**Python4-4**

**基于python的框架**

**• Django是一个开放源代码的Web应用框架,由Python写成**

**• Django 框架的核心组件有:**

**– 用于创建模型的对象关系映射**

**– 为最终用户设计的完美管理界面**

**– 一流的 URL 设计**

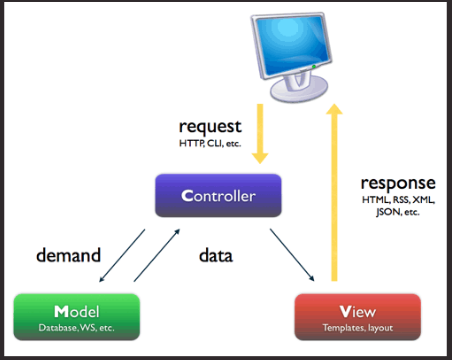
**– 设计者友好的模板语言**

**– 缓存系统**

**MVC模式**

**• 把数据存取逻辑、业务逻辑和表现逻辑组合在一起的概念被称为软件架构的 Model-View-Controller(MVC)模式**

**• 在这个模式中, Model 代表数据存取层,View 代表的是系统中选择显示什么和怎么显示的部分,Controller 指的是系统中根据用户输入并视需要访问模型,以决定使用哪个视图的那部分**

****

**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**

**MTV响应模式**

**• Web服务器(中间件)收到一个http请求**

**• Django在URLconf里查找对应的视图(View)函数来处理http请求**

**• 视图函数调用相应的数据模型来存取数据、调用相应的模板向用户展示页面**

**• 视图函数处理结束后返回一个http的响应给Web服务器**

**• Web服务器将响应发送给客户端**

**• Python3中已经自带虚拟环境,只要创建并激活即可**

**[root@room9pc01 ~]# cd kuang/**

**[root@room9pc01 kuang]# python3 -m venv djenv**

**[root@room9pc01 kuang]# source djenv/bin/activate**

**(djenv) [root@room9pc01 kuang]#**

**(djenv) [root@room9pc01 kuang]# pip3 install django==1.11.6**

**• Djanog可以自动生成一些代码,这些代码创建一个Django项目:一个Django实例的设置集合,包括数据库的配置、Django有关的选项和应用有关的选项**

**(djenv) [root@room9pc01 kuang]# django-admin startproject mysite**

**(djenv) [root@room9pc01 kuang]# cd mysite/**

**(djenv) [root@room9pc01 mysite]# tree .**

**.**

**├── manage.py #**

**└── mysite**

**├── \_\_init\_\_.py**

**├── settings.py**

**├── urls.py**

**└── wsgi.py**

**1 directory, 5 files**

**• 外层的mysite/根目录仅仅是项目的一个容器。 它的名字与Django无关;可以将其重命名为你喜欢的任何内容**

**• manage.py:一个命令行工具,可以使用多种方式对Django项目进行交互。 可以在django-admin和manage.py中读到关于manage.py的所有细节**

**• 内层的mysite/目录是项目的真正的Python包。 它是导入任何东西时将需要使用的Python包的名字(例如 mysite.urls)**

**• mysite/\_\_init\_\_.py:一个空文件,它告诉Python这个目录应该被看做一个Python包**

**• mysite/settings.py:该Django 项目的设置/配置。Django settings 将告诉你这些设置如何工作。**

**• mysite/urls.py:此Django项目的URL声明;Django驱动的网站的“目录”**

**• mysite/wsgi.py:用于项目的与WSGI兼容的Web服务器入口**

**2、启动django自带的web服务器测试**

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

**3、配置pycharm使用虚拟环境**

**file -> settings -> Project day04 -> Project Interpreter -> 点右上角的齿**

**轮 -> add local -> 选择下面的existing，勾选make available ...，**

**点击右侧...，在弹出的对话框中输出/root/djenv/bin/python**

**4、创建mysql数据库，用于存储django项目的数据**

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

**MariaDB [(none)]> create database dj1807 default charset utf8;**

**5、• Django默认只允许本机访问、英文环境,这些均可在settings.py中进行修改**

**修改django配置setting.py**

**ALLOWED\_HOSTS = '\*' # 允许所有的客户端访问**

**DATABASES = {**

**'default': {**

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

**'NAME': 'dj1807',**

**'USER': 'root',**

**'PASSWORD': 'tedu.cn',**

**'HOST': '127.0.0.1',**

**'PORT': '3306',**

**}**

**}**

**LANGUAGE\_CODE = 'zh-hans'**

**TIME\_ZONE = 'Asia/Shanghai'**

**USE\_TZ = False**

**6、配置django使用pymysql模块mysite/\_\_init\_\_.py**

**import pymysql**

**pymysql.install\_as\_MySQLdb()**

**7、启动测试服务器运行在所有地址的80端口**

**(djenv) [root@room8pc16 mysite]# python manage.py runserver 0:80**

**8、生成django缺省的数据库表**

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

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

1. **创建管理员帐号，密码必须8位以上，复杂**

**• 访问后台需要有超级用户身份**

**• 超级用户需要单独创建**

**• 用户将写到数据库中**

**(django\_env) [root@localhost mysite]# python manage.py**

**createsuperuser**

**Username (leave blank to use 'root'): admin**

**Email address: zzg@tedu.cn**

**Password:**

**Password (again):**

**Superuser created successfully.**

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

**10、访问后台http://127.0.0.1/admin**

**一个项目最好分成N多个应用，可以把不同的应用分派给不同的程序员编写**

**编写投票应用APP：**

**1、数据**

**两张表：**

**question -> 记录问题**

**choice -> 记录每个问题的选项，以及该选项所得的票数**

**2、URL：所有的投票应用，网址都以http://127.0.0.1/polls/开头**

**http://127.0.0.1/polls/ -> 投票首页，列出所有的问题**

**http://127.0.0.1/polls/1/ -> 1号问题的详情页**

**http://127.0.0.1/polls/1/result/ -> 1号问题投票的结果页**

**• 应用是一个Web应用程序,它完成具体的事项 ——比如一个博客系统、一个存储公共档案的数据库或者一个简单的投票应用**

**• 项目是特定网站的配置和应用程序的集合**

**• 一个项目可以包含多个应用**

**• 一个应用可以运用到多个项目中去**

**• 应用可以放在Python path上的任何位置**

**• 可以在manage.py文件同级目录创建应用,以便可**

**以将它作为顶层模块导入,而不是mysite的子模块**

**(django\_env) [root@localhost mysite]# python manage.py startapp polls**

**4、将创建的应用注册到项目中，修改mysite/settings.py**

**INSTALLED\_APPS = [**

**... ...**

**'polls',**

**]**

**5、授权。将以http://127.0.0.1/polls/开头的URL都交给polls应用处理**

**修改mysite/urls.py**

**from django.conf.urls import url, include**

**from django.contrib import admin**

**urlpatterns = [**

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

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

**]**

**6、在polls应用下创建urls.py**

**from django.conf.urls import url**

**urlpatterns = [**

**]**