# 第二章 Django入门

## 2.1 Django开发环境配置

1、安装python

2、安装Django ：pip install django==1.10.3

## 2.2 开始第一个项目应用

创建项目名：

命令：django-admin startproject guest【项目名】

创建项目应用：

命令：python manage.py startapp sign

项目文件解说：

**guest/\_\_init\_\_.py**：一个空的文件，用它标识一个目录为 Python 的标准包。

**guest/settings.py**：Django 项目的配置文件，包括 Django 模块应用配置，数据库配置，模板配置等。

**guest/urls.py**：Django 项目的 URL 声明。

**guest/wsgi.py**：为 WSGI 兼容的 Web 服务器服务项目的切入点。

**manage.py**：一个命令行工具，可以让你在使用 Django 项目时以不同的方式进行交互。

应用文件解释：

migrations/：用于记录 models 中数据的变更。

admin.py：映射 models 中的数据到 Django 自带的 admin 后台。

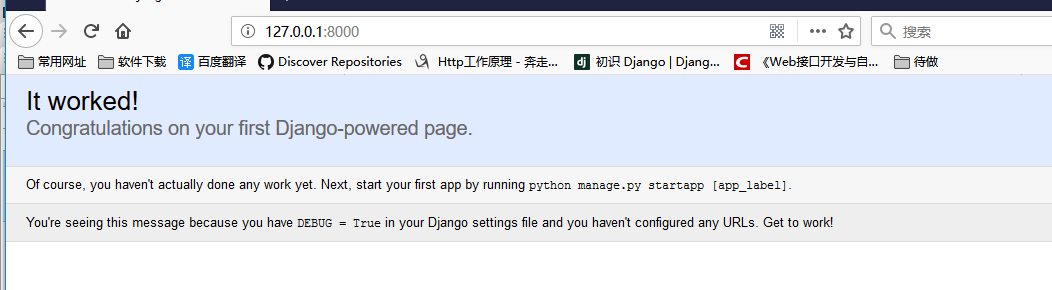
apps.py：在新的 Django 版本中新增，用于应用程序的配置。

models.py：创建应用程序数据表模型（对应数据库的相关操作）。

tests.py：创建 Django 测试。

views.py：控制向前端显示哪些数据。





## 创建一个Hello Django

我们首先需要配置一下 guest/settings.py 文件，将 sign 应用添加到项目中。



在guest/urls中添加访问的路径

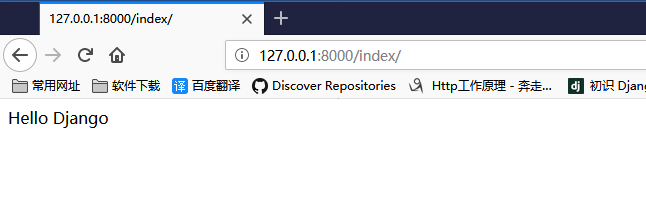
1. 在urls中导入sign的views，**from sign import views**
2. 添加路径：url(r'^index/$', views.index),
3. 在sign 中添加代码

from django.http import HttpResponse

# Create your views here.

def index(request):

return HttpResponse("Hello Django!")



## 使用模板

在应用 sign/目录下创建 templates/index.html 文件。

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Title</title>

</head>

<body>

<h1>Hello Django!</h1>

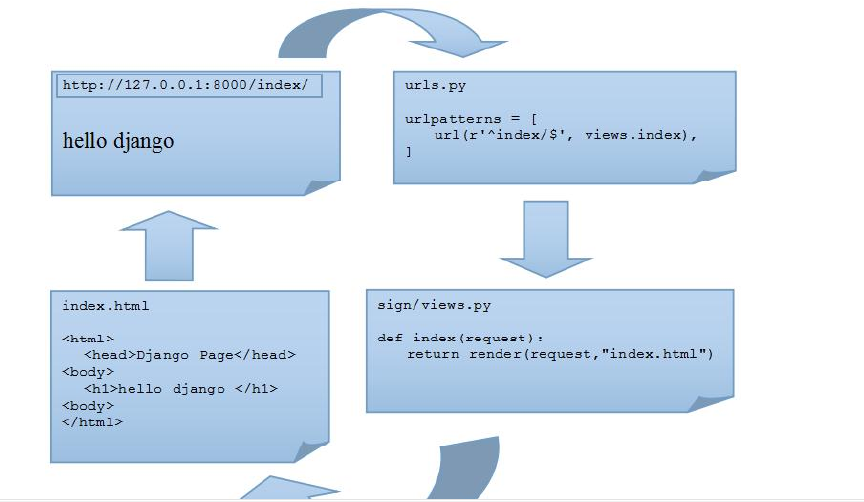
</body>

</html>

Views修改：

def index(request):

return render(request, "index.html")



# 第三章 Django视图

在index.html修改视图

<html>

<head>

<title>Django Page</title>

</head>

<body>

<h1>发布会管理</h1>

<form>

<input name="username" type="text" placeholder="username" ><br>

<input name="password" type="password" placeholder="password"><br>

<button id="btn" type="submit">登录</button>

</form>

</body>

</html>

## Get与Post之间的区别

GET - 从指定的资源请求数据。

POST - 向指定的资源提交要被处理的数据

Django 针对 CSRF 的保护措施是在生成的每个表单中放置一个自动生成的令牌通过这个令牌判断 POST请求是否来自同一个网站。

## 3.1.3 登录成功页

1、新建event\_manage.html，在url中添加路径以及在views中写login\_action的函数

def login\_action(request):

if request.method == 'POST':

username = request.POST.get('username', '')

password = request.POST.get('password', '')

if username == 'admin' and password == 'admin123':

return **HttpResponseRedirect**('/event\_manage/')

else:

return render(request, 'index.html', {'error': "username or password error"})

def event\_manage(request):

return render(request, "event\_manage.html")

**HttpResponseRedirect：它可以对路径进行重定向，从而将登录成功之后的请求**

**指向/event\_manage/目录。**

## 3.2 cookie和session

from sign.models import Event,Guest

## 5.2.2 发布会搜索

event\_manage.html 页面上创建搜索表单

div class="page-header" style="padding-top: 60px;">

<div id="navbar" class="navbar-collapse collapse">

<form class="navbar-form" method="get" action="/search\_name/">

<div class="form-group">

<input name="name" type="text" placeholder="名称" class="form-control">

</div>

<button type="submit" class="btn btn-success"> 搜索 </button>

</form>

</div>

</div>

添加路径url(r'^search\_name/$', views.search\_name),

打开.../sign/views.py 文件，创建 search\_name()视图函数。

@login\_required

def search\_name(request):

username = request.session.get('user', '')

search\_name = request.GET.get("name", "")

event\_list = Event.objects.filter(name\_\_contains=search\_name)

return render(request, "event\_manage.html", {"user": username, "events": event\_list})