## 来源菜鸟教程：<https://www.runoob.com/django/django-tutorial.html>

后续逐渐学习其他的操作，基本会搭，路由会用会写接口即可

## 创建第一个项目

django-admin startproject HelloWorld

接下来我们进入 HelloWorld 目录输入以下命令，启动服务器：

python3 manage.py runserver 0.0.0.0:8000

会报错：Invalid HTTP\_HOST header

解决：去HelloWorld/HelloWorld/settings.py，修改ALLOWED\_HOSTS = ['\*']

（文件位置均是最外层的相对位置）

## 视图和 URL 配置

推荐使用path函数，便于理解，且支持新版本Django，都写在一个urlpatterns

开发时前后端分离，接口返回json格式，在views里用

from django.http import JsonResponse，响应为json

## Django 模板

由于前后端分离，无需考虑模板问题

## Django 模型

#### 数据库配置

注意，在settings.py中改配置时“NAME”参数指定的库事先要建好，否则会找不到库

#### 定义模型

django-admin startapp TestModel

在settings.py里增添模型应用

在建立模型的class时，会用到包含整型、浮点型等，具体使用查看文档

包括外键、索引等操作，需要在使用的时候学习，2.0版本定义外键需要额外的参数on\_delete=models.CASCADE，否则会报错

在命令行中运行：

$ python3 manage.py migrate # 创建表结构

# 首次执行，会创建Django登陆认证所需的一系列表

$ python3 manage.py makemigrations TestModel # 让 Django 知道我们在我们的模型有一些变更

$ python3 manage.py migrate TestModel # 创建表结构

这里可以指定更改的表名，不指定的话就全部检查和创建改动的新表，表名组成结构为：应用名\_类名（如：TestModel\_test）

数据库连接有一些坑：

**错误一**：No module named 'MySQLdb'

原因：python3连接MySQL不能再使用mysqldb，取而代之的是pymysql。

解决方法：

首先pip install pymysql

在python的MySQL包中，即路径：C:\Users\adong\AppData\Local\Programs\Python\Python36\Lib\site-packages\Django-2.0.3-py3.6.egg\django\db\backends\mysql

下的\_\_init\_\_.py文件中加入：

import pymysql

pymysql.install\_as\_MySQLdb()

**错误二**：django.core.exceptions.ImproperlyConfigured: mysqlclient 1.3.3 or newer is required; you have 0.7.11.None

原因：在解决了错误一以后出现了此错误。

解决方法：在python的MySQL包中，即路径：C:\Users\adong\AppData\Local\Programs\Python\Python36\Lib\site-packages\Django-2.0.3-py3.6.egg\django\db\backends\mysql

下的 base.py 文件中，注释掉一下两行代码：

if version < (1, 3, 3):

raise ImproperlyConfigured("mysqlclient 1.3.3 or newer is required; you have %s" % Database.\_\_version\_\_)

**错误三**：AttributeError: 'str' object has no attribute 'decode'

原因：在解决了错误二以后出现了此错误。

解决方法：在python的MySQL包中，即路径：C:\Users\adong\AppData\Local\Programs\Python\Python36\Lib\site-packages\Django-2.0.3-py3.6.egg\django\db\backends\mysql

下的operations.py文件中，找到query = query.decode(errors=‘replace’)

将decode修改为encode即可

#### 数据库操作

Django的ORM语句操作数据库增删改查，可以根据需求相应学习

## Django 表单

针对像接口传参数使用，与前端商议好使用GET方法还是POST方法（我们通常是GET较多），对应从request对象的request.GET或者request.POST里得到对应的参数即可。一些基本的request类的属性与方法可以了解，但通常用的不多

## Django Admin 管理工具

Django的特点之一，方便管理，但目前看来使用数据库管理软件，这个工具有点鸡肋

你可以通过命令 **python3 manage.py createsuperuser** 来创建超级用户