# How to create a SpringBoot Web API in VSCode

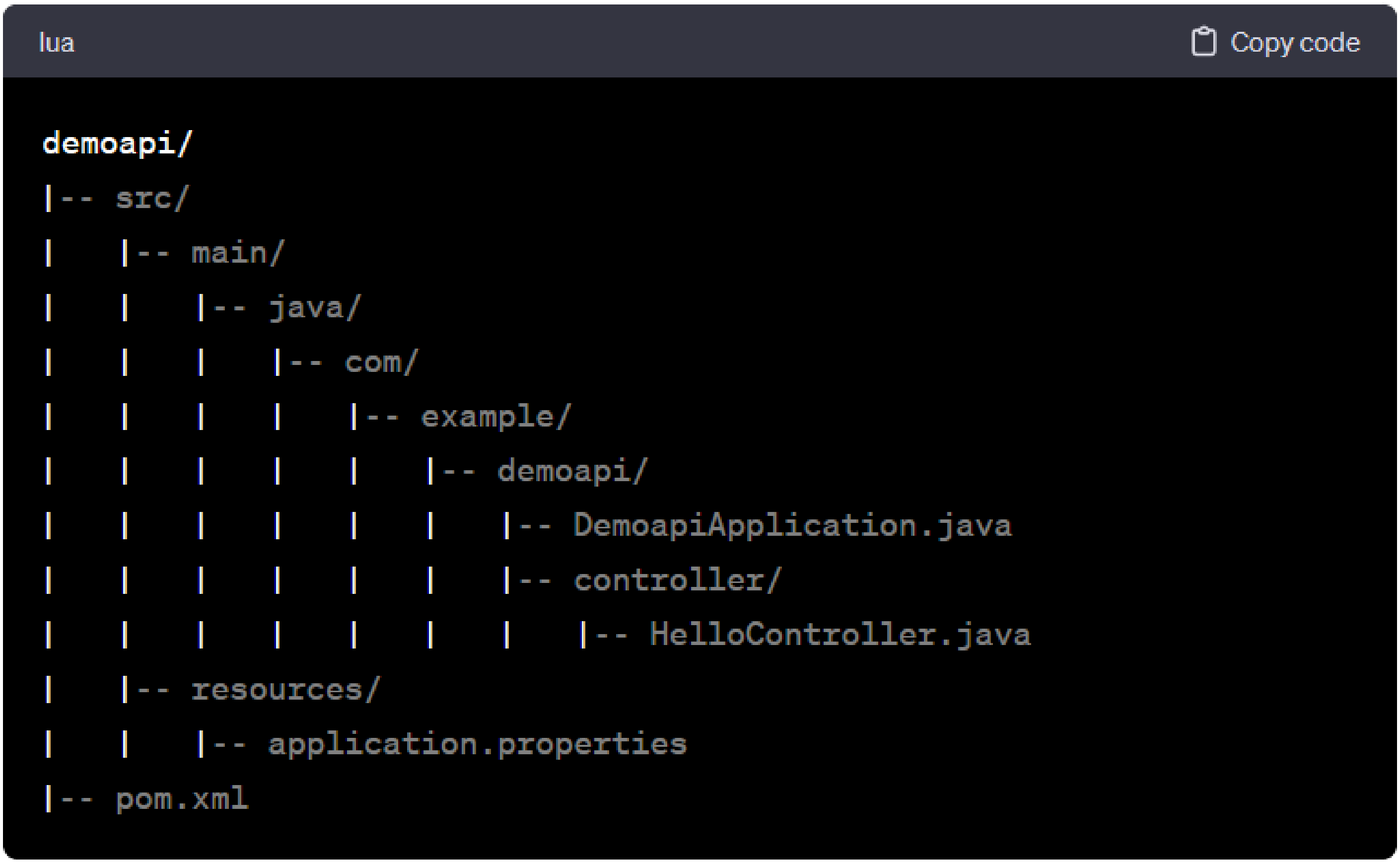
You can find this example source code in this github repo:
https://github.com/luiscoco/SpringBoot_Sample2-created-WebAPI-with-VSCode

## 1. Set up the environment

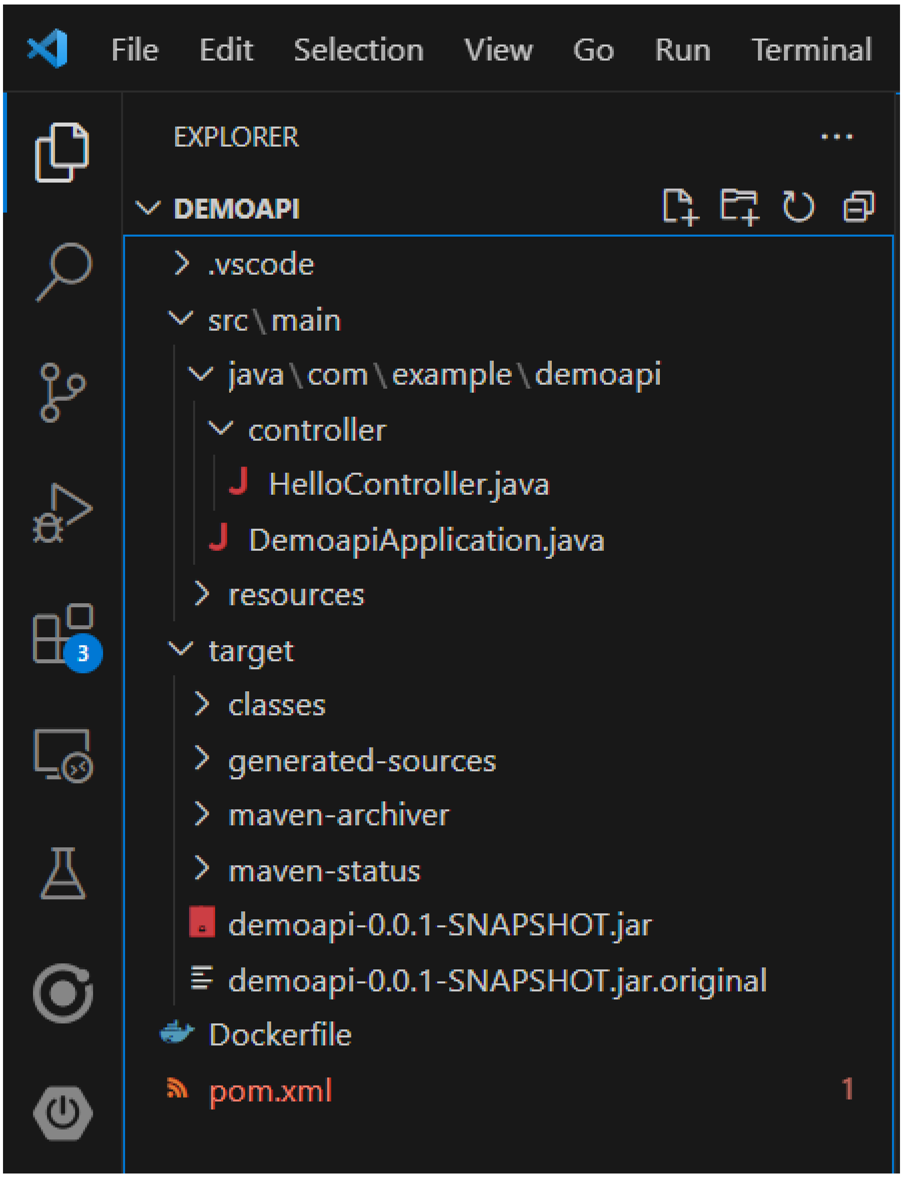
First, ensure you have the following setup in your VS Code environment:

- Java Development Kit (**JDK**)
- **VS Code** with the Java extension pack installed
- **Maven** for dependency management

## 2. Project folders structure



```
demoapi/
|-- src/
|   |-- main/
|   |   |-- java/
|   |   |   |-- com/
|   |   |   |   |-- example/
|   |   |   |   |   |-- demoapi/
|   |   |   |   |   |   |-- DemoapiApplication.java
|   |   |   |   |   |   |-- controller/
|   |   |   |   |   |   |   |-- HelloController.java
|   |-- resources/
|   |   |-- application.properties
|-- pom.xml
```



DEMOAPI
- .vscode
- src\main
  - java\com\example\demoapi
    - controller
      - HelloController.java
    - DemoapiApplication.java
  - resources
- target
  - classes
  - generated-sources
  - maven-archiver
  - maven-status
  - demoapi-0.0.1-SNAPSHOT.jar
  - demoapi-0.0.1-SNAPSHOT.jar.original
- Dockerfile
- pom.xml

## 3. Here's a simple Java Spring Boot application source code

### pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mav
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>demoapi</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>demoapi</name>
    <description>Demo project for Spring Boot</description>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.5.0</version>
        <relativePath/>
    </parent>

    <dependencies>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-actuator</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

### DemoapiApplication.java

```
package com.example.demoapi;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class DemoapiApplication {
    public static void main(String[] args) {
        SpringApplication.run(DemoapiApplication.class, args);
    }
}
```

### HelloController.java

http://localhost:8080/hello

```
package com.example.demoapi.controller;

import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HelloController {
    @GetMapping("/hello")
    public String sayHello() {
        return "Hello, World!";
    }
}
```

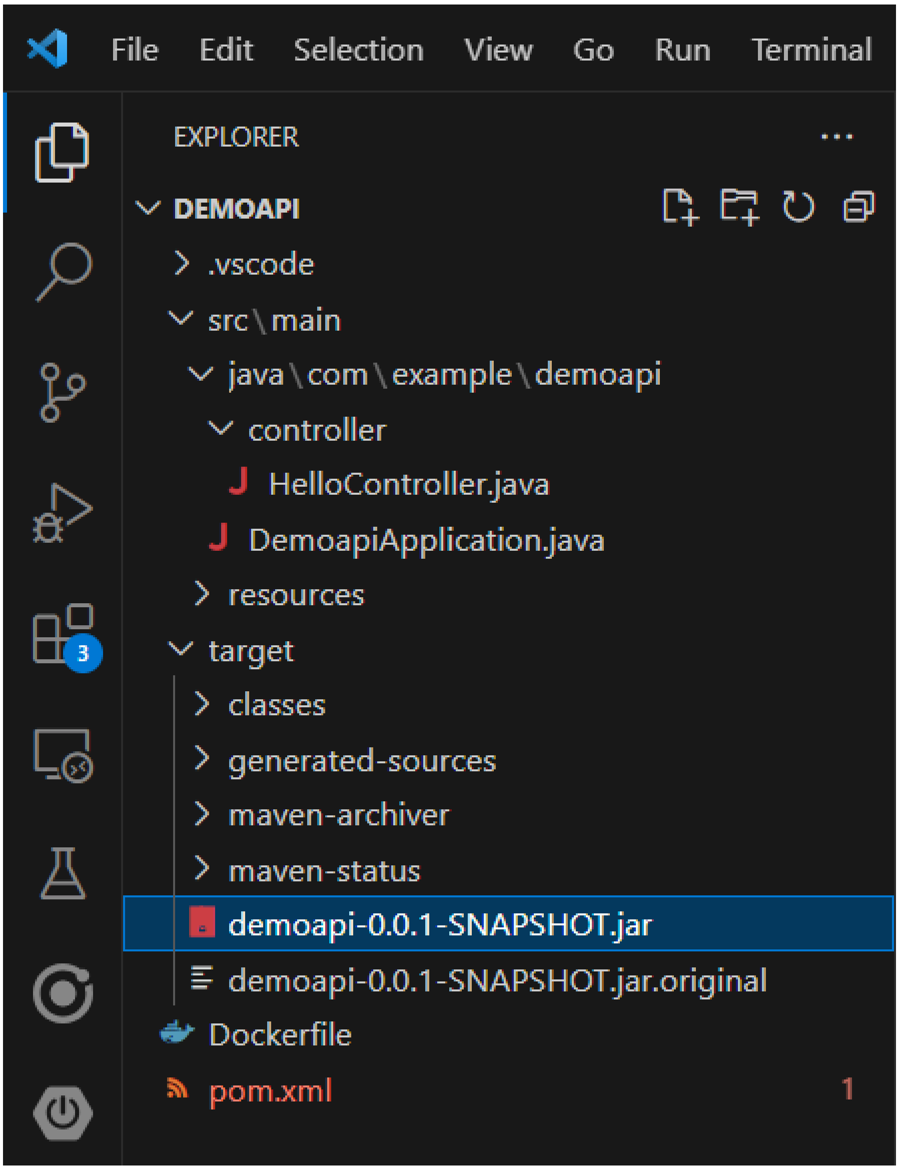
## 4. Dockerfile

```
# Start with a base image containing Java runtime
FROM openjdk:11-jdk-slim as build

# Add Maintainer Info
LABEL maintainer="your_email@example.com"

# Add a volume pointing to /tmp
VOLUME /tmp

# Make port 8080 available to the world outside this container
EXPOSE 8080

# The application's jar file
ARG JAR_FILE=target/demoapi-0.0.1-SNAPSHOT.jar

# Add the application's jar to the container
ADD ${JAR_FILE} demoapi.jar

# Run the jar file
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/demoapi.jar"]
```

## 5. To build and package the application

To build the application execture this command in the VSCode terminal window:

```
mvn clean install
```



```
DEMOAPI
  .vscode
  src\main
    java\com\example\demoapi
      controller
        HelloController.java
      DemoapiApplication.java
    resources
  target
    classes
    generated-sources
    maven-archiver
    maven-status
    demoapi-0.0.1-SNAPSHOT.jar
    demoapi-0.0.1-SNAPSHOT.jar.original
  Dockerfile
  pom.xml                                  1
```

To run the application run this command:

```
java -jar target/demoapi-0.0.1-SNAPSHOT.jar
```

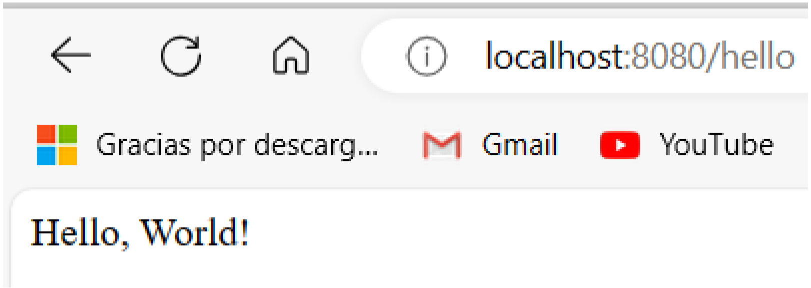
## 6. How build the Docker image and run it

To build the Docker image execute this command:

```
docker build -t demoapi .
```

To run the Web API docker image run this command:

```
docker run -p 8080:8080 demoapi
```



Hello, World!

## 7. How to create this example with ChatGPT-4

- Please provide me a Java API code sample in VSCode
- Please give me the whole application in different files
- Is it possible you compress the above files in a zip file to download them?
- Please tell me the command to execute this project in VSCode
- Can you provide me the Dockerfile to create a docker image for the above application API?
- Please provide me the command to run the docker image and access to the endpoint in my internet web browser