Introduction

Installation

Settings Module

Requests and Response

Running development server

Django admin site introduction

Installation

- Python Installation
- Django Installation

```
pip install django
```

Settings Module

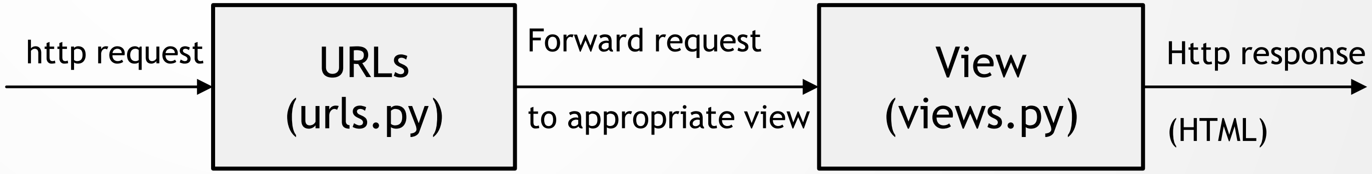
- settings.py – why these variables and values are used

Separating dev, prod, test environments

- Create a folder called settings or config
- Move the settings.py to that dir
- Create a __init__.py
- Create a dev.py, import settings to it, overwrite the values in dev.py
- Repeat the same for prod.py
- In settings.py, point the base dir one step up
- In wsgi.py and manage.py, point the settings to dev
- Run the server

Requests & Response

- When a url is striked in the web browser, ***urls.py*** is involved to map the path to a view (views.py)
- ***views.py*** is responsible to send a response HTML



Requests & Response

urls.py

```
from .views import index

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

views.py

```
from django.http import HttpResponse

def index(request):
    return HttpResponse('<b>Hello world<b>')
```

Running development server

- Make a new project directory
- Start the project

```
django-admin startproject mysite
```

- Run the server

```
python manage.py runserver
```

Recap

- Settings module – separating dev & prod environments
- Requests & Response
- Running a development server
- Template overview with context

Today

- Django admin site introduction
- Templates built in tags and filters

Django admin site introduction

- Apply / enable the default app(s) by migrate

```
python manage.py migrate
```

- Create a super user

```
python manage.py createsuperuser
```

- Run the server and play around admin site

```
python manage.py runserver
```

Templates Built-in tags and filters

- For
- If

- Block

{% block content %}

{% endblock %}

- Extends

{% extends ‘backend/base.html’ %}

Templates Built-in tags and filters

- Add
- Capfirst

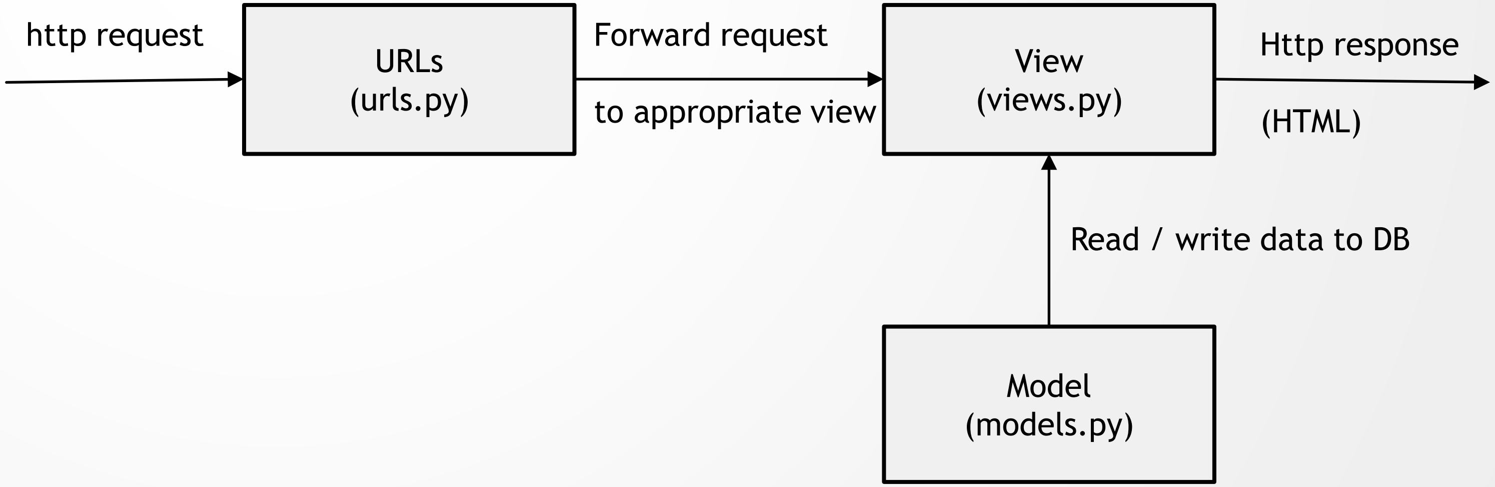
Additional Reference - official doc

Models Layer

Model Introduction

Models Introduction

- Sqlite3 download
- View all the tables in sqlite3 db



Models Introduction

- Start an app

```
python manage.py startapp members
```

- Edit members/models.py (as per next slide)
- Edit gymsite/settings.py (as per next slide)
- Makemigrations

```
python manage.py makemigrations
```

- Migrate

```
python manage.py migrate
```

Models Introduction

models.py

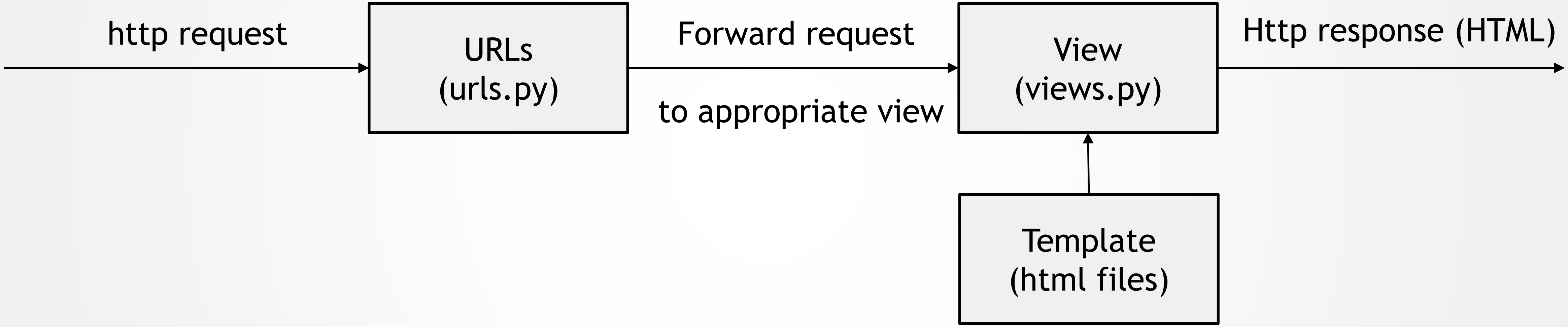
```
from django.db import models

# members records
class members(models.Model):
    name = models.CharField(max_length=30)
```

settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'members'
]
```

Template Layer

xxx



http request

URLs
(urls.py)

Forward request

to appropriate view

View
(views.py)

Http response (HTML)

Template
(html files)

Template - Overview

- Start an app

```
>python manage.py startapp members
```

- Add app to Installed Apps

In project directory, in *settings.py*, add the newly created members

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'members'
]
```

- In app directory, create a folder called **templates** → backend → index.html

- Edit index.html

index.html

```
<html >

<style>
body {
   background-color:
lightblue;
}
</style>
<body>
    hey
</body>
```

Template - Overview

- Edit members/views.py

- Add path to to project’s urls.py

- Makemigrations

```
> practice\gymsite\gymsite>python manage.py makemigrations
```

- Migrate

```
> practice\gymsite\gymsite>python manage.py migrate
```

- Runserver

```
> practice\gymsite\gymsite>python manage.py runserver
```

views.py

```
def url1(request):
    return render(request, 'backends/blog.html')
```

urls.py

```
from members.views import page

urlpatterns = [
    path('admin/', admin.site.urls),
    path('members/', page)
]
```

Template- Overview
Context

- In the app directory, edit views.py
- In the app directory, edit templates → backend → index.html
- Run the server

index.html

```
<html >

<style>
body {
   background-color: lightblue;
}
</style>
<body>
    hey {{name}}
</body>
```

views.py

```

from django.shortcuts import render

# Create your views here.

context = {'name': 'hephzi'}

def url1(request):
    return render(request, 'backend/index.html', context)
```

Template- Overview

Template files with little complicated context

- In the app directory, edit views.py

- In the app directory, edit templates → backend → index.html
- Run the server

index.html

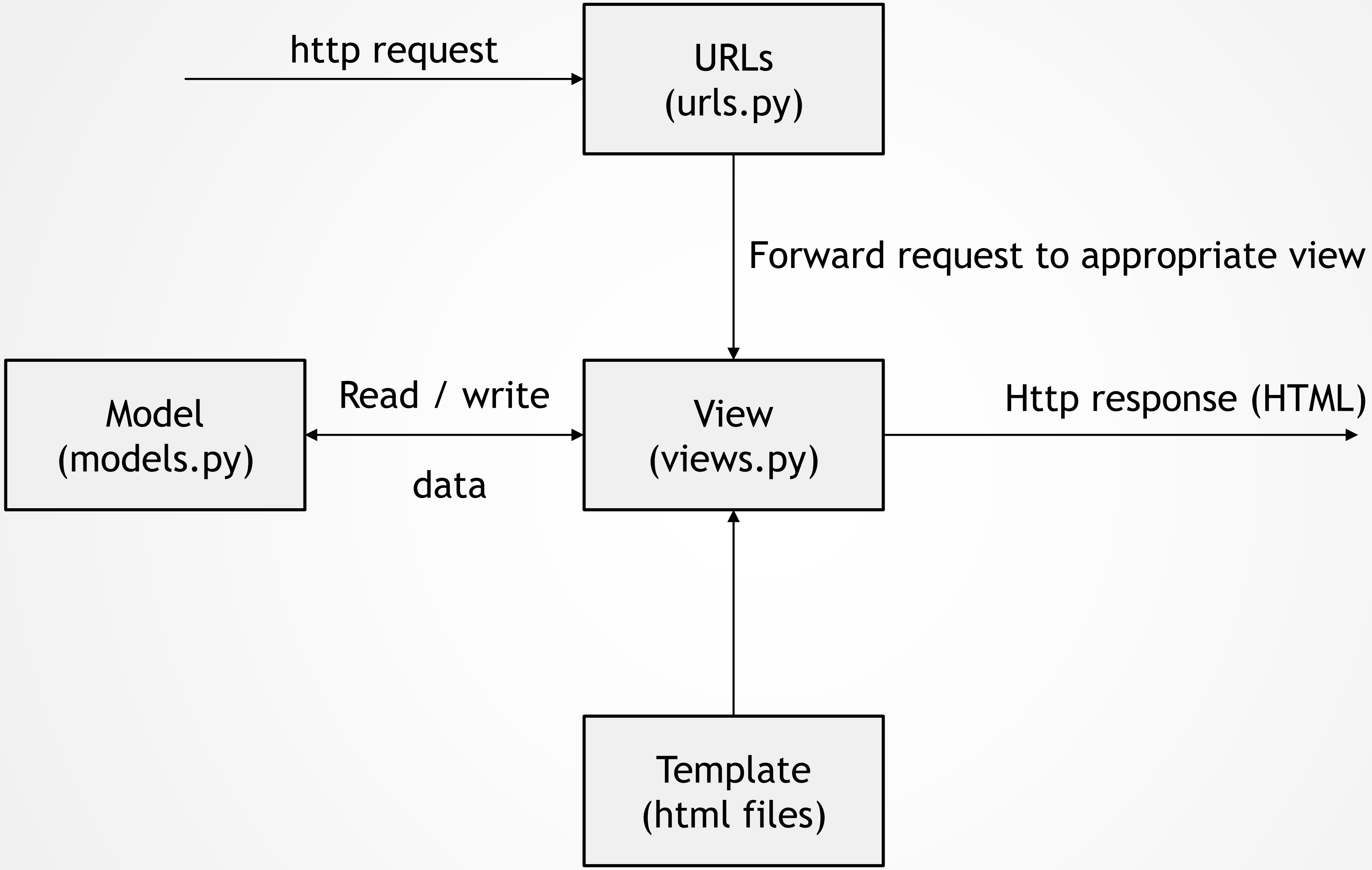
```
<body>
<table>
    <tr>
    <th> Name </th><th> Active </th>
    </tr>

    {% for member in members %}
    <tr>
    <td> {{ member.name }} </td>
    <td> {{ member.active }} </td>
    </tr>
    {% endfor %}
</table>
</body>
```

views.py

```
from django.shortcuts import render
# Create your views here.

context = {'members': [
    {'name': 'jessie', 'active': True},
    {'name': 'malini', 'active': False}
]}

def url1(request):
    return render(request,'backend/index.html',
context)
```

Templates

- {{ request.user }}
- {{ request.user.is_authenticated }}
- Extends
- Block
- For
- if


http request
URLs
(urls.py)
Forward request to appropriate view
Model
(models.py)
Read / write
data
View
(views.py)
Http response (HTML)
Template
(html files)