

CSE-465

Web Programming

URL Dispatcher in Django

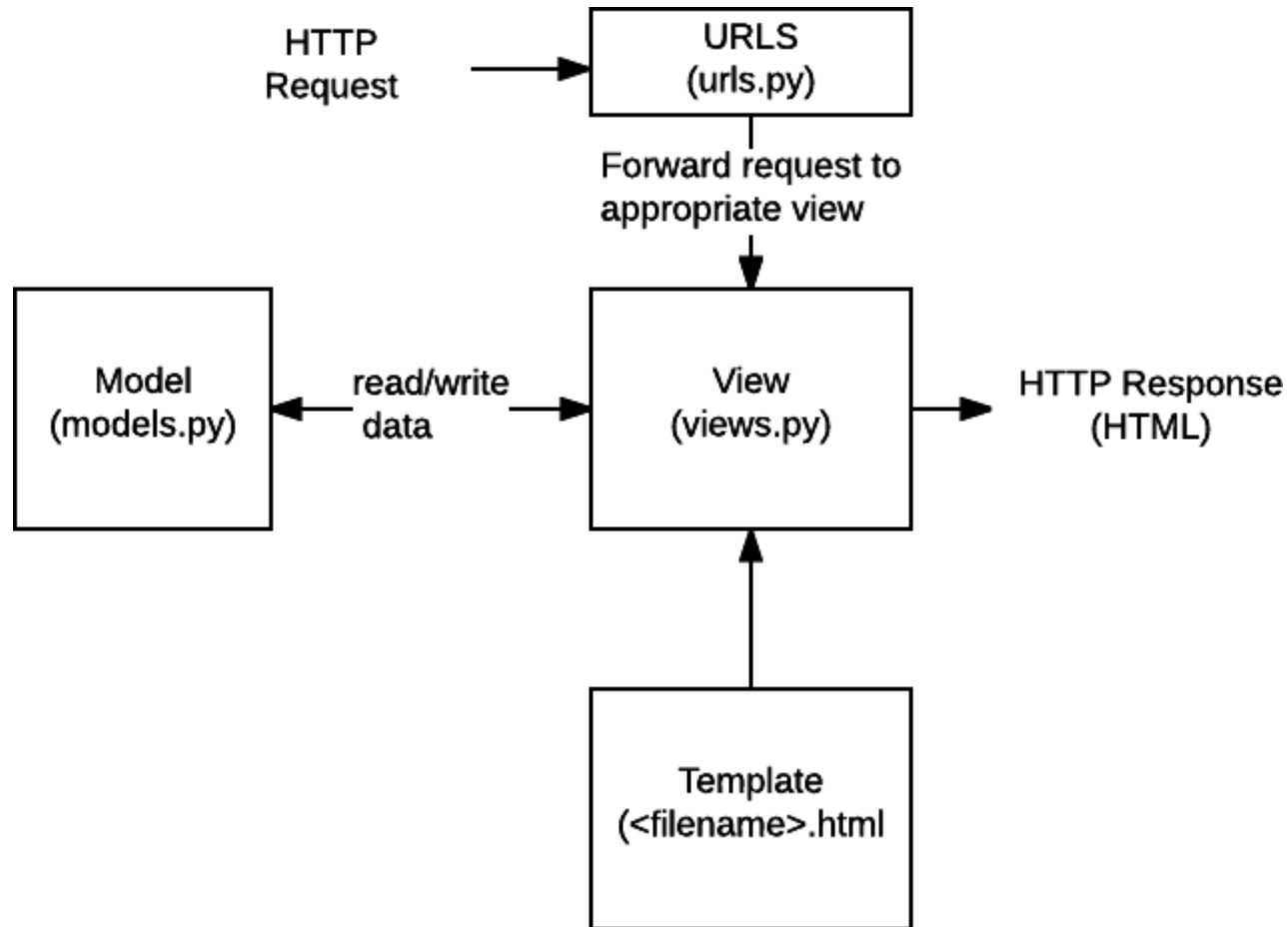
Python Setup

- Download Python
<https://www.python.org/downloads/>
- To verify if Python is installed, open command prompt and type-
`python -version`
- To create and activate virtual environment
`python -m venv env`
`env\Scripts\activate`
- Install Django
Open Command prompt, and type the following code-
`pip install Django`
- To verify if Django is installed, open command prompt and type-
`python -m django --version`

Setting up Django

- To create a new django project:
`django-admin startproject project_name`
- To run the django server:
`python manage.py runserver`
- To create another app inside the project:
`python manage.py startapp app_name`

Workflow of Django



Key concepts of Python

Learn the following key concepts from here:

<https://www.programiz.com/python-programming/tutorial>

- Variables Declaration
- For loop
- List
- Dictionary
- Functions, Returning from a function
- Classes
- Importing external files

Configuring the App

You need to register your app inside the INSTALLED_APPS before you can use

the features of Django.

1. Create a file called apps.py and put the following code there-

```
from django.apps import AppConfig  
class <Any Valid ClassName>(AppConfig):  
    name = '<YOUR_APP_NAME>'
```

[The name of a class usually starts with uppercase]

2. Add the class-name with location in the INSTALLED_APPS in settings.py

```
INSTALLED_APPS = [  
    'facebook2.apps.Facebook2Config',  
    'django.contrib.admin'
```

views.py

```
from django.shortcuts import render
```

```
def home(request):  
    return render(request, 'index.html')
```

urls.py

```
from . import views
```

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


Task

1. Create a URL and View for an profile page that will return your name and your id
2. Create a URL and View for an about page that will return your course name and course id

Example

```
from . import views

urlpatterns = [
    path('articles/2003/', views.special_case_2003),
    path('articles/<int:year>', views.year_archive),
    path('articles/<int:year>/<int:month>', views.month_archive),
    path('articles/<int:year>/<int:month>/<slug:slug>', views.article_detail),
]
```

Notes:

- **To capture a value from the URL, use angle brackets.**
- Captured values can optionally include a converter type. For example, use `<int:name>` to capture an integer parameter. If a converter isn't included, any string, excluding a `/` character, is matched.
- There's no need to add a leading slash, because every URL has that. For example, it's `articles`, not `/articles`.

Dynamically rendering HTML

1. Have the content as a python dictionary in views.py
2. pass it as third argument in the render function

```
context = { 'username' : 'Tom' }  
return render(request, 'blog/home.html', context)
```

Blog Example

```
blog > view.py
```

```
from django.shortcuts import render
```

```
db = [  
    {  
        'author': 'Peter',  
        'title': 'Blog Post 1',  
        'content': 'Content 1',  
        'date_posted': '15 April 2021'  
    },  
    {  
        'author': 'Tom',  
        'title': 'Blog Post 2',  
        'content': 'Content 2',  
        'date_posted': '16 April 2021'  
    }  
]  
  
def home(request):  
    context = {  
        'posts': db  
    }  
    return render(request, 'blog/home.html', context)
```

Referencing a template in HTML

blog > templates > blog > home.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Blog - Homepage</title>
</head>
<body>
    <h1>Welcome to our blog</h1>
    <br>
    {%for post in posts %}
    <h2>{{ post.title }}</h2>
    <h4>{{ post.author }}, {{ post.date_posted }}</h4>
    <p>{{ post.content }} </p>
    <br>
    {%endfor %}
</body>
</html>
```

Django Template Language

- `{{ }}` to access a variable
- `{% for ... %} ... {% endfor %}` : for iteration
- `{% if ... %} ... {% endif %}` : for conditionals
- `{% if ... %} ... {% else %} ... {% endif %}` : for conditionals
- Read more:
<https://docs.djangoproject.com/en/3.1/ref/templates/language/>

Another Example

```
blog > view.py
```

```
...
```

```
def about(request):
```

```
    return render(request, 'blog/about.html', {'title' :  
'CSE-456 Project About page'})
```

Task

- Generate a view for profile page and send your Name and ID to it from the view function

Usage of if-else

blog > templates > blog > about.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    {%if title %}
```

```
    <title>Blog Project - {{ title }}</title>
```

```
    {%else %}
```

```
    <title>Blog Project</title>
```

```
    {%endif %}
```

```
</head>
```

```
<body>
```

```
    <h1>Welcome to our blog</h1>
```

```
    <h2>CSE-456 sample django project</h2>
```

```
</body>
```

```
</html>
```

Referring to a link

Instead of hard-coding a link, we should use `{% url '<name>' %}`

blog > urls.py

```
...
urlpatterns = [
    path('', views.home, name='blog-home'),
    path('about/', views.about, name='blog-about'),
]
...
```

blog > templates > blog > base.html

```
...
<a href="{% url 'blog-home' %}">Home</a>
<a href="{% url 'blog-about' %}">About</a>
...
```

Linking Static Files

To link to a css/js file:

Create the path: blog > static > blog, and put the files (eg. main.css) here.

- To link main.css to these files in an HTML:

- At the top of the file, write this:

```
{% load static %}
```

- In href, write:

```
{% static 'blog/main.css' %}
```

- To hard refresh, use Ctrl + F5 or Restart the server

Task

1. Create a URL and View that will receive any of the four operations (add/subtract/multiply/divide) and two operands (both integer) and show a page with the result.
2. Create a URL and View that will receive a date in dd/mm/yyyy format and show a page with the name of the month (January/February/...)

Reference

1. <https://docs.djangoproject.com/en/3.2/topics/http/urls/>
2. Python Django Tutorial (Corey Schafer):
<https://www.youtube.com/playlist?list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p>