<https://yiyibooks.cn/zdj999123/Django111_LTS/index.html>

今天只不过是django的入门

**Django**

是一个开放源代码的Web应用框架,由Python写成

最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻（时效性，快速部署）内容为主的网站

例如可以处理：https://news.sina.com.cn/下各类型新闻为一个目录

客户端

view

Template

Model

MTV模式

• Django的MTV模式本质上和MVC是一样的,也是为了各组件间保持松耦合关系,只是定义上有些许不同

– M 代表**模型**(Model):负责业务对象和数据库的关系映射(ORM)

– T 代表**模板** (Template):负责如何把页面展示给用户(html)

– V 代表**视图**(View):负责业务逻辑,并在适当时候调用Model和Template

• 除了以上三层之外,还需要一个URL分发器,它的作用是将一个个URL的页面请求分发给不同的View处理,View再调用相应的Model和Template

解释：

客户端访问不同的页面；urlconf

view返回相应的页面（页面其实是一个个的函数）；

Model对应数据库（不需要写sql语句，类SQLAlchemy分关系型数据库自动对应）（class与数据库的表关联）；

Template对应网页

搭建django

]# cat ~/.pip/pip.conf ##使用国内镜像仓库

[global]

index-url = http(djenv) [root@room8pc16 day20]# cat ~/.pip/pip.conf

[global]

index-url = http://pypi.douban.com/simple/

[install]

trusted-host=pypi.douban.com

配置虚拟环境

Python3中已经自带虚拟环境,只要创建并激活即可

###意思相当于docker搭建容器

1、创建虚拟环境

[root@room8pc16 day20]# mkdir -pv /opt/py3env

[root@room8pc16 day20]# python3 -m venv /opt/py3env/djenv

2、激活虚拟环境

[root@room8pc16 day20]# source /opt/py3env/djenv/bin/activate

激活之后，运行python默认是py3

3、激活之后，安装django，就是安装在了虚拟环境下

(djenv) [root@room8pc16 day20]# pip install django==1.11.6

(djenv) [root@room8pc16 day20]# pip install pymysql

查看当前环境安装的软件包 ##可以通过本地文件安装django

# pip freeze

# pip freeze > pkgs.txt

下载

# pip install --download /存储目录 pkgs.txt

安装本地软件包

# pip install 文件名

DJANGO

1、创建项目

(djenv) [root@room8pc16 day20]# django-admin startproject mysite

(djenv) [root@room8pc16 day20]# cd mysite/

2、pycharm设置，使用虚拟环境

file -> settings -> project xxx-> project interpreter -> 添加现有环境

找到/opt/py3env/djenv/bin/python

3、为django项目创建数据库

(djenv) [root@room8pc16 mysite]# mysql -uroot -ptedu.cn

MariaDB [(none)]> create database dj\_tedu default charset utf8;

4、修改django的设置 settings.py

ALLOWED\_HOSTS = '\*' # 允许所有客户端访问

DATABASES = {

'default': {

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

'NAME': 'dj\_tedu',

'USER': 'root',

'PASSWORD': 'tedu.cn',

'HOST': '127.0.0.1',

'PORT': '3306',

}

}

LANGUAGE\_CODE = 'zh-Hans'

TIME\_ZONE = 'Asia/Shanghai'

5、加载连接mysql的模块，在\_\_init\_\_.py中写入以下内容

import pymysql

pymysql.install\_as\_MySQLdb()

6、启动服务

(djenv) [root@room8pc16 mysite]# python manage.py runserver 0:8000

7、生成数据库

(djenv) [root@room8pc16 mysite]# python manage.py migrate

8、创建后台管理员

(djenv) [root@room8pc16 mysite]# python manage.py createsuperuser

9、启动服务后，访问http://127.0.0.1:8000/admin

**django基础测试**

**1、新建视图**

# vim mysite/views.py

from django.shortcuts import HttpResponse

def index(request):

return HttpResponse('<h1>Hello World!</h1>')

2、修改urls.py，访问主页时，使用views模块的index函数响应

from django.conf.urls import url

from django.contrib import admin

from . import views

urlpatterns = [

url(r'^admin/', admin.site.urls),

url(r'^$', views.index, name='index'), # name='index'是首页URL的名字

]

##实例：

管理“投票”应用 ##poll

1、创建应用

(djenv) [root@room8pc16 mysite]# python manage.py startapp polls

2、在项目中注册应用，修改settings.py

INSTALLED\_APPS = [

... ...

'polls', ##添加此行

]

3、规划url，polls应用下有很多个页面，但是每个页面访问时，都以

http://127.0.0.1/polls开头，如http://127.0.0.1/polls/1

polls应用有自己的URLConf，当访问到http://127.0.0.1/polls开头的网址

都交给polls应用的urls.py

# vim mysite/urls.py

from django.conf.urls import url, include

urlpatterns = [

... ...

url(r'^polls/', include('polls.urls')),

]

4、创建应用的url映射文件

# vim polls/urls.py

from django.conf.urls import url

from . import views

urlpatterns = [

url(r'^$', views.polls\_index, name='polls\_index'),

]

5、创建视图

# vim polls/views.py

from django.shortcuts import render, HttpResponse

def polls\_index(request):

return HttpResponse('<h1>Polls index</h1>')

6、创建模板

（1）创建模板目录

[root@room8pc16 mysite]# mkdir -p polls/templates/polls

##templates目录名字规定这个

（2）创建模板文件

# vim polls/templates/polls/polls\_index.html

（3）更改视图views.py

def polls\_index(request):

return render(request, 'polls/polls\_index.html')

7、添加boostrap样式

将前面解压的bootstrap拷贝过来

[root@room8pc16 mysite]# cp -r ../../day19/static/ polls/

在模板文件polls\_index.html中添加引用

文件顶部增加一行：

{% load staticfiles %}

head标签中加入

<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">

**建立模型Model**

**• 在这个简单的投票应用中,我们将创建两个模型:Question和Choice**

**• Question对象具有一个question\_text(问题)属性和一个publish\_date(发布时间)属性**

**• Choice有两个字段:选择的内容和选择的得票统计**

**• 每个Choice与一个Question关联**

1、创建ORM映射类

# vim polls/models.py

from django.db import models

class Question(models.Model):

question\_text = models.CharField(max\_length=200)

publish\_date = models.DateTimeField()

def \_\_str\_\_(self): #回显输入的内容

return self.question\_text

class Choice(models.Model):

choice\_text = models.CharField(max\_length=200)

votes = models.IntegerField(default=0)

question = models.ForeignKey(Question, models.CASCADE)

def \_\_str\_\_(self):

return self.choice\_text

2、生成数据库中的表

(djenv) [root@room8pc16 mysite]# python manage.py makemigrations

(djenv) [root@room8pc16 mysite]# python manage.py migrate

3、在数据库中show tables查看表名

多了两个表：

polls\_choice

polls\_question

4、查看表中字段

（1）django会在表中自动加入一列id，作为主键

（2）表中的每个字段与类变量相同

（3）外键是类变量名加上\_id

5、把模型注册到后台管理界面，修改polls/admin.py如下：

from django.contrib import admin

from .models import Question, Choice

admin.site.register(Question)

admin.site.register(Choice)

6、访问http://127.0.0.1:8000/admin，在后台可以添加数据