

- 目标
 - 说明
 - 练习 1
 - 练习 2
 - 练习 3
- [GitHub Copilot 实验练习](#)

目标

目标是使用.NET 7.0创建一个最小化的WebAPI，并借助GitHub Copilot创建一个对应的Docker镜像。请按照下面的说明尝试使用GitHub Copilot。尝试不同的方法，看看GitHub Copilot能为你做什么，比如生成一个Dockerfile或一个类，添加注释等。

确保GitHub Copilot已配置并启用当前语言，只需检查VS Code右下角的状态栏。

说明

`dotnet`文件夹包含`MinimalAPI.sln`解决方案，其中有两个项目：

- `MinimalAPI` 是使用 `dotnet new webapi -minimal` 创建的最小 WebAPI 项目
- `MinimalAPI.Tests` 是使用 `dotnet new xunit` 创建的最小 xUnit 项目

要运行测试，请在`dotnet`文件夹中打开终端并运行：

```
dotnet test
```

要运行该应用程序，请在`dotnet`文件夹中打开终端并运行：

```
dotnet run --project .\MinimalAPI\MinimalAPI.csproj
```

练习 1

- 在`MinimalAPI\Program.cs`中添加一个新的Hello World端点，路径为`/`，返回一个`Hello World!`字符串。

- 运行 `dotnet test`
- 如果测试通过，你应该看到类似如下的内容：

```
Microsoft (R) Test Execution Command Line Tool Version 17.6.0 (x64)
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
Starting test execution, please wait...
```

```
A total of 1 test files matched the specified pattern.
```

```
Passed! - Failed: 0, Passed: 1, Skipped: 0, Total: 1, Duration: <
1 ms - MinimalAPI.Tests.dll
```

练习 2

- 在 `MinimalAPI\Program.cs` 中添加以下端点：
- **/DaysBetweenDates:**
 - 计算两个日期之间的天数
 - 通过查询字符串接收名为 `date1` 和 `date2` 的两个参数，并计算这两个日期之间的天数。
- **/validatephonenumber:**
 - 通过查询字符串接收名为 `phoneNumber` 的参数
 - 用西班牙格式验证 `phoneNumber`，例如 `+34666777888`
 - 如果 `phoneNumber` 有效则返回 `true`
- **/validatespanishdni:**
 - 通过查询字符串接收名为 `dni` 的参数
 - 计算 DNI 字母
 - 如果 DNI 有效则返回 `"valid"`
 - 如果 DNI 无效则返回 `"invalid"`

我们将创建自动化测试来检查功能是否正确实现。开发完成后，我们将使用 **Docker** 构建一个容器。

- **/returncolorcode:**
 - 通过查询字符串接收名为 `color` 的参数
 - 读取 `colors.json` 文件并返回 `rgba` 字段

- 从查询字符串获取color变量
- 遍历colors.json中的每种颜色，以找到该颜色
- 返回code.hex字段

- **/tellmeajoke:**

调用笑话API并使用axios返回一个随机笑话

- **/moviesbydirector:**

(这将需要浏览<https://www.omdbapi.com/apikey.aspx>并请求一个免费的API密钥)

通过查询字符串接收名为 **director** 的参数

调用电影API，并使用axios返回该导演的电影列表

返回完整的电影列表

- **/parseurl:**

从查询字符串获取名为 **someurl** 的参数

解析URL并返回协议、主机、端口、路径、查询字符串和哈希

返回解析后的主机

- **/listfiles:**

获取当前目录

获取当前目录中的文件列表

返回文件列表

- **/calculatememoryconsumption:**

返回进程的内存消耗（以GB为单位，保留2位小数）

- **/randomeuropeancountry:**

制作一个欧洲国家及其ISO代码的数组

从数组中返回一个随机国家

返回国家及其ISO代码

练习 3

- 为最小API项目创建一个Dockerfile。
- 构建镜像并将应用程序运行在端口8080上

```
docker build -t dotnetapp .  
docker run -d -p 8080:80 --name dotnetapp dotnetapp
```

GitHub Copilot 实验练习

可以使用 Copilot 实验室插件执行以下任务，这是当前的预览功能，可能会出现一些错误。

请确保安装了GitHub Copilot实验室扩展：<https://marketplace.visualstudio.com/items?itemName=GitHub.copilot-labs>

打开GitHub Copilot扩展以查看所有可用功能。

- 解释

选择`validatePhoneNumber`方法中具有正则表达式的行，在"解释"部分点击"询问Copilot"。您将看到一条详细说明，说明正则表达式中的每个不同符号的含义。

- 语言翻译

选择一些源代码：

```
var countries = new[] { "Spain", "France", "Germany", "Italy", "Portugal",  
    "Sweden", "Norway", "Denmark", "Finland", "Iceland", "Ireland", "United Kingdom",  
    "Greece", "Austria", "Belgium", "Bulgaria", "Croatia", "Cyprus", "Czech Republic",  
    "Estonia", "Hungary", "Latvia", "Lithuania", "Luxembourg", "Malta", "Netherlands",  
    "Poland", "Romania", "Slovakia", "Slovenia" };  
    return countries[new Random().Next(0, countries.Length)];
```

然后在"语言翻译"部分选择python并点击"询问Copilot"按钮，您应该看到新的python代码。

- 可读性

选择MakeZipFile的内容

在"**BRUSHES**"部分，点击"可读性"，看看如何添加注释以及将拥有简短名称的变量重命名为更容易理解的名称。