**Flask 快速入门**

**步骤一：安装 Flask**

首先需要安装 Flask。可以使用 pip 安装 Flask：

pip install flask

**步骤二：创建 Flask 应用**

创建一个 Python 文件（例如 app.py），导入 Flask 模块并创建一个 Flask 应用实例。

**from** flask **import** Flask

app **=** Flask(\_\_name\_\_)

**步骤三：创建视图函数**

创建视图函数并将其与 URL 绑定，以便在请求到达该 URL 时调用该视图函数并返回响应。

@app**.**route('/')

**def** **hello**():

**return** 'Hello, World!'

在上面的示例中，我们创建了一个简单的视图函数 hello，并将其与根 URL（/）绑定。当用户访问应用的根 URL 时，该函数将返回一个包含字符串“Hello, World!”的响应。

**步骤四：运行应用**

在 Python 文件中添加以下代码以运行 Flask 应用：

if \_\_name\_\_ == '\_\_main\_\_':

app.run()

运行 Flask 应用的最简单方法是在终端中运行 Python 文件：

python app.py

这将在本地主机上的默认端口（5000）上启动 Flask 应用。

**目录树**

在创建 Flask 应用时，通常会组织应用程序的代码和资源以保持良好的结构。以下是一个基本 Flask 应用的目录树示例：

├── app.py

├── static

│ └── style.css

├── templates

│ └── index.html

└── uploads

* app.py：包含 Flask 应用的主要代码。
* static/：包含应用程序使用的静态文件（例如 CSS、JavaScript 和图像文件）。
* templates/：包含应用程序使用的 HTML 模板。
* uploads/：用于存储上传的文件。

**创建 /upload 接口**

要创建 /upload 接口，我们需要添加一个视图函数来处理文件上传。Flask 使用 request.files 对象来访问上传的文件。

**from** flask **import** request

@app**.**route('/upload', methods**=**['POST'])

**def** **upload\_file**():

file **=** request**.**files['file']

file**.**save('uploads/' **+** file**.**filename)

**return** 'file uploaded successfully'

在上面的代码中，我们创建了一个名为 upload\_file 的视图函数，并将其与 /upload URL 绑定。我们使用 request.files 对象访问上传的文件，并将其保存在 uploads/ 目录中。最后，我们返回一个简单的字符串来表示文件已成功上传。

注意：在上面的代码中，我们假设上传的文件的表单字段名为 file。确保表单中的文件字段与代码中使用的名称相同。

完整的 app.py 文件应如下所示：

**from** flask **import** Flask, request

app **=** Flask(\_\_name\_\_)

@app**.**route('/')

**def** **hello**():

**return** 'Hello, World!'

@app**.**route('/upload', methods**=**['POST'])

**def** **upload\_file**():

file **=** request**.**files['file']

file**.**save('uploads/' **+** file**.**filename)

**return** 'file uploaded successfully'

**if** \_\_name\_\_ **==** '\_\_main\_\_':

app**.**run()

在上面的代码中，我们导入了 Flask 和 request 模块，并在应用程序中创建了两个视图函数。hello 视图函数将在根 URL 上显示“Hello, World!”，而 upload\_file 视图函数将在 /upload URL 上处理文件上传。

请注意，我们使用了 methods 参数来限制 upload\_file 视图函数仅接受 POST 请求。我们还使用 request.files 对象访问上传的文件，并使用 file.save() 方法将文件保存在 uploads/ 目录中。

**运行应用**

在完成上述步骤后，我们可以在终端中运行应用程序：

Copy code

python app.py

运行应用程序后，我们可以在浏览器中访问 http://localhost:5000/ URL 来查看“Hello, World!”消息，或者使用文件上传表单来上传文件到 /upload URL。希望这可以帮助你快速入门 Flask 并创建一个文件上传接口！

**结语**

总的来说，Flask 是一款灵活、轻量级的 Python Web 应用框架，能够帮助开发人员快速构建 Web 应用程序。通过本文的介绍，我们可以了解 Flask 的主要特点和优势，以及如何快速入门并创建一个文件上传接口。同时，Flask 也是一个非常活跃的社区，提供了丰富的扩展库和文档，这使得开发人员能够更加轻松愉快地使用 Flask 开发应用程序。如果你正在寻找一个简单而强大的 Python Web 应用框架，Flask 绝对是一个不错的选择。