django.db.models.Model.__str__) method to both **Question** and **Choice**:

polls/models.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial02/" \l "id4" \o "Permalink to this code)**

**from** **django.db** **import** models

**class** **Question**(models.Model):

*# ...*

**def** \_\_str\_\_(self):

**return** self.question\_text

**class** **Choice**(models.Model):

*# ...*

**def** \_\_str\_\_(self):

**return** self.choice\_text

**NOW Creating an admin user**[**¶**](https://docs.djangoproject.com/en/2.2/intro/tutorial02/#creating-an-admin-user)

First we’ll need to create a user who can login to the admin site. Run the following command:

/ 

**...\>** py manage.py createsuperuser

Enter your desired username and press enter.

Username: admin

You will then be prompted for your desired email address:

Email address: [admin@example.com](mailto:admin@example.com)

The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

Password: \*\*\*\*\*\*\*\*\*\*

Password (again): \*\*\*\*\*\*\*\*\*

Superuser created successfully.

add wimpty before cmd command in win 10

login myloginasadmin20193009

mypassisgood2019@@@

**py manage.py runserver**

volodia

@@@2019vova2019@@

# PART 2 like

### Make the cert app modifiable in the admin[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial02/" \l "make-the-poll-app-modifiable-in-the-admin" \o "Permalink to this headline)

But where’s our poll app? It’s not displayed on the admin index page.

Just one thing to do: we need to tell the admin that **Question** objects have an admin interface. To do this, open the **polls/admin.py** file, and edit it to look like this:

*cert means certification82*

**cert**/admin.py**[¶](https://docs.djangoproject.com/en/2.2/intro/tutorial02/" \l "id6" \o "Permalink to this code)**

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

**from** **.models** **import** Cert

admin.site.register(Cert)