Django

Database management

- mysite/settings.py - normal python module. PostgreSQL - When we are using real world database than we need to install this to manage the database.
- python manage.py migrate -

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)

mysite/settings.py

INSTALLED_APPS = [

'polls.apps.PollsConfig',

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

python manage.py makemigrations polls - this tells django that you had make some changes in your model and changes are stored as migration.

It is the file polls/migrations/001_initial.py

sqlmigrate - it takes migration names and returns their SQL.

python manage.py sqlmigrate polls 0001

python manage.py shell

from polls.models import Choice, Question # Import the model classes we just wrote.

Question.objects.all()

Create a new Question.

Support for time zones is enabled in the default settings file, so

Django expects a datetime with tzinfo for pub_date. Use timezone.now()

instead of datetime.datetime.now() and it will do the right thing.

>>> from django.utils import timezone

>>> q = Question(question_text="What's new?", pub_date=timezone.now())

Save the object into the database. You have to call save() explicitly.

>>> q.save()

Now it has an ID.

>>> q.id

1

Access model field values via Python attributes.

>>> q.question_text

"What's new?"

>>> q.pub_date

datetime.datetime(2012, 2, 26, 13, 0, 0, 775217, tzinfo=<UTC>)

Change values by changing the attributes, then calling save().

>>> q.question_text = "What's up?"

>>> q.save()

objects.all() displays all the questions in the database.

>>> Question.objects.all()

<QuerySet [<Question: Question object (1)>]>

polls/models.py

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

🏁 Virtual environment

🚢 Django installation

🖱️ Django project creation

🚇 Admin

🔦 Creating apps

💈 Practise Problem

▼ **Setting.py**

import os #django works on os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(_file_))) # path where manage.py is present

print(BASE_DIR)

SECRET_KEY #change it to make in avoid public interference

DEBUG = True #true for developer and switch to false for other user

INSTALLED_APPS #products, list of products or other things, like components.

MIDDLEWARE

ROOT_URLCONF #to rout given url

TEMPELATES

WSGI_APPLICATION #setting

DATABASES #map to database, generally backend, normally sqlite

STATIC_URL #where to store the files

python manage.py migreate #create database things

First custom app

!root of jango is at manage.py

Create app

open models.py from product

▼ Open settings.py

INSTALLED_APPS = [

'products',]

python manage.py makemigrations

python manage.py migrate

go to admin.py

from django.contrib import admin

from .models import product

admin.site.register(product)

Create python objects

create python shell and add objects

adding new fields

Page

from django.contribute import admin

from django.url import path

from pages import views

urlpatterns = [

path('',,

```
pwd is used to find path
```