

今日内容

django下载安装

官网地址:<https://www.djangoproject.com/download/>

1、下载Django:

```
pip3 install django==1.11.9
```

2、创建一个django project

```
django-admin startproject mysite
```

 创建了一个名为"**mysite**"的Django 项目:

3、创建的项目文件介绍

`manage.py` ----- Django项目里面的工具, 通过它可以调用django shell和数据库, 启动关闭项目与项目交互等, 不管你将来分几个文件, 必然有一个启动文件, 其实他们本身就是一个文件。

`settings.py` ----- 包含了项目的默认设置, 包括数据库信息, 调试标志以及其他一些工作的变量。

`urls.py` ----- 负责把URL模式映射到应用程序。

`wsgi.py` ----- `runserver`命令就使用`wsgiref`模块做简单的web server, 后面会看到`runserver`命令, 所有与socket相关的内容都在这个文件里面了, 目前不需要关注它。

4、运行项目

```
python manage.py runserver 127.0.0.1:8080
```

 #此时已经可以启动django项目了, 只不过什么逻辑也没有呢

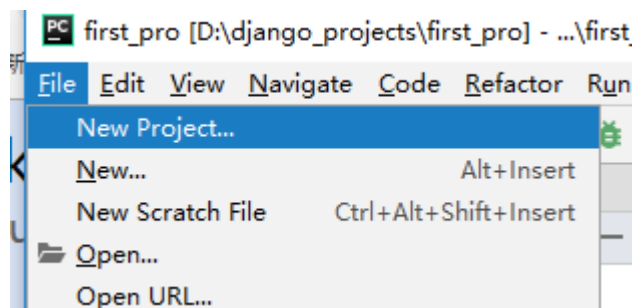
通过指令运行项目的时候, 可以不写ip地址, 如果不写, 默认是127.0.0.1

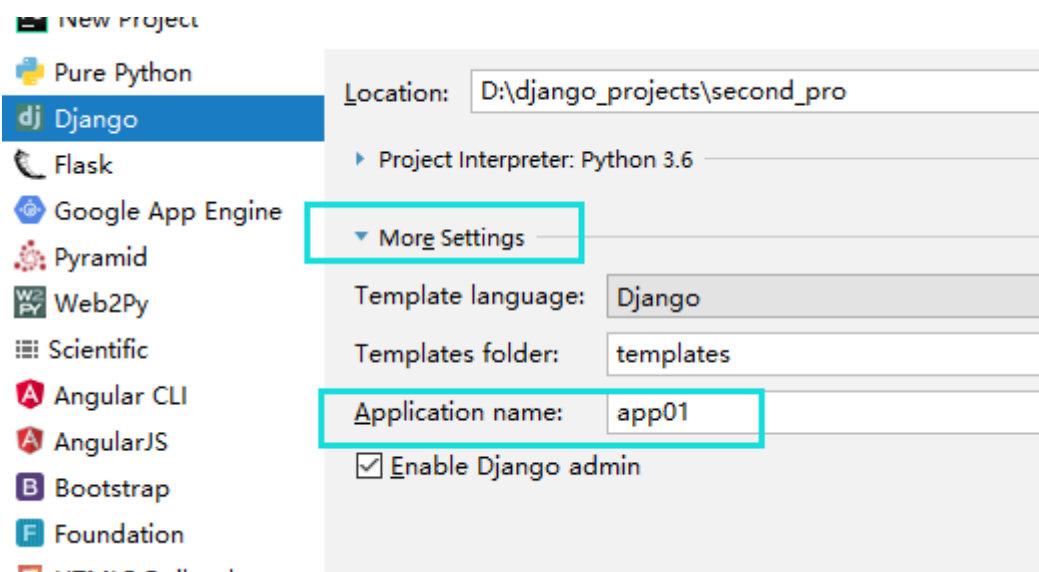
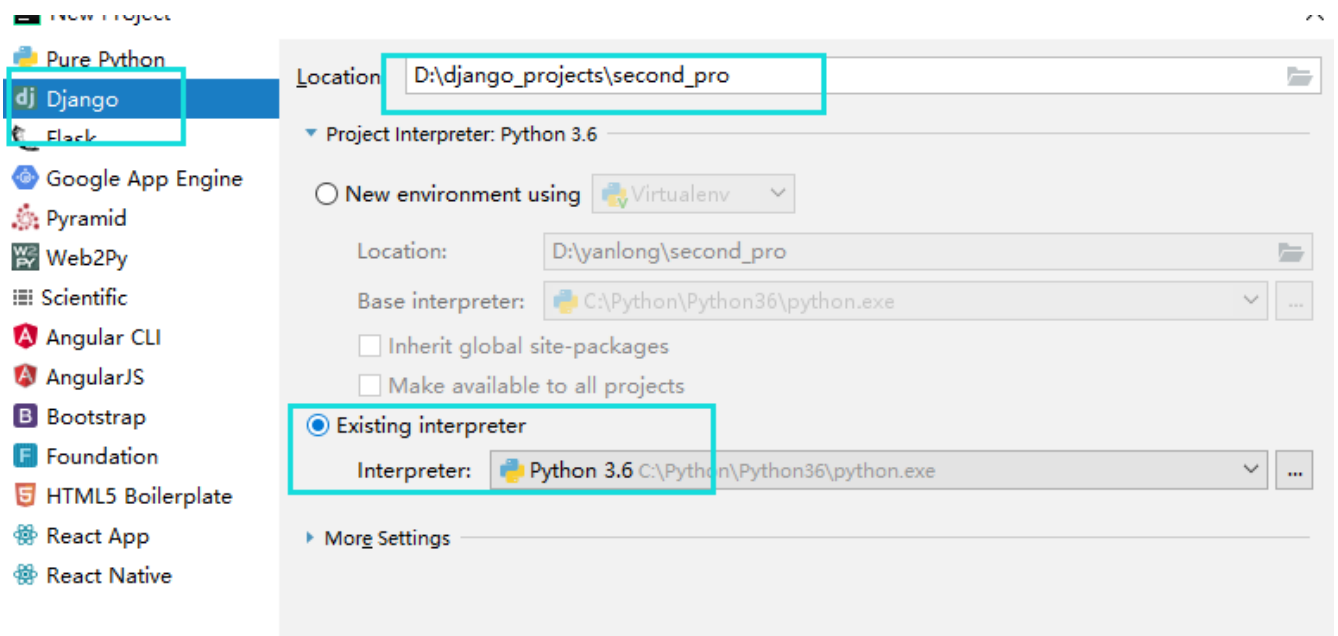
```
python manage.py runserver 8080
```

如果连端口都不写, 默认是8000端口

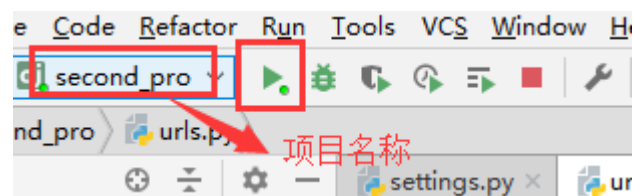
```
python manage.py runserver
```

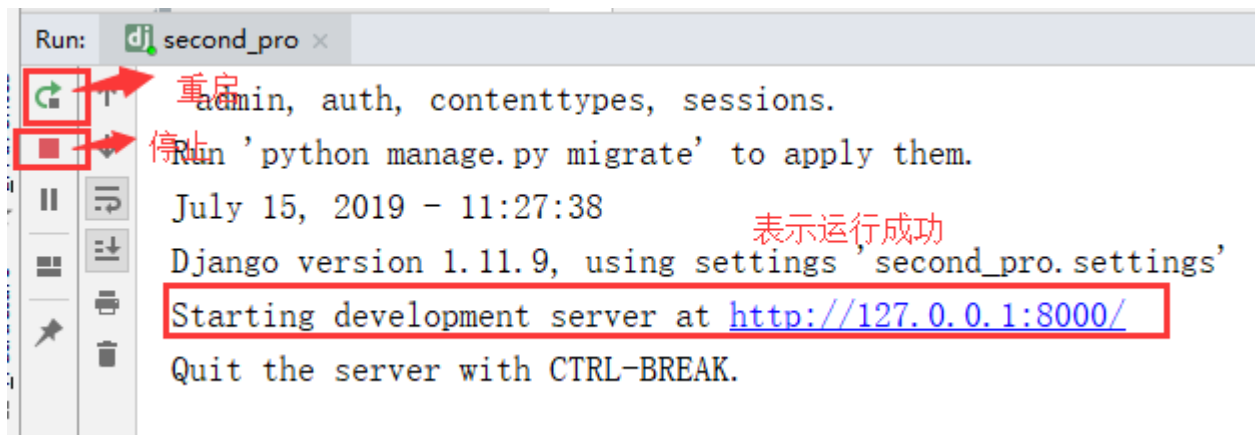
pycharm创建项目





运行:





写项目

1 创建项目 first_pro

2 创建app app01

做一个登录页面的web项目,浏览器输入一个网址得到一个web页面

用户: http:127.0.0.1:8001/login/

1 urls.py

```
from django.conf.urls import url
from django.contrib import admin
from app01 import views

urlpatterns = [
    # url(r'^admin/', admin.site.urls),
    url(r'^index/', views.index), #配置路径,r'^/index/',这个前置导航斜杠不需要添加.
]
```

2 写逻辑视图

```
from django.shortcuts import render,HttpResponse
# Create your views here.

def index(request):
    print(request.method) # 'POST' 'GET'
    if request.method == 'GET':
        return render(request,'login.html')
    else:
        # print(request.GET)
        print(request.POST)
        username = request.POST.get('username')
        password = request.POST.get('password')
        if username == 'dazhuang' and password == '123':
            return HttpResponse('登录成功!')
        else:
            return HttpResponse('登录失败!')
```

3 创建html文件

在templates文件夹中创建一个login.html文件

```

<!DOCTYPE html>
<html lang="zh-CN">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- 上述3个meta标签*必须*放在最前面, 任何其他内容都*必须*跟随其后! -->
  <title>Bootstrap 101 Template</title>
  <!-- Bootstrap -->
</head>
<body>

<h1>欢迎来到登录页面</h1>

<form action="/index/" method="post">
  用户名:<input type="text" name="username">
  密码:<input type="text" name="password">
  <input type="submit">

</form>

</body>
</html>

```

urls.py里面需要注意的问题

url(r'^index/', views.index), 第一个参数:路径正则字符串 第二个参数:对应的视图函数

views.py

```

def login(request):
    获取请求方法,request.method-- 'GET', 'POST'
    request.GET -- GET请求发送来的所有数据,queryDict类型
    request.POST -- POST请求发送来的所有数据,queryDict类型
    <input name='username'>
    request.POST.get('username')
    request.GET.get('username')
    return HttpResponse('字符串') --- 回复字符串数据
    return render(request, 'login.html')

```

get请求获取数据

```
def index(request):
    print(request.GET) #<QueryDict: {'username': ['dazhuang'], 'password': ['123']}>
    username = request.GET.get('username')
    password = request.GET.get('password')
    print(username,password)

    if username == 'dazhuang' and password == '123':
        return HttpResponse('登录成功!')
    else:
        return HttpResponse('失败!')
```

post请求提交数据时关掉一个认证机制,settings配置文件中

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    #'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

url配置

无名分组

```
url(r'^books/(\d{4})/', views.books),
```

位置参数

```
url(r'^books/(\d{4})/', views.year_books), #匹配年份的
url(r'^books/(\d{4})/(\d{1,2})/', views.year_month_books), #匹配年份和月份的
# http://127.0.0.1:8000/books/2001/
```

视图:

```
def year_month_books(request,year,month): #位置参数,第一个参数接收的就是无名分组路径中匹配到的第一个分组的数据,第二个参数接收的就是无名分组路径中匹配到的第二个分组的数据
    print(year,month)
    # return render(request,'books.html')
    return HttpResponse(year+month)
```

有名分组

```
url(r'^books/(?P<year>\d{4})/(?P<month>\d{1,2})/', views.year_month_books), #匹配年份和月份的
```

```
def year_month_books(request, month, year): #形参名称要和url中的分组名对应好,参数位置就没有要求了
```

```
    print(year, month)
```

```
    # return render(request, 'books.html')
```

```
    return HttpResponse(year+month)
```

默认值

```
# urls.py中
```

```
from django.conf.urls import url
```

```
from . import views
```

```
urlpatterns = [
```

```
    url(r'^blog/$', views.page),
```

```
    url(r'^blog/page(?P<num>[0-9]+)/$', views.page),
```

```
]
```

```
# views.py中, 可以为num指定默认值
```

```
def page(request, num="1"):
```

```
    pass
```