Django基础

2020年9月23日 11:25

python三大主流web框架

```
"""
django
  特点:大而全 自带的功能特别特别特别的多 类似于航空母舰
  不足之处:
    有时候过于笨重

flask
  特点:小而精  自带的功能特别特别特别的少 类似于游骑兵
  第三方的模块特别特别特别的多，如果将flask第三方的模块加起来完全可以盖过django
  并且也越来越像django
  不足之处:
    比较依赖于第三方的开发者

tornado
  特点:异步非阻塞 支持高并发
    牛逼到甚至可以开发游戏服务器
```

```
A:socket部分
B:路由与视图函数对应关系(路由匹配)
C:模版语法

django
  A用的是别人的    wsgiref模块
  B用的是自己的
  C用的是自己的(没有jinja2好用 但是也很方便)

flask
  A用的是别人的    werkzeug(内部还是wsgiref模块)
  B自己写的
  C用的别人的(jinja2)

tornado
  A, B, C都是自己写的
```

注意事项

```
# 如何让你的计算机能够正常的启动django项目
  1.计算机的名称不能有中文
  2.一个pycharm窗口只开一个项目
  3.项目里面所有的文件也尽量不要出现中文
  4.python解释器尽量使用3.4~3.6之间的版本
    (如果你的项目报错 你点击最后一个报错信息
    去源码中把逗号删掉)

# django版本问题
  1.X 2.X 3.X(直接忽略)
  1.X和2.X本身差距也不大 我们讲解主要以1.X为例 会讲解2.X区别
  公司之前用的1.8 满满过渡到了1.11版本 有一些项目用的2.0
```

django基本操作

```
# 命令行操作
  # 1.创建django项目
    """
    你可以先切换到对应的D盘 然后再创建
    """
    django-admin startproject mysite
```

```
      mysite文件夹
        manage.py
        mysite文件夹
          __init__.py
          settings.py
          urls.py
          wsgi.py
 # 2.启动django项目
  """
    一定要先切换到项目目录下
    cd /mysite
  """
  python3 manage.py runserver
  # http://127.0.0.1:8000/
```

```
# 3.创建应用
"""
Next, start your first app by running python manage.py startapp
[app_label].
"""
  python manage.py startapp app01
  应用名应该做到见名知意
```

应用

```
"""
django是一款专门用来开发app的web框架

django框架就类似于是一所大学
app就类似于大学里面各个学院
  比如开发淘宝
    订单相关
    用户相关
    投诉相关
    创建不同的app对应不同的功能
```

```
**********创建的应用一定要去配置文件中注册**********
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app01.apps.App01Config',  # 全写
    'app01',        # 简写
]
# 创建出来的的应用第一步先去配置文件中注册 其他的先不要给我干
```

主要文件介绍

```
-mysite项目文件夹
  --mysite文件夹
    ---settings.py   配置文件
    ---urls.py       路由与视图函数对应关系(路由层)
    ---wsgi.py       wsgiref模块(不考虑)
  --manage.py        django的入口文件
  --db.sqlite3       django自带的sqlite3数据库(小型数据库 功能不是很多还有bug)
  --app01文件夹
    ---admin.py      django后台管理
    ---apps.py       注册使用
    ---migrations文件夹      数据库迁移记录
    ---models.py     数据库相关的 模型类(orm)
    ---tests.py      测试文件
    ---views.py      视图函数(视图层)
```

django小白必会三板斧

```
"""
HttpResponse
  返回字符串类型的数据

render
  返回html文件的

redirect
  重定向
    return redirect('https://www.mzitu.com/')
    return redirect('/home/')
"""
```

```
from django.shortcuts import HttpResponse,render,redirect

return HttpResponse('字符串')

return render(request,'login.html')
def ab_render(request):
    # 视图函数必须要接受一个形参request
    user_dict = {'username':'jason','age':18}
    # 第一种传值方式:更加的精确 节省资源
    # return render(request,'01 ab_render.html',
{'data':user_dict,'date':123})
    # 第二种传值方式:当你要传的数据特别多的时候
    """locals会将所在的名称空间中所有的名字全部传递给html页面"""
    return render(request,'01 ab_render.html',locals())

return redirect(url)
```

静态文件配置

```
# 登陆功能

"""
我们将html文件默认都放在templates文件夹下
我们将网站所使用的静态文件默认都放在static文件夹下

静态文件
  前端已经写好了的 能够直接调用使用的文件
    网站写好的js文件
    网站写好的css文件
    网站用到的图片文件
    第三方前端框架
    ...
"""
```

```
# django默认是不会自动帮你创建static文件夹 需要你自己手动创建
一般情况下我们在static文件夹内还会做进一步的划分处理
  -static
    --js
    --css
    --img
    其他第三方文件

"""
在浏览器中输入url能够看到对应的资源
是因为后端提前开设了该资源的借口
如果访问不到资源 说明后端没有开设该资源的借口

http://127.0.0.1:8000/static/bootstrap-3.3.7-dist/css/bootstrap.min.css
"""

# 静态文件配置

"""
**********************************************************************
当你在写django项目的时候 可能会出现后端代码修改了但是前端页面没有变化的情况
  1.你在同一个端口开了好几个django项目
    一直在跑的其实是第一个django项目

  2.浏览器缓存的问题
    settings
      network
        disable cache 勾选上
**********************************************************************
```

```
STATIC_URL = '/ooo/'  # 类似于访问静态文件的令牌
"""如果你想要访问静态文件 你就必须以static开头"""
"""
/static/bootstrap-3.3.7-dist/js/bootstrap.min.js

/static/令牌
取列表里面从上往下依次查找
    bootstrap-3.3.7-dist/js/bootstrap.min.js
    都没有才会报错
"""
# 静态文件配置
STATICFILES_DIRS = [
    os.path.join(BASE_DIR,'static'),
    os.path.join(BASE_DIR,'static1'),
    os.path.join(BASE_DIR,'static2'),
```

```
# 静态文件动态解析
    {% load static %}
    <link rel="stylesheet" href="{% static 'bootstrap-3.3.7-
dist/css/bootstrap.min.css' %}">
    <script src="{% static 'bootstrap-3.3.7-dist/js/bootstrap.min.js'
%}"></script>
```

request对象方法初识

```
request.method # 返回请求方式 并且是全大写的字符串形式  <class 'str'>
request.POST  # 获取用户post请求提交的普通数据不包含文件
  request.POST.get()      # 只获取列表最后一个元素
  request.POST.getlist()  # 直接将列表取出
```

```
def login(request):
    # 返回一个登陆界面
    """
    get请求和post请求应该有不同的处理机制
    :param request: 请求相关的数据对象 里面有很多简易的方法
    :return:
    """
    # print(type(request.method))  # 返回请求方式 并且是全大写的字符串形式
<class 'str'>
    # if request.method == 'GET':
    #     print('来了 老弟')
    #     return render(request,'login.html')
    # elif request.method == 'POST':
    #     return HttpResponse("收到了 宝贝")

    if request.method == 'POST':
        return HttpResponse("收到了 宝贝")
    return render(request, 'login.html')
```

```
request.GET  # 获取用户提交的get请求数据
  request.GET.get()  # 只获取列表最后一个元素
  request.GET.getlist()  # 直接将列表取出
"""
get请求携带的数据是有大小限制的
```

```
# django链接MySQL
  1.第一步配置文件中配置
    DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'day60',
        'USER':'root',
        'PASSWORD':'admin123',
        'HOST':'127.0.0.1',
        'PORT':3306,
        'CHARSET':'utf8'
    }
```

2.代码声明

django默认用的是mysqldb模块链接MySQL

但是该模块的兼容性不好 需要手动改为用pymysql链接

你需要告诉django不要用默认的mysqldb还是用pymysql

```
# 在项目名下的init或者任意的应用名下的init文件中书写以下代码都可以
import pymysql
pymysql.install_as_MySQLdb()
```