

18 | 初识Flask：搭建第一个Flask项目

2023-06-02 Barry 来自北京

《Python实战·从0到1搭建直播平台》



你好，我是 Barry。

从这节课开始，我们就进入到后端实战篇。这个章节重点培养的是我们从 0 到 1 做项目后端开发的能力。我们将经历从代码设计到具体功能的模块接口开发，深度体验独立平台搭建和后端开发的完整链路，学会灵活应用框架技术来应对多种类型的项目需求。

shikekey.com转载分享

这节课，我们要拿下的就是后端部分的第一个核心知识点—— Flask 项目搭建。之所以选择基于 Python 的 Flask 框架，是因为它更合适用来完成视频直播平台的后端开发。

shikekey.com转载分享

为什么选用 Flask


为了让你知道选型的来龙去脉。我们先聊聊为什么要选用 Flask 框架，而不是选择使用 Django 这个应用也很广泛的框架呢？

这就要说到 Flask 和 Django 的区别了。首先，二者都是企业当中用的很多的 Web 的后端框架。接下来，我们一起来看一下二者的对比。

Flask 主要有两大优点。第一个优点，它是轻量级 Python 开发框架。意思就是 Flask 核心就是几个文件，不用大量的配置文件，很快就能搭建起来一个小型的 Web 应用并快速启动运行，很适合搞微服务。

第二个优点就是灵活，其实这不光是 Flask 的优点，也是 Python 的优点，因为 Flask 是基于 Python 的 Web 应用框架。灵活的地方就在于这个框架用的人多，整体生态比较好。Flask 有很多别人做好的扩展包，并且整体处在一个持续的维护和更新状态，我们可以灵活使用这些包和库做定制和扩展。它的 API 部分也不需要大量的配置文件和模板，几行代码就可以完成一个实现简单功能的应用。

我们可以结合后面这个例子来看看。通过这个例子，我们不难发现 API 开发在 Flask 中不复杂，几行代码就可以实现一个简单的功能。

 复制代码

```
1 from flask import Flask, jsonify
2 app = Flask(__name__)
3 def get_weather(city):
4     # 根据城市名称查询天气
5     # 返回查询结果
6     return {'city': city, 'weather': 'sunny'}
7 @app.route('/weather/<city>')
8 def weather(city):
9     # 调用get_weather函数获取天气情况
10    result = get_weather(city)
11    # 将查询结果以json格式返回
12    return jsonify(result)
13 if __name__ == '__main__':
14    app.run()
```

上面这段代码里，我们定义了一个 `get_weather` 函数，用来根据城市名称查询天气情况。我们使用了 `@app.route` 这个装饰器，把 `get_weather` 函数绑定到 `"/weather/"` 的 URL 路径上。这样当用户去访问这个 URL 的时候，Flask 就会自动调用 `get_weather` 函数，并将查询结果以 JSON 格式返还给用户。

说完了 Flask，我们再来聊一聊同样是基于 Python 的 Web 框架——Django。那么 Django 有什么优点呢？

首先，Django 的库非常完整，就像一个工具箱一样，你想要的它都有。

其次，Django 有着比较严密的架构设计，安全性也比较高，可以保护网站抵御像 SQL 注入这样的恶意操作。

最后，Django 可以直接用 Python 语言对数据库进行操作，这一点也比较方便。

Django 的库完整、架构设计严密这些优点，注定了它非常适用于大型的应用，比如电子商务网站。不过，Django 的优点其实也算是它的缺点，因为 Django 架构严密，又非常完整的库，这就造成了一个问题——Django 的学习成本比较高。

无论应用大小怎样，Django 所需要的大量配置文件都少不了，而且占用的系统资源比较多，比如像机器性能、网路带宽等等这些资源。应用不需要的功能模块，Django 也包括在内了，所以它不如 Flask 轻便。

我们总结下 Django 和 Flask 的区别。Django 就像是一个工具箱，你要的它都有；而 Flask 是一把锤子，非常轻便，只要你善于使用它，就可以做出更多的、自由度更大的东西。

说到这里，我们已经明白了选用 Flask 的原因。考虑到我们平台的特点，还有后端学习路径循序渐进的原则，Flask 更合适一些。

Flask 知识点梳理

接下来我们就一起来看看，学习 Flask 都应该掌握哪些知识点。使用 Flask 去做项目的话，有几样技术知识需要去掌握：**Python 语言、数据库和前端应用。**

想要掌握 Flask，就需要我们对 Python 语言的应用非常熟悉。通过前面先导课的学习，对于 Python 语言你或多或少已经具备了一些基础。

其中需要重点掌握的就是 **Python 语言应用**。比如变量的应用、Python 当中的一些控制结构语言（比如 if 的 while、for 循环等等简单的控制语句，还有像函数要怎么构造、类怎么构造，怎样使用类当中的操作等等，这些都是高频用到的知识。

第二类技术知识就是**数据库**。从字面意思理解，数据库就是存储数据的一个仓库，能对项目中的数据进行管理和操作。因为我们的在线视频项目有很多数据，比如用户上传的视频、用户的个人信息预计日志等等，这些都需要存放的容器。而这个容器就是数据库，因此我们整个项目都不留离不开它。这里我也明确一下，数据库我们使用 MySQL 来实现。

第三就是**前端应用**。前面我们刚刚完成前端语言和框架的学习，相信你已经能够使用 Vue 完成整个前端部分的实现了。等我们学完后端部分，就可以实现整个平台前后端的联动，完成平台整体功能的开发。

接下来就是实践环节，我们一起来完成整个 Flask 框架的搭建。

框架搭建

在开始之前，你需要先准备好项目所使用到的 IDE。安装成功之后，我们再来实现 Flask 框架的搭建。

第一步：安装项目所使用的 IDE

我们在后端开发中使用 Pycharm 这个工具，你可以直接到 [Pycharm 的官网](#)。它的安装比较简单，你根据教程一步步操作就能搞定。这里我们需要选择 community，也就是社区版本。

shikey.com转载分享

shikey.com转载分享



版本: 2023.1.1
内部版本号: 231.8770.66
2023年4月28日

[系统要求](#)[安装说明](#)[其他版本](#)[第三方软件](#)

下载 PyCharm

Windows macOS Linux

Professional

适用于科学和 Web Python 开发。支持 HTML、JS 和 SQL。

[下载](#)[.dmg](#)

可免费试用 30 天

Community

适用于纯 Python 开发

[下载](#)[.dmg](#)

免费，开源

选择适用于 Intel 或 Apple Silicon 的安装程序

第二步：配置 Pycharm 的编译环境

下一步我们需要创建一个新的项目。在 PyCharm 中，一个项目对应一个文件夹。我们需要在 PyCharm 的欢迎界面，点击 **“Create New Project”** 按钮，然后输入项目的名称和存放路径。


这里我给项目起的名字是 “First_Flask”，你可以自己创建命名。点击 New environment using，选择 “Virtualenv”，然后在 Base interpreter 中选择 Python 版本。这里我使用的是软件 anaconda 自带的 Python 解释器，你使用先导课中下载过的 Python 解释器即可。若系统未自动检测到，还需要点击右边省略号图标，找到并选中 python.exe 文件路径。

第三步：安装 Flask

接下来就是安装环节。在 Pycharm 下方，有一个 Terminal 按钮，点击之后，就会弹出后面截图里的对话框。



我们使用 `pip` 命令来安装，在下方的终端——Terminal 输入命令行即可。

 复制代码

```
1 pip install flask
```

这里的 `pip` 其实是一个 Python 语言的包管理器，用来安装和管理 Python 包和依赖。

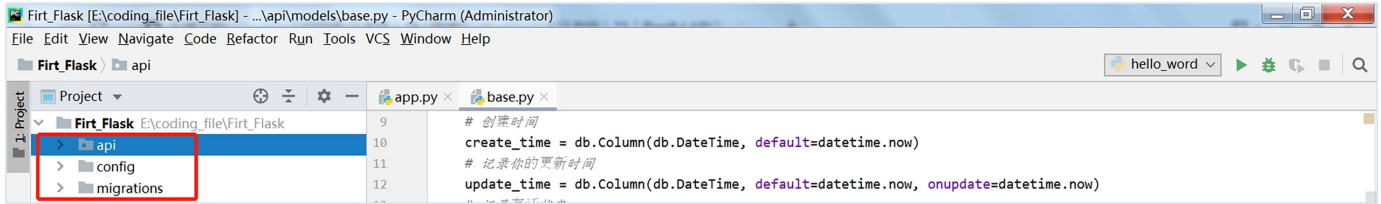
第四步：框架文件的搭建

由于项目会有许多的程序，我们提前设置好整个项目目录结构。后端开发中，我们需要创建文件夹来存放不同功能的代码块。

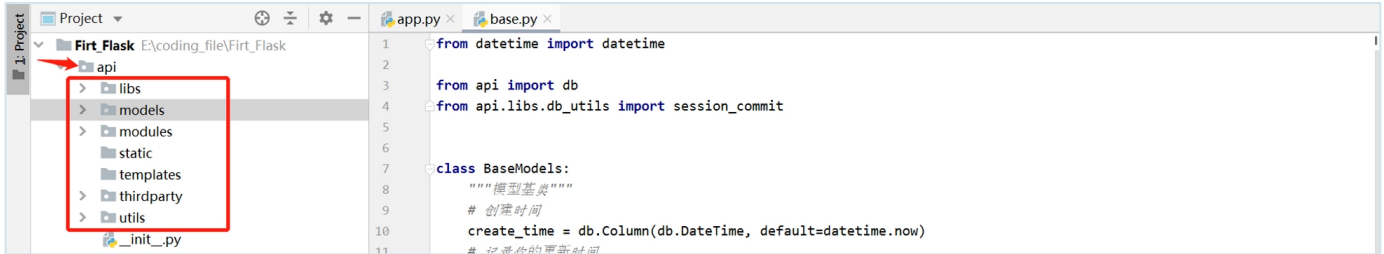
接下来，我们看看这个项目需要的 Python Package。我们先把基础的几个文件夹建好。每个文件夹的作用你可以参考后面的表格，之后我们有其他需要再创建即可。我画了一张表，帮你梳理了整个 `api` 文件下的项目目录。

文件名	用途
api	存放应用程序的 API 路由和相关函数等
config	其中包含一些与评论回复相关的配置项等
migrations	存放数据库的操作记录以及数据库的表结构变化记录等
libs	包含了一些管理后台的函数或者工具类
models	存放我们的核心功能代码。比如 <code>admin.py</code> 、用户的登入登出操作、 <code>video.py</code> 等
modules	存放不同模块，包含一些 Flask 的蓝图等
static	静态资源文件，比如 CSS 文件、Javascript 和字体文件
templates	存放我们的模板，HTML 模板文件
thirdparty	存放用户上传的头像图片等
utils	应用程序的一些工具函数文件，比如数据库工具类或者函数

后面是对应的项目目录图，你可以对照图片来了解文件之间的层级关系。

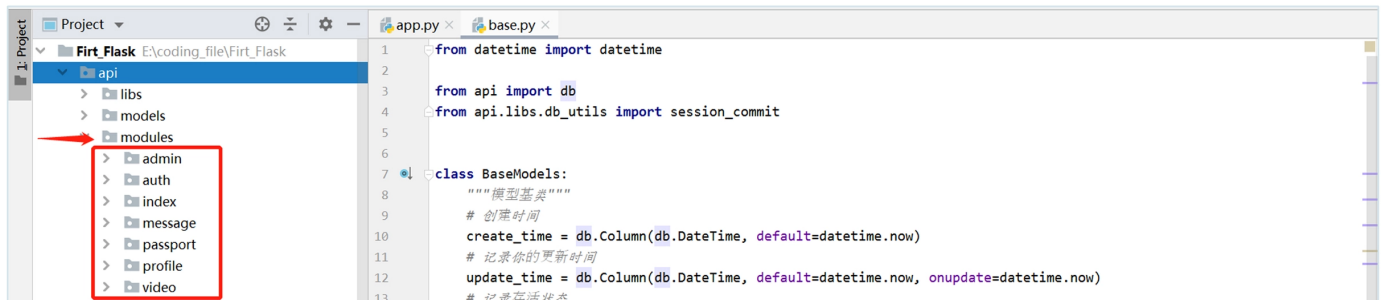


极客时间



极客时间

对于 model 文件夹中的内容，主要存放的是 admin, video, message 等项目文件。



极客时间

下面进入实战环节，我们通过简单的案例尝试在浏览器中输出打印 hello word，借此感受一下 Flask 的魅力。

我们先来新建一个 hello_word 的 Python 文件。只需要在项目文件下面直接创建即可，对应的代码参考如下。

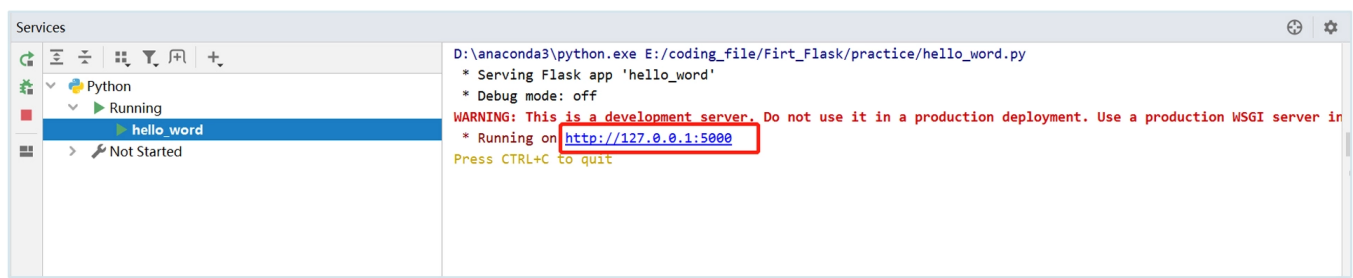
复制代码

```
1 from flask import Flask      # 导入Flask模块
2 app = Flask(__name__)        # 创建Flask实例，并指定模块名
```

```
3 @app.route('/') # 定义路由，即当访问 根目录 时返回下面的函数结果
4 def hello_world():
5     return 'Hello, World!' # 返回字符串Hello, World!
6 if __name__ == '__main__':
7     app.run() # 运行Flask应用程序
```

请注意，写在最后的 if **name** == 'main' 方法中 **name** 表示当前模块名，如果是直接运行的 Python 脚本，则为 **main**，所以该判断语句整体的作用是判断当前模块是否为主模块，如果是则执行下面的语句。

接着，我们一起来看看运行结果，运行完前面那段代码，在 Pycharm 的终端就会显示后面截图这样的运行结果，生成一个 URL。



极客时间

紧接着，你需要在浏览器中输入该地址，然后就可以看到对应的效果了。



极客时间

到这里，我们就完成了 Flask 框架的搭建，并且完成了一个简单的案例测试。建议你在课后用配套代码亲自动手练习一下，这样学习效果会更好。

总结

课程接近尾声了，我来带你对今天学到的内容做一个回顾总结。

我们先是比较了 Flask 和 Django 两种框架的特点。结合我们平台的需要和学习路径的合理性，我们最终选择了更合适做视频直播平台后端开发的 Flask 框架。

之后，我带你梳理了掌握 Flask 必备的技术知识，包括 Python 基础应用和数据库操作等，虽然这一部分我们已经在先导课学过了，但如果想更熟练地应用框架，还是建议你课后继续强化练习。

最后，通过 Flask 框架的搭建实操，我们一起演练了 Flask 安装到项目搭建的整个过程。至于 hello word 这个简单案例，其实也是为后续我们使用 Flask 开发后端做一个预热，后续课程里我们还会让 Flask 发挥更大的价值。

思考题

你可以尝试设置两个路由，实现——用户输入不同 URL，页面显示不同内容。

期待你在留言区和我交流互动。如果觉得这节课的内容实用有趣，不妨推荐给你的伙伴，和他一起学习进步。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

精选留言 (4)



peter

2023-06-02 来自北京

请教老师几个问题：

Q1：微服务需要很多组件，比如springCloud就包括配置中心、服务发现、网关、负载均衡等。Flask做微服务，功能组件齐全吗？

Q2：get_weather绑定到url上，是个笔误吧。

文中有“把 get_weather 函数绑定到 “/weather/” 的 URL 路径上”，感觉是weather函数绑定到url的，也许是个笔误。

Q3：Flask不能用python语言访问数据库吗？

Q4: 用PyCharm, 每个项目都需要指定python.exe吗?

Q5: 对于文件夹的解释, "libs"和"thirdparty"的解释不太对吧。 Thirdparty:存放用户上传的头像等。 "libs": 包含了管理后台的一些函数或工具等。

Q6: 平级目录怎么变成包含关系了?

"后面是对应的项目目录图, 你可以对照图片来了解文件之间的层级关系", 这句话下面有两个图, 上面的图中目录还是平级关系, 下面怎么就不是平级了? (紧挨着下面的图同样)

Q7: 怎么判断是main模块?

Hello_world.py为什么是main模块? 标志是什么? 是放在特定目录下面就是main吗? 如果在hello_world.py同样的目录下面创建一个aaa.py, 也是main吗?

另外, 放在哪个目录下面就不会是main了?

作者回复: 1、Flask 可以用于构建微服务。Flask是一个轻量级的Python Web框架, 易于扩展, 因此非常适合用于开发微服务。

2、言简意赅了, 可能会造成一些误解

3、Flask可以使用Python语言访问数据库。

4、每个项目都需要指定python.exe。但是, 如果你的PyCharm版本较新, 它会自动检测所使用的Python版本, 因此你不必手动指定Python解释器, 否则都需要进行配置。

5、你的说法是正确的, 我们重点在明确每个文件夹的作用, 方便项目管理。

6、这个我说了, 把api放第一个是解释一下它的作用, 他们非平级, 你看下面的图就知道关系了。

7、在 Python 中, 当一个文件被运行时, 它就成为了 main 模块。如果和 hello_world.py 在同一个目录下创建了一个 aaa.py 文件, 那么 aaa.py 就不会被当做 main 模块运行, 除非在 aaa.py 中包含了上述的代码块并且直接运行该文件。

整体的回答就是这样的情况, 希望你继续努力, 继续加油。



1



Forest

2023-06-02 来自四川

老师, 每次在新加一个路由后, 就需要重新运行程序, 否则就访问不到新加的路由, 有什么方式可以实现热重载呢?

作者回复: 你可以使用热部署工具, 比如 Flask-LiveServer 或 Flask-Wox, 这些工具可以在不重启程序的情况下更新代码和路由。同时, Flask 社区开发了许多第三方扩展, 可以帮助你解决自动重新加载问题。其中最受欢迎的是 Flask-Script, 它提供了一个交互式命令行界面, 可用于运行管理和操作 Fla

sk 应用程序，你可以尝试一下，继续加油，希望对你有帮助。



Forest

2023-06-02 来自四川

实现——用户输入不同 URL，页面显示不同内容

```
from flask import Flask # 导入Flask模块
```

```
app = Flask(__name__) # 创建Flask实例，并指定模块名
```

```
@app.route('/') # 定义路由，即当访问 根目录 时返回下面的函数结果
```

```
def hello_world():
```

```
    return 'Hello, World!' # 返回字符串Hello, World!
```

```
@app.route('/any/<content>')
```

```
def other(content):
```

```
    return '输入的内容是：{}'.format(content)
```

```
if __name__ == '__main__':
```

```
    app.run() # 运行Flask应用程序
```

shikey.com转载分享

浏览器或者postman 中访问：http://127.0.0.1:5000/any/flask

现实结果为：输入的内容是：flask

作者回复：完成的非常不错，花了心思，希望你继续保持，持续登顶！！





友

2023-06-02 来自云南

前端的终于过去了 我亲爱的老师前端完整代码上传了吗 嘻嘻 😄

作者回复: 近期整理就上传, 逐一留意git, 惊喜马上呈现

共 2 条评论 >



shikey.com转载分享

shikey.com转载分享