一、Django 概览

MVT 设计模式（与 MVC 差不多）MTV 是 Model、Template、View 三个单词的简写

MTV 模式

名称

说明

M 模型（Model）,负责业务对象和数据库的关系映射。

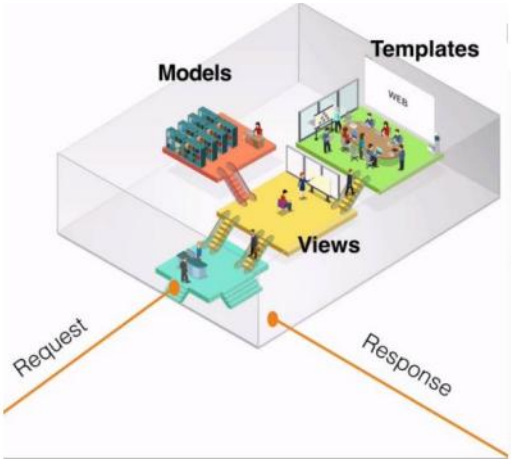
T 模板（Template）,负责如何把页面展示给用户。

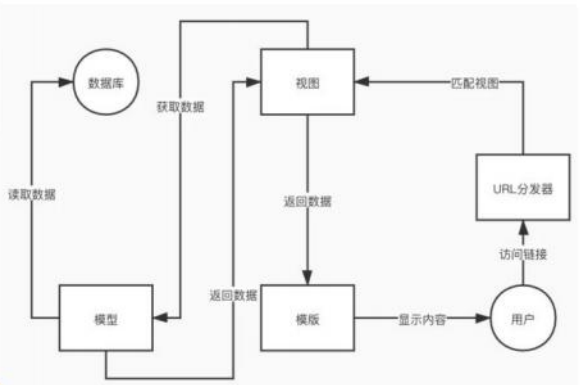
V 视图（View）,负责业务逻辑，并在适当时候调用模型和模板。

+

URL 分发器

URL 分发器相当于一个路由器，其作用是定义相应的路由，将页面请求匹配路由分发给不同的视图 View 处理，视图再调用相应的模型 Model 和模板 Template。





二、Django 使用

Step1 初始化项目

1.安装 django 目前最新版是 Django :

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple django
```

2.使用 django 命令创建新工程：

```
django-admin startproject webtest .
```

安装完后会发现出现新文件夹，里面有很多新文件

3.启动 django 服务

```
python manage.py runserver
```

4.创建一个新的 app

```
python manage.py startapp mywebtest
```

5.注册新 app，修改项目里的 settings.py

```
INSTALLED_APPS = [
```

```
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'mywebtest'
]
```

Step2 测试 view function 及 urlconfig

1. 编辑 products app 文件夹下的 view.py，修改成如下

```
from django.http import HttpResponse
from django.shortcuts import render
# /products - > index
# Uniform Resource Locater (Adress)
# Create your views here.
def index(request):
return HttpResponse("Hello World")
```

2. URL 映射,让 url 能指向视图函数

直接修改项目中的 urls.py

```
from django.contrib import admin
from django.urls import path
from mywebtest.views import index
urlpatterns = [
path('admin/', admin.site.urls),
path('mywebtest/', index)
]
```

```
python manage.py runserver
```

测试路径下的显示吧

Step3 上一步完成后，将 view function 返回渲染后的 html 页面

1. 在 app 文件夹下创建一个 templates 文件夹（注意别建错地方）

2.

2.在 templates 中创建一个 index.html（名字随意），内容如下进行测试

```
<h1>Products</h1>
<ul> <li>item1</li>
<li>item2</li>
<li>item3</li>
</ul>
```

3.修改视图函数返回的内容

```
def index(request):
```

```
    # return HttpResponse("Hello World")
        return render(request, 'index.html')
```

用 get 和 post 两种方式实现一个登录界面

什么是 HTTP?

超文本传输协议（HTTP）的设计目的是保证客户机与服务器之间的通信。

HTTP 的工作方式是客户机与服务器之间的请求-应答协议。

举例：客户端（浏览器）向服务器提交 HTTP 请求；服务器向客户端返回响应。响应包含关于请求的状态信息以及可能被请求的内容。

两种 HTTP 请求方法：GET 和 POST

在客户机和服务器之间进行请求-响应时，两种最常被用到的方法是：GET 和 POST。

GET - 从指定的资源请求数据。

POST - 向指定的资源提交要被处理的数据

三、登录界面实例

step1.在网上搜一个 form 标签实例。

```
<form action="form_action.asp" method="get">
<p>First name: <input type="text" name="fname" /></p>
<p>Last name: <input type="text" name="lname" /></p>
<input type="submit" value="Submit" />
</form>
```

Step2.若用 GET 方式请求登录

1. 自定义 form 中的输入框参数 Action：表单提交地址

```
<form action="/mywebtest/login/" method="get">
<p>用户名: <input type="text" name="username" /></p>
<p>密 码: <input type="password" name="pwd" /></p>
<input type="submit" value="登录" />
</form>
```

2. 在视图文件中(views.py)定义一个新的 view function,如下:

```
def login(request):
username = request.GET.get('username',' ')
pwd = request.GET.get('pwd',' ')
#判断是否登录成功
if username== 'aaa' and pwd== '111111':
return HttpResponse('登录成功')
else:
return HttpResponse('FAILED')
```

3. 配置路由,urls.py:

```
urlpatterns = [
```

```
path('admin/', admin.site.urls),
path('mywebtest/', index),
path('mywebtest/login/', login)
]
```

4. 重启服务器，测试登陆..

四、引入数据库/django /实现登录界面

创建 models

1.编辑 mywebtest app 下的 models.py，创建一个类，相当于表名，再编辑好表
中的各个字段。

```
from django.db import models

class User_info(models.Model):
    username = models.CharField(max_length= 255)
    pwd = models.CharField(max_length= 255)
```

2.观察 db.sqlite3 文件

注：在未迁移前，此数据库文件为空，可以使用 db browser for sqlite 进行观察，将 db.sqlite3 拉入程序中，没有表格显示

3.做 migration，将 model 迁移到数据库文件中

（a）使用命令 python manage.py makemigrations
做完动作（a）将新增 migrations 文件夹，初始化成功可以查看
0001_initial.py 看详细情况，将 db.sqlite3 拉入可视化中，但依然没有表格显示
（b）运行 python manage.py migrate,将 db.sqlite3 拉入程序中，表格出现

将后台 or 数据库中的数据显示到网页上

1. 修改 view.py 文件如下，修改如下，创建一个对象获取 User_info 里的 all 信息

```
from .models import User_info
def index(request):
    user_info = User_info.objects.all
    # return HttpResponse("Hello World")
    return  render(request,  'index.html',  {'user_info': user_info})
```

2.接下来修改 html 里的内容，让其能够使用上数据库里的内容

```
{% for users in user_info %}
<li>{{ users.username }} ${{ users.pwd }}</li>
{% endfor %}
```

2.如果想让登录界面的数据内容是从数据库中读取判断的，修改如下函数

```
def login(request):
    username = request.GET.get('username',' ')
    pwd = request.GET.get('pwd',' ')
    conn = sqlite3.connect("db.sqlite3")
    cur = conn.cursor()
    cur.execute('SELECT * FROM mywebtest_user_info')
    rows = cur.fetchall()
    print(rows)
    acc = []
    for row in rows:
        acc.append((row[1],row[2]))
    if (username,pwd) in acc:
        return HttpResponse('登录成功')
    else:
        return HttpResponse('FAILED')
```