Cd.. 跳出一级目录 cd 目录名 进入一级目录

1. 创建项目

在工作空间目录下执行 django-admin startproject projectName

1. 创建app

在项目的目录下 执行 python manage.py startapp appName

1. 将自己创建的app 配置到项目中

在项目settings 文件 添加

#配置需要的app

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'myapp01', 自己添加

]

1. 写程序

在app应用中 定义试图函数 views 文件

Def fun(req):

#可执行代码

Return render(req,”模板名称”)/redirect()

1. 在项目下的urls文件 中配置url
2. 文件中引入自己的app的views模块
3. from django.urls import path
4. from myapp01 import views as myViews #别名 引入自己的app模块
5. urlpatterns = [
6. path('admin/', admin.site.urls),
7. #设置自己的url
8. path("te/test/",myViews.test)
9. ]
10. 启动服务器 python manage.py runserver
11. 访问自定义试图函数

<http://127.0.0.1:8080/url>

1. 模板页面：

在app应用中创建templates 文件夹 在此文件夹中创建html 文件

1. 数据交互：

页面提交方式是get

<body>

<form action="/user/log" method="get">

帐号：<input type="text" name="userAcc" placeholder="请输入帐号" required >d

<br>

密码：<input type="password" name="userPass"> <br>

<input type="submit" value="登录">

</form>

</body>

Views试图函数获取的方式是：

Info=Request.GET.get(“paramName”)

*def* login(*req*):

#获得客户提交数据 get请求方式

userAcc= req.GET.get("userAcc",None) #获得帐号信息

userPass=req.GET.get("userPass",None) #获得密码

print(userAcc+"===="+userPass)

if "admin"==userAcc and "123456" == userPass: #判断帐号密码

print("帐号密码存在")

return render(req,"test.html")

return render(req,"login.html")

views 中的函数要在urls 中配置path

get -> 1.form 表单提交method=get 时 2.访问超链接时 3.地址栏直接访问

post form 表单提交method=post时 django 中post 方式提交 需要设置跨域请求 csrf\_token

如果提示APPEND\_SLASH 错误时在项目setting 文件中配置

APPEND\_SLASH=False #设置

Post 提交时 表单代码中加入

｛% csrf\_token %｝

<form action="/user/reg/" method="POST">

{% csrf\_token %}

帐号：<input type="text" name="userAcc" placeholder="请输入帐号" required >d

<br>

密码：<input type="password" name="userPass"> <br>

确认密码：<input type="password" > <br>

地址：<select name="addr">

<option value="北京">北京</option>

<option value="上海">上海</option>

<option value="山西">山西</option>

<option value="陕西">陕西</option>

</select>

<select name="addr">

<option value="北京">北京</option>

<option value="上海">上海</option>

<option value="太原">太原</option>

<option value="西安">西安</option>

</select>

<input type="submit" value="注册">

</form>

Views 中写： 在urls 文件中配置path

#注册功能

*def* reg(*req*):

print("---------------------------")

userAcc=req.POST.get("userAcc") #post方式获得帐号信息

userAdd=req.POST.getlist("addr") #获得地址 一个参数多个值获取

print(userAcc+" ")

print(userAdd)

return render(req,"login.html")

请求转发 相应冲定向

# Create your views here.

#测试请求转发 一个可访问应用的转发

*def* testRender(*request*):

cont=*dict*() #字典类型 ｛key:value,key:value｝

cont["name"]='wm001' #设置数据 requset 设置数据

cont["hobby"]=['吃','爬山']

return render(request,"testReder.html",cont) #请求转发给客户端传递数据

#相应重定向 新发起请求 可以是任何可访问的url

# xiang

*def* testRdirect(*req*):

# return redirect("http://www.baidu.com")

return redirect("/page/login.html")

请求转发 可以传递数据给模板页面 最后展示给客户端

testReder.html页面中写

<body>

请求转发

{{name}}<br>

{{hobby}}

</body>

Session 使用 django 依赖于表

执行 python manage.py migrate 创建表

Session设置

settings中  
SESSION\_COOKIE\_AGE=60\*30 30分钟。  
SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE=True

：会话cookie可以在用户浏览器中保持有效期。True：关闭浏览器，则Cookie失效。

Session 中存储数据

Req.session[‘key’]=值

Html 模板页面显示session数据

模板页面取数据：{{request.session.key}}

Views中获得session中的数据：request.session.get("key")

**模板**  ---如何显示数据

常用标签 for if标签 ifequals if notequals include

Views中 请求转发传递数据

'''

模板中复杂数据展示

'''

*def* testTemplate(*req*):

cont=*dict*() #创建字典类型对象

cont['userInfo']={"name":'王敏','age':18,'sex':'男'}

cont['infos']=[{"name":'王敏','age':18,'sex':'女','hobby':['吃','爬山']},

{"name":'谷美颖','age':16,'sex':'女','hobby':['跳舞','弹钢琴']},

{"name":'李文浩','age':16,'sex':'男','hobby':['打游戏','打篮球','打台球']}]

#cont={'userInfo':{"name":'王敏','age':18,'sex':'女'},'infos'=[{"name":'王敏','age':18,'sex':'女','hobby':['吃','爬山']},

# {"name":'谷美颖','age':16,'sex':'女','hobby':['跳舞','弹钢琴']},

# {"name":'李文浩','age':16,'sex':'男','hobby':['打游戏','打篮球','打台球']}]}

return render(req,"testTeplate.html",*context*=cont)

模板中显示testTeplate.html

{% include 'top.html' %} 包含页面

显示数据<br>

{{userInfo}}<br>

{# 显示字典中数据 --注释 #}

姓名 {{userInfo.name }} 年龄 {{userInfo.age }} 性别 {{userInfo.sex }}<!-- 自定对象中的属性信息 -->

{#显示list、set、tuple 类型数据 #}

<br>

<table border="1">

<tr><td>姓名</td><td>性别</td><td>年龄</td><td>hobby</td></tr>

{% for us in infos %}

<tr><td>{{us.name}}</td><td>{{us.sex}}</td><td>{{us.age}}

</td>

<td><!--{{us.hobby}} -->

{% for ho in us.hobby %} {{ho}}、 {%endfor%}

</td></tr>

{%endfor%}

</table>

<hr>

--------------------if----------------------

性别 {{userInfo.sex}} <br>

男 <input type="radio" name="sex" {% if userInfo.sex == '男' %} checked {%endif%} >

女 <input type="radio" name="sex" {% if userInfo.sex == '女' %} checked {%endif%} >

<br>

年龄范围

{% if userInfo.age >= 60 %}

老年人

{% elif userInfo.age >= 30 and userInfo.age < 60 %}

中年人

{%else%}

青年

{%endif%}

{% include 'bottom.html' %} <!-- 包含页面 --></60>

模型：Model 使用mysql数据库

1．安装mysql 数据库相关模块 pip install pymysql

2.Django项目settings 文件中配置数据库

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql ',

'HOST':'127.0.0.1',

'PORT':3306,

'USER':'ROOT',

'PASSWORD':'ROOT',

'NAME': 'DJANGODATABASE',

'OPTIONS': {'isolation\_level':None}

}

}

3.写模型对象

Models.py文件中定义：

from django.db import models

# Create your models here.

#定义模型对象

*class* UserInfo(*models*.*Model*):

'''

列 类型 要求

用户编号 int pk

用户帐号 varchar unique

用户密码 varchar

生日 date

性别 varchar

'''

#属性定义 表中列定义

userID=models.BigIntegerField(*primary\_key*=True ) #编号设置为主键

userAccount=models.CharField(*max\_length*=50,*unique*=True)

userPass=models.CharField(*max\_length*=30)

userBirth=models.DateField()

userSex=models.CharField(*max\_length*=4)

*class* Meta:

db\_table='userInfotable'

pass

3.在 项目文件下\_\_init\_\_ 文件中加入：

import pymysql

pymysql.install\_as\_MySQLdb()

4.创建注册模型对象

Python manage.py makemigrations

5.创建表

Python magage.py migrate