

Postman 工具使用指南

目标

使用 postman 工具构建请求，访问 Minimal API 应用暴露的接口。

步骤

1. 安装 postman 工具

从官网 <https://www.postman.com> 下载 postman 桌面端软件，安装。

2. 访问所有 fruit 记录

运行 Minimal API 项目。

访问所有 fruit 记录对应 `app.MapGet("/fruit", () => Fruit.All)` 端点。

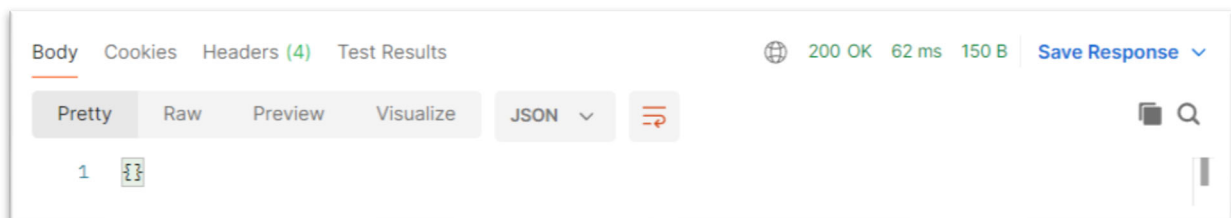
打开 postman，构建 GET 请求，资源路径为 `/fruit`。

注意，在 postman 输入 URL 的域名和端口号部分必须跟命令行的项目输出信息一致。

```
nfo: Microsoft.Hosting.Lifetime[14]
     Now listening on: https://localhost:7007
nfo: Microsoft.Hosting.Lifetime[14]
     Now listening on: http://localhost:5127
nfo: Microsoft.Hosting.Lifetime[0]
     Application started. Press Ctrl+C to shut down.
nfo: Microsoft.Hosting.Lifetime[0]
     Hosting environment: Development
nfo: Microsoft.Hosting.Lifetime[0]
     Content root path: C:\Users\Lee\source\repos\CH01-MinimalAPI
```



可以看到 postman 响应部分显示空 JSON 数据，表示目前没有任何 fruit 记录。



3. 添加 fruit 记录

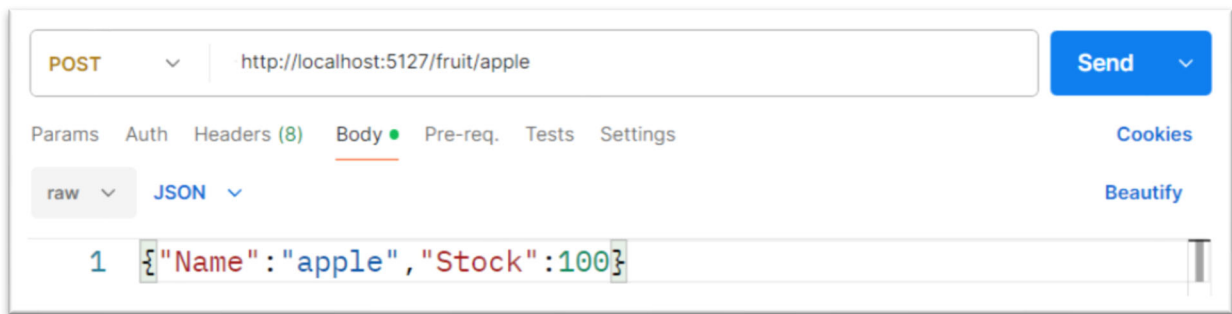
确保项目在运行状态。

添加 fruit 对应 `app.MapPost("/fruit/{id}", Handlers.AddFruit)` 端点。

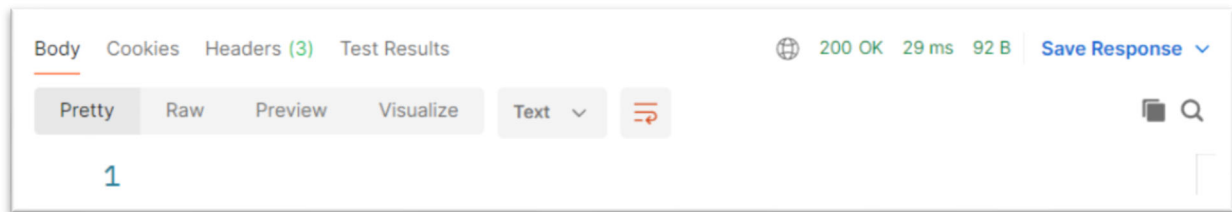
打开 postman，构建 POST 请求，资源路径为 `/fruit/apple`。表示 fruit 的 id 为 apple，作为 `AddFruit` 方法的 id 参数。

在请求的正文部分输入以下 JSON 数据，该数据将提交到服务器端，作为 `AddFruit` 方法的 `fruit` 参数。

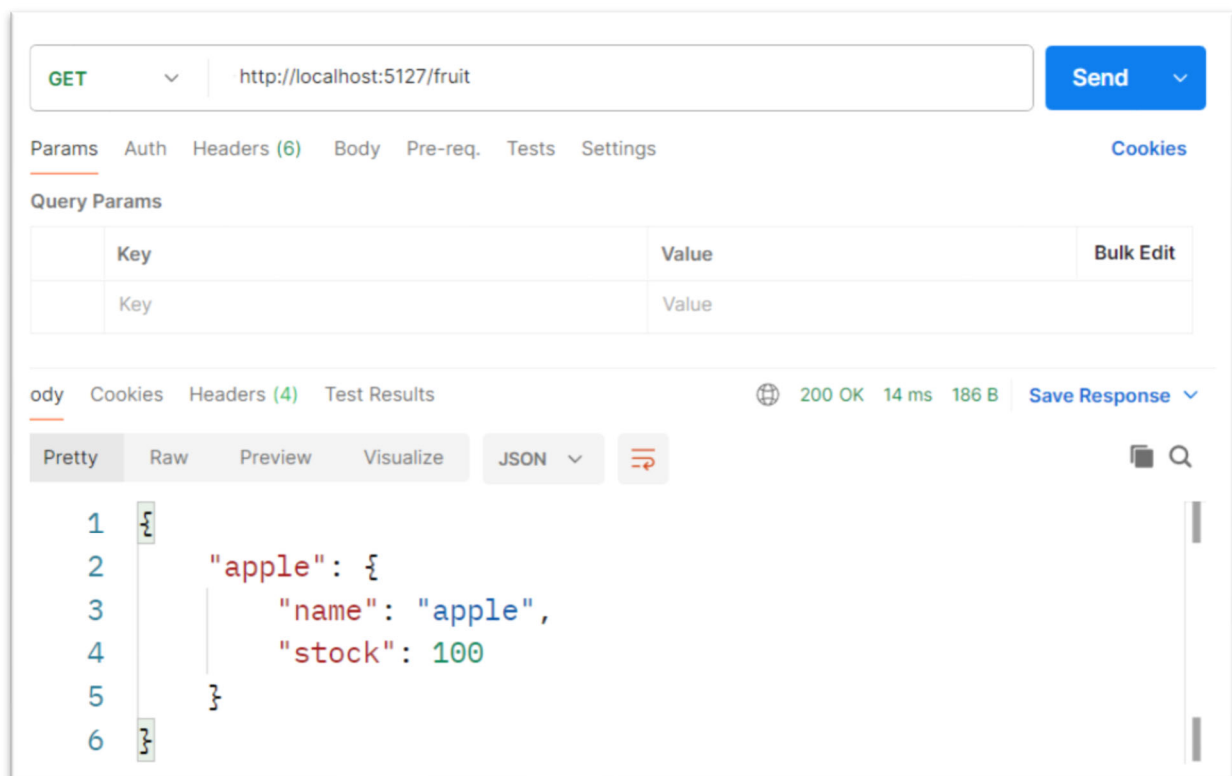
```
{ "Name": "apple", "Stock": 100 }
```



发送请求后，得到 200 响应。



再次查看所有 fruit 记录，可以看到刚刚添加的 apple 记录。



4. 修改 fruit 记录

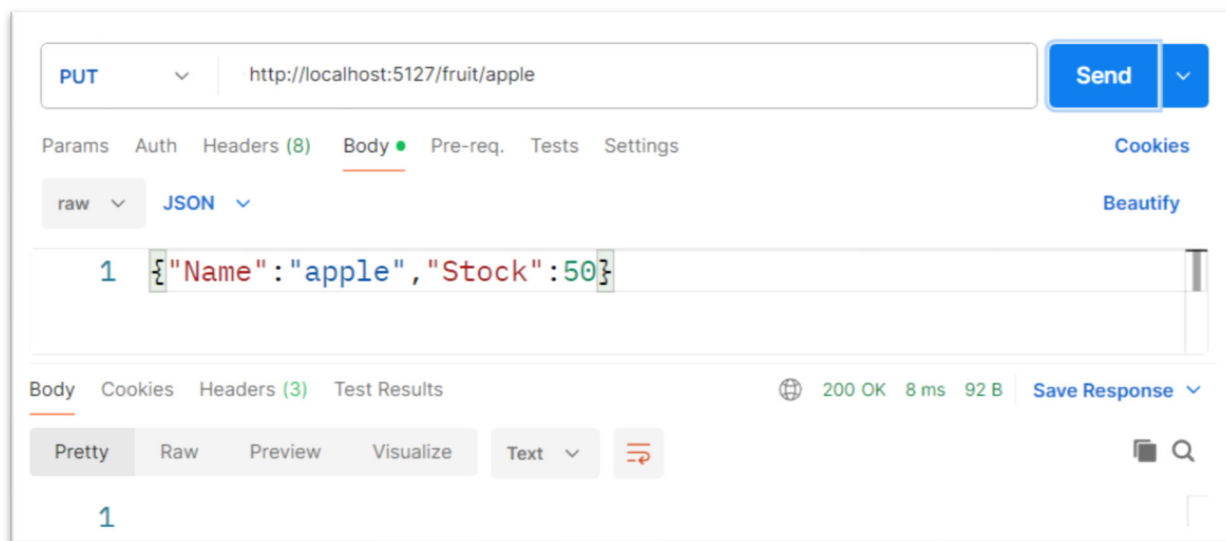
确保项目在运行状态。

修改 fruit 记录对应 `app.MapPut("/fruit/{id}", handlers.ReplaceFruit)` 端点。

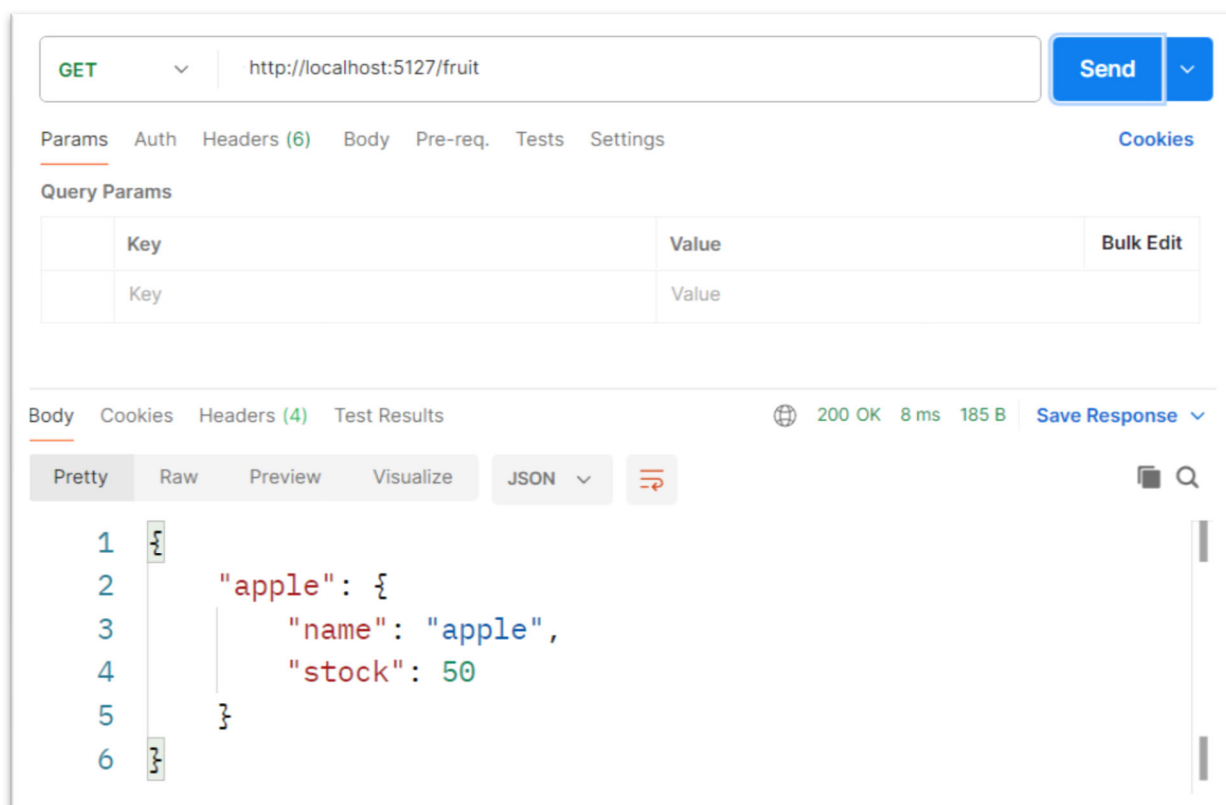
打开 postman，构建 PUT 请求。假设修改 id 为 apple 的记录，资源路径为 `/fruit/apple`。表示 fruit 的 `id` 为 apple，作为 `ReplaceFruit` 方法的 `id` 参数。

在请求的正文部分输入以下 JSON 数据，该数据将提交到服务器端，作为 `ReplaceFruit` 方法的 `fruit` 参数。

```
{"Name": "apple", "Stock": 50}
```



发送请求后得到 200 响应。再次查看所有 fruit 记录，可以看到 apple 记录的 stock 从 100 修改为 50。

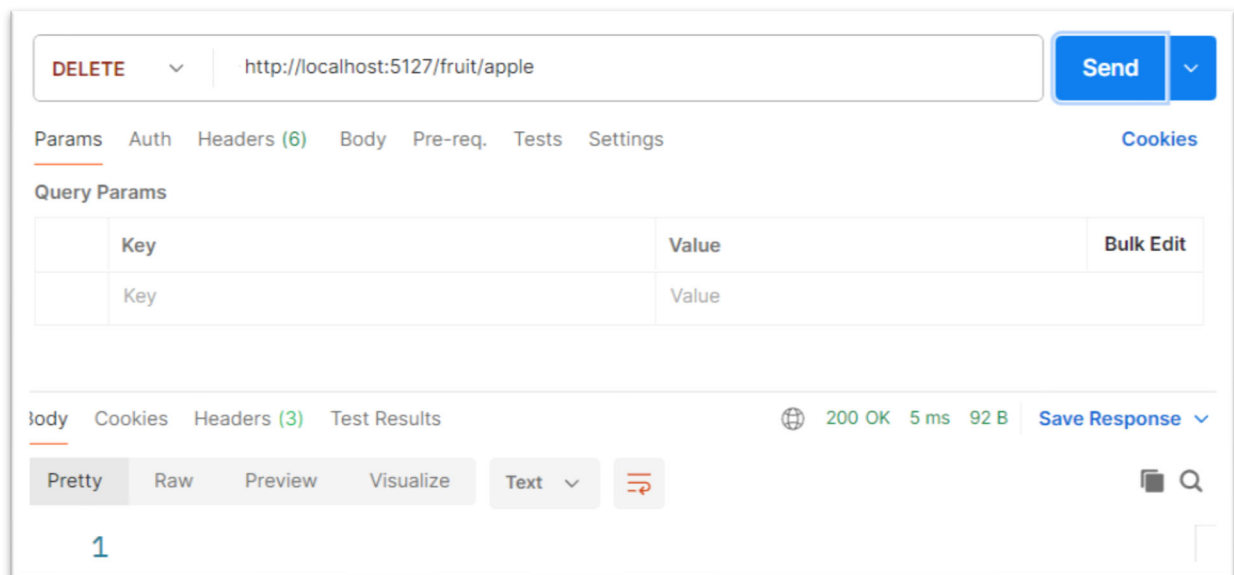


5. 删除 fruit 记录

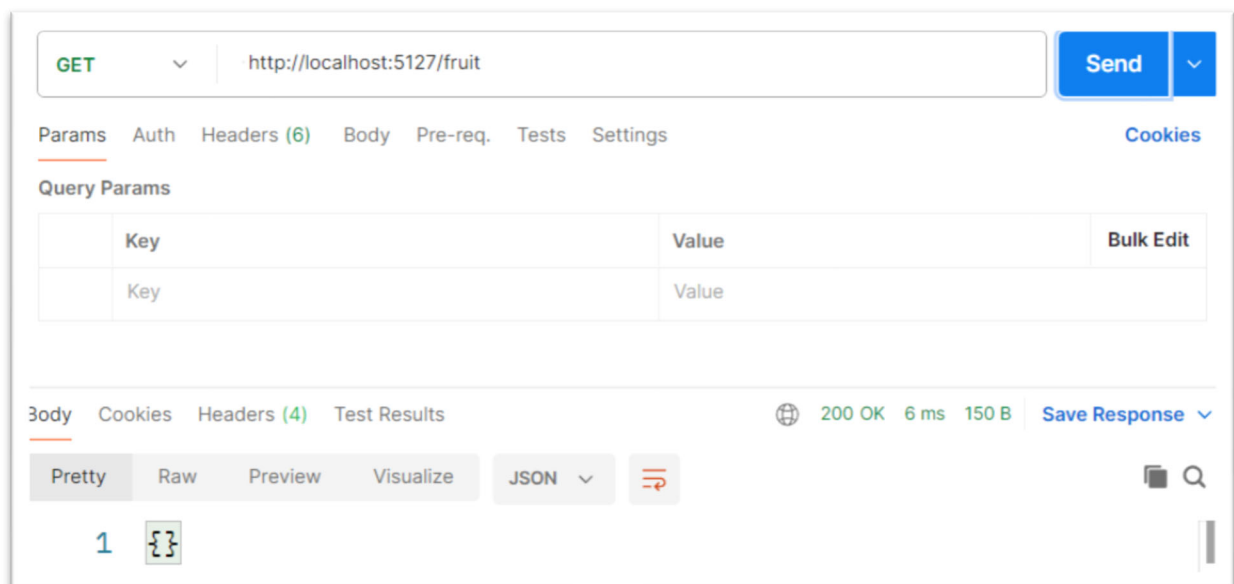
确保项目在运行状态。

删除 fruit 记录对应 `app.MapDelete("/fruit/{id}", DeleteFruit)` 端点。

打开 postman，构建 `DELETE` 请求。假设修改 id 为 apple 的记录，资源路径为 `/fruit/apple`。表示 fruit 的 id 为 apple，作为 `DeleteFruit` 方法的 `id` 参数。



发送请求后得到 200 响应。再次查看所有 fruit 记录，可以看到 fruit 记录为空，表示 apple 记录已删除。。



注意事项

构建 HTTP 请求时，确保 HTTP 方法、资源路径、请求正文部分数据格式正确。