

# 初识Django

## 学习目标

整体上对Django项目的开发流程有个全面认识。详细内容会在之后的课堂上深入讲解

## Django简介

- Django是一个开放源代码的Web应用框架，由Python写成
- Django的主要目标是使得开发复杂的、数据库驱动的网站变得简单，Django注重组件的重用性和“可插拔性”，敏捷开发和DRY法则（Don't Repeat Yourself）。
- 采用MVC(MTV)设计模式

## Django的开发环境配置

### • 安装

#### ◦ 命令安装

```
#安装的是最新版本的
pip install django
#卸载框架
pip uninstall django
#指定版本号
pip install django==1.8.2
```

- Pycharm安装（自动）
- pip install 包 （安装失败）那么怎么解决？.whl(文件)

### • 环境变量

- 在cmd中输入一个命令，做了以下两件事情
- 在当前的目录下寻找输入的命令，没有的话去PATH的路径下依次寻找。
- 找到就执行，找不到就出错了
- 得出结论，配置环境变量的时候，往前面配置。

### • Django-admin

- 创建，运行项目的时候都会用到这个命令

### • 创建Django项目

#### ◦ 命令创建

```
django-admin startproject sxt
```

- Pycharm创建
- 命令运行 # 如果不指定端口，默认端口是8000 django-admin manage.py runserver 端口 # 可以运行多次，使用多个端口 # nginx

#### ◦ 成功界面

□

- 如果启动不了，很有可能是端口占用
- Pycharm console是和项目相关的一个上下文
- termial 就是cmd

### • Django项目目录介绍

- manage.py 命令工具，其实就是一个脚本
- templates 存放模板的（html）
- init.py 说明当前目录是模块(包)
- settings.py 配置文件
- wsgi 通用网关
- urls 地址（url）urls（多个地址），路由

## 第一个Django项目

- 定制index.html界面
  1. urls文件中配置url路径
  2. 书写处理函数
  3. 返回HttpResponse对象
- 完善index.html界面
  - 将处理的函数放到views里面了
  - 还可以返回html
  - 可以将html内容写在一个html文件中，之后读取内容，交给HttpResponse响应
- 引入model

- python manage.py startapp post
- 配置模块到settings中
- python manage.py makemigrations 生成迁移文件（生成sql语句）
- python manage.py sqlmigrate post 0001 查看迁移文件所对应的sql语句

```
CREATE TABLE "post_category" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(20) NOT NULL);
```

- python manage.py migrate(迁移命令)系统默认sqlite数据库（文件型数据库）（本质是根据我的迁移文件生成表结构）

```
cate = Category(name='Django全栈开发')
cate.save() 保存到数据库

Category.objects.all()获得了所有的类别
```

- admin后台
- 修改设置 汉化 LANGUAGE\_CODE = 'zh-hans'
- 修改设置 时区 TIME\_ZONE = 'Asia/Shanghai'

## Django的核心组件

- 基于正则表达式的URL分发器 (配置url路径)
- 处理请求的视图系统（views里面写的函数）
- 显示给用户的模板系统
- 面向对象的ORM映射器（对象关系影射（object relative map））

## Post

- 创建完项目的第一步

```
python manage.py migrate 先生成系统默认的一些表
```

- 创建用户

```
python manage.py createsuperuser 创建管理员
本地化
LANGUAGE_CODE = 'zh-hans'

TIME_ZONE = 'Asia/Shanghai'
```

- MD5

```
import hashlib
md5 = hashlib.md5()创建md5对象
md5.update(明文) 传入明文
md5.hexdigest()返回16进制的密文
```

- 创建模块

- python manage.py startapp sxt
- 在settings文件中的INSTALL\_APPS中配置
- 创建实体类（表）

```
class Category(models.Model):
    name = models.CharField(max_length=20)
    def __unicode__(self):
        return u'%s'%self.name
```

- python manage.py makemigrations 产生表？产生的是迁移文件（产生类似的sql语句）
- python manage.py sqlmigrate sxt 0001 查看迁移文件所对应的sql语句

```
CREATE TABLE "sxt_category" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(20) NOT NULL);
```

- python manage.py migrate 根据迁移文件生成表结构
- 理解url路由的工作流程

□

□

- 产生数据

- Template（模板）
- Context（内容，上下文）

```
#核心渲染的代码
from django.template import Template
t = Template('{{name}}:你好') #创建模板
from django.template.context import Context
c = Context({'name':'李四'})
content = t.render(c)
return HttpResponse(content)

# Template 支持这个语法
from django.template import Template
t = Template('{{name}}:你好<br>{%for data in datas%}{{data}}<br>{%endfor%}') #创建模板
from django.template.context import Context
c = Context({'name':'李四','datas':range(10)})
content = t.render(c)
return HttpResponse(content)

# 本质上还是没变
from django.template import Template
from post.settings import BASE_DIR
import os
with open(os.path.join(BASE_DIR,'abc.txt')) as fr:
    content = fr.read()
t = Template(content) #创建模板
from django.template.context import Context
c = Context({'name':'李四','datas':range(10)})
content = t.render(c)
return HttpResponse(content)

#读取html
from django.template import Template
from post.settings import BASE_DIR
import os
with open(os.path.join(BASE_DIR,'templates/index.html')) as fr:
    content = fr.read()
t = Template(content) #创建模板
from django.template.context import Context
c = Context({'name':'李四','datas':range(10)})
```

```

        content = t.render(c)
        return HttpResponse(content)

# 最终提取代码

def myrender(request,templatepath,dict):
    from django.template import Template
    from django.template.context import Context
    import os
    #创建模板
    BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
    with open(os.path.join(BASE_DIR,templatepath)) as fr:
        content = fr.read()
    t = Template(content)
    # 创建Context
    c = Context(dict)
    renderHtml = t.render(c)
    from django.http import HttpResponse
    return HttpResponse(renderHtml)

```

## MVC

- M Model(数据相关逻辑，底层操作数据库)
- V View（显示给用户的）
- C Control (业务逻辑)

□

## MTV（MVT）

- M Model(数据相关逻辑，底层操作数据库)
- V 控制器（控制业务逻辑）
- T 模板（显示给用户的） □

## 最基本的有

- 先显示类别
- html加载a
  - url路由问题有没有理解的很清晰 category/post/{{category.id}}
  - 获得类别下面所有的帖子，首先得获得类别。如何获得类别（什么能唯一的表示类别id（主键））

```

cate = Category.objects.get(id = categoryId)
cate.post_set.all()

```

- url(r'^category/post/(\d+)',views.getpostby\_category)
- getpostby\_category(request,categoryid):
- 解决了