

Django 创建第一个项目

本章我们将介绍Django 管理工具及如何使用 Django 来创建项目，第一个项目我们以 HelloWorld 来命名项目。

测试版本说明：

- Python 3.6.0
- Django 2.1.7

Django 管理工具

安装 Django 之后，您现在应该已经有了可用的管理工具 **django-admin**。我们可以使用 **django-admin** 来创建一个项目：

我们可以来看下django-admin 的命令介绍：

```
$ django-admin
```

```
Type 'django-admin help <subcommand>' for help on a specific subcommand.
```

```
Available subcommands:
```

```
[django]
```

```
check
```

```
compilemessages
```

```
createcachetable
```

```
dbshell
```

```
diffsettings
```

```
dumpdata
```

```
flush
```

```
inspectdb
```

```
loaddata
```

```
makemessages
```

```
makemigrations
```

```
migrate
```

```
runserver
```

```
sendtestemail
```

```
shell
```

```
showmigrations
```

```
sqlflush
```

```
sqlmigrate
```

```
sqlsequencereset
```

```
squashmigrations
```

```
startapp
```

```
startproject
```

```
test
```

```
testserver
.....省略部分.....
```

创建第一个项目

使用 django-admin 来创建 HelloWorld 项目：

```
django-admin startproject HelloWorld
```

创建完成后我们可以查看下项目的目录结构：

```
$ cd HelloWorld/
$ tree
.
|-- HelloWorld
|   |-- __init__.py
|   |-- settings.py
|   |-- urls.py
|   `-- wsgi.py
`-- manage.py
```

目录说明：

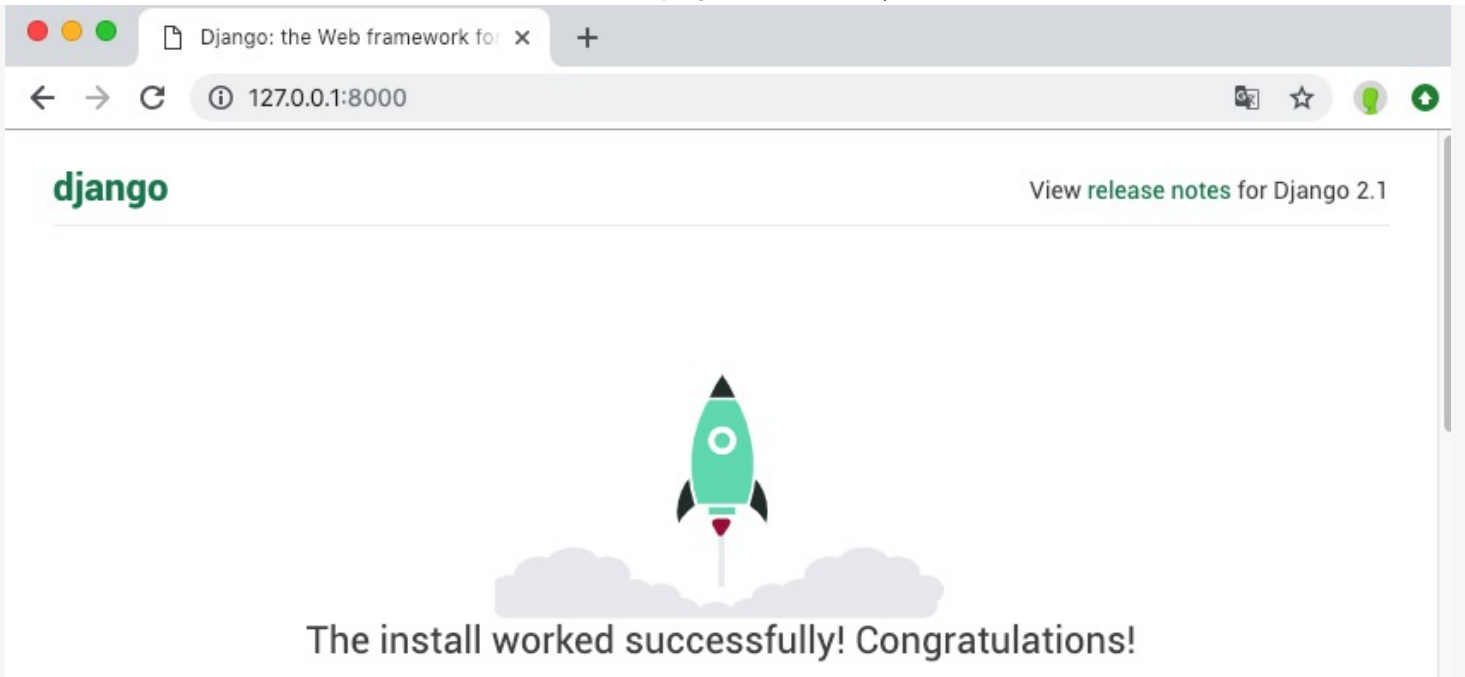
- **HelloWorld:** 项目的容器。
- **manage.py:** 一个实用的命令行工具，可让你以各种方式与该 Django 项目进行交互。
- **HelloWorld/__init__.py:** 一个空文件，告诉 Python 该目录是一个 Python 包。
- **HelloWorld/settings.py:** 该 Django 项目的设置/配置。
- **HelloWorld/urls.py:** 该 Django 项目的 URL 声明; 一份由 Django 驱动的网站"目录"。
- **HelloWorld/wsgi.py:** 一个 WSGI 兼容的 Web 服务器的入口，以便运行你的项目。

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
```

0.0.0.0 让其它电脑可连接到开发服务器，8000 为端口号。如果不说明，那么端口号默认为 8000。

在浏览器输入你服务器的 ip（这里我们输入本机 IP 地址：**127.0.0.1:8000**）及端口号，如果正常启动，输出结果如下：



视图和 URL 配置

在先前创建的 HelloWorld 目录下的 HelloWorld 目录新建一个 view.py 文件，并输入代码：

HelloWorld/HelloWorld/view.py 文件代码：

```
from django.http import HttpResponse
def hello(request):
    return HttpResponse("Hello world ! ")
```

接着，绑定 URL 与视图函数。打开 urls.py 文件，删除原来代码，将以下代码复制粘贴到 urls.py 文件中：

HelloWorld/HelloWorld/urls.py 文件代码：

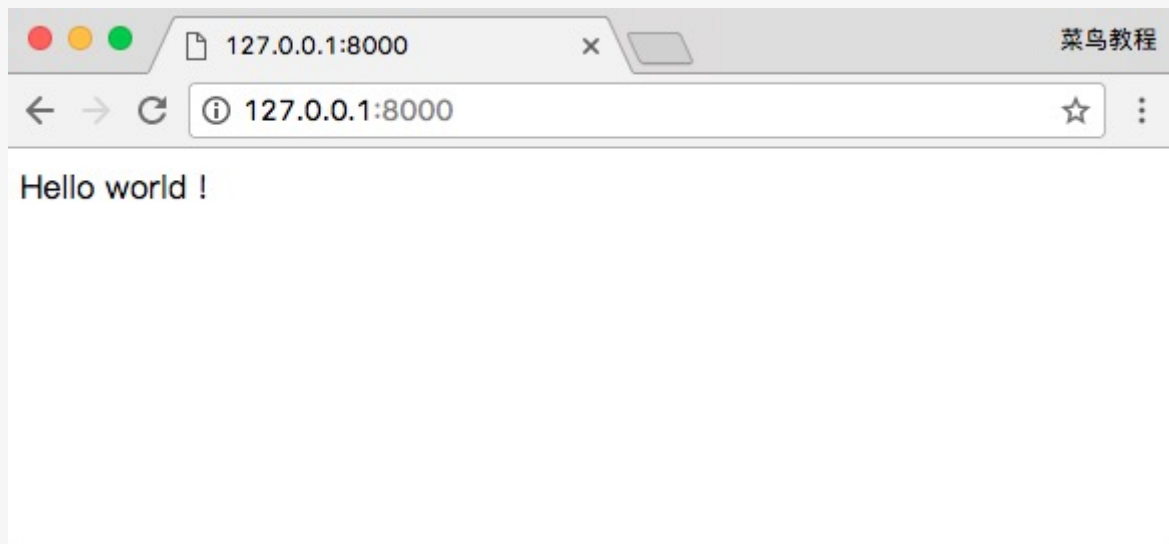
```
from django.conf.urls import url
from . import view
urlpatterns = [
    url(r'^$', view.hello),
]
```

整个目录结构如下：

```
$ tree
.
|-- HelloWorld
|   |-- __init__.py
|   |-- __init__.pyc
|   |-- settings.py
|   |-- settings.pyc
|   |-- urls.py          # url 配置
|   |-- urls.pyc
|   |-- view.py          # 添加的视图文件
|   |-- view.pyc         # 编译后的视图文件
```

```
| |-- wsgi.py
| |-- wsgi.pyc
|-- manage.py
```

完成后，启动 Django 开发服务器，并在浏览器访问打开浏览器并访问：

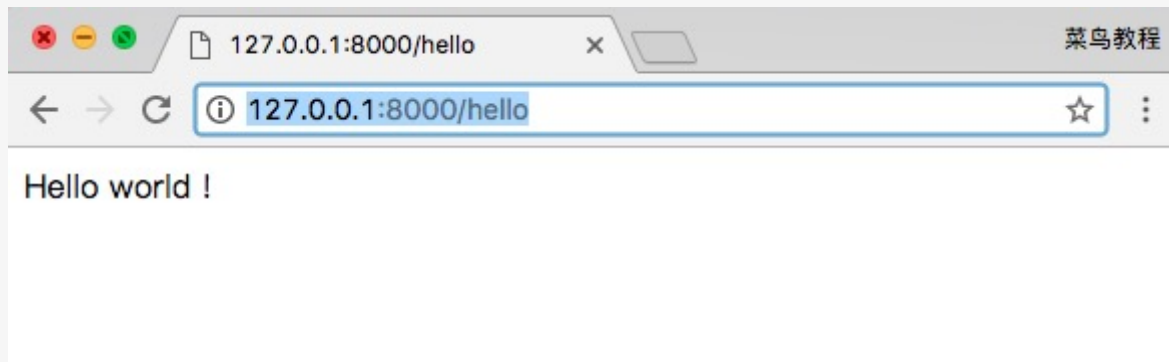


我们也可以修改以下规则：

HelloWorld/HelloWorld/urls.py 文件代码：

```
from django.urls import path
from . import view
urlpatterns = [
    path('hello/', view.hello),
]
```

通过浏览器打开 <http://127.0.0.1:8000/hello>，输出结果如下：



注意：项目中如果代码有改动，服务器会自动监测代码的改动并自动重新载入，所以如果你已经启动了服务器则不需手动重启。

path() 函数

Django path() 可以接收四个参数，分别是两个必选参数：route、view 和两个可选参数：kwargs、name。

语法格式：

```
path(route, view, kwargs=None, name=None)
```

- route: 字符串, 表示 URL 规则, 与之匹配的 URL 会执行对应的第二个参数 view。
- view: 用于执行与正则表达式匹配的 URL 请求。
- kwargs: 视图使用的字典类型的参数。
- name: 用来反向获取 URL。

Django2.0中可以使用 `re_path()` 方法来兼容 1.x 版本中的 `url()` 方法, 一些正则表达式的规则也可以通过 `re_path()` 来实现。

```
from django.urls import include, re_path

urlpatterns = [
    re_path(r'^index/$', views.index, name='index'),
    re_path(r'^bio/(?P<username>\w+)/$', views.bio, name='bio'),
    re_path(r'^weblog/', include('blog.urls')),
    ...
]
```

[← Django 安装](#)[Django 模板 →](#)**3 篇笔记** **写笔记**