

- Spring Boot REST API 练习
 - 目标
 - 练习
 - 1. 创建处理简单的 **GET** 请求的代码
 - 2. 比较日期
 - 3. 验证西班牙电话号码的格式
 - 4. 验证西班牙身份证号（DNI）的格式
 - 5. 从颜色名称到十六进制代码
 - 6. 笑话创建者
 - 7. URL 解析
 - 8. 列出文件和文件夹
 - 9. 单词计数
 - 10. 压缩
 - 11. 将应用程序容器化

Spring Boot REST API 练习

目标

这个练习的目标是学习如何使用 GitHub Copilot，使用一个构建 Spring Boot REST API 的练习。

练习

我们已经创建了一个 Spring Boot 项目，并已经创建了一些文件，你可以在文件夹 **exercisefiles/springboot** 中找到这个项目。

让我们开始使用 Copilot 一起编程吧！

1. 创建处理简单的 **GET** 请求的代码

移动到 'DemoController.java' 文件中，开始编写处理简单的 **GET** 请求的代码。在这个第一个练习中，我们提供了一个注释，描述了你需要生成的代码。只需按下回车键，等待几秒钟，Copilot 就会为你生成代码。

这个练习已经有一个已实现的单元测试，你可以在生成的代码通过之前和之后使用命令 `mvn test` 运行这个单元测试，以验证 Copilot 生成的代码是否正确。

然后，为请求中未提供密钥的情况创建一个新的单元测试。

每个练习之后，随时可以打包并运行你的应用来测试它。

打包： `mvn package`

运行： `mvn spring-boot:run`

测试： `curl -v http://localhost:8080/hello?key=world`

2. 比较日期

在 `/diffdates` 下创建一个新的操作，计算两个日期之间的差值。该操作应接收两个以 `dd-MM-yyyy` 格式的日期作为参数，并返回以天为单位的差值。

此外，创建一个验证该操作的单元测试。

从现在开始，你需要为每个新的操作创建单元测试。使用 Copilot 难道不是很容易吗？

3. 验证西班牙电话号码的格式

验证西班牙电话号码的格式（+34 前缀，然后是以 6、7 或 9 开头的9位数字）。该操作应接收一个电话号码作为参数，并在格式正确时返回 `true`，否则返回 `false`。

4. 验证西班牙身份证号（DNI）的格式

验证西班牙身份证号（8位数字和1位字母）的格式。该操作应接收一个身份证号作为参数，并在格式正确时返回 `true`，否则返回 `false`。

5. 从颜色名称到十六进制代码

根据资源目录下的现有 `colors.json` 文件，给定颜色名作为路径参数，返回十六进制代码。如果找不到颜色，返回 404。

提示：使用测试驱动开发（TDD）。先创建单元测试，然后才实现代码。

6. 笑话创建者

创建一个调用 API `https://api.chucknorris.io/jokes/random` 并返回笑话的新操作。

7. URL 解析

给定一个作为查询参数的 **URL**，解析它并返回协议、主机、端口、路径和查询参数。响应应以 **Json** 格式返回。

8. 列出文件和文件夹

列出给定路径下的文件和文件夹。路径应作为查询参数。响应应以 **Json** 格式返回。

9. 单词计数

给定文件的路径并计算提供的单词出现的次数。路径和单词应作为查询参数。响应应以 **Json** 格式返回。

10. 压缩

创建一个包含给定文件夹内容的**zip**文件。文件夹的路径应作为查询参数。

11. 将应用程序容器化

使用提供的 **Dockerfile** 创建应用程序的 **Docker** 镜像。**Dockerfile** 中有一些注释，可以帮助你完成这个练习。

为了构建、运行和测试 **Docker** 镜像，你也可以使用 **Copilot** 生成相应的命令。

例如，创建一个 **DOCKER.md** 文件，你可以在其中存储构建、运行和测试 **Docker** 镜像的命令。你会注意到 **Copilot** 还认为帮助你记录项目和命令。

文档步骤示例：构建容器镜像、运行容器、测试容器。