**Day 3**

**Creating our first microservices**

Go to <https://start.spring.io/>

Select:

Project: Maven

Language: Java,

Springboot version: 3.310

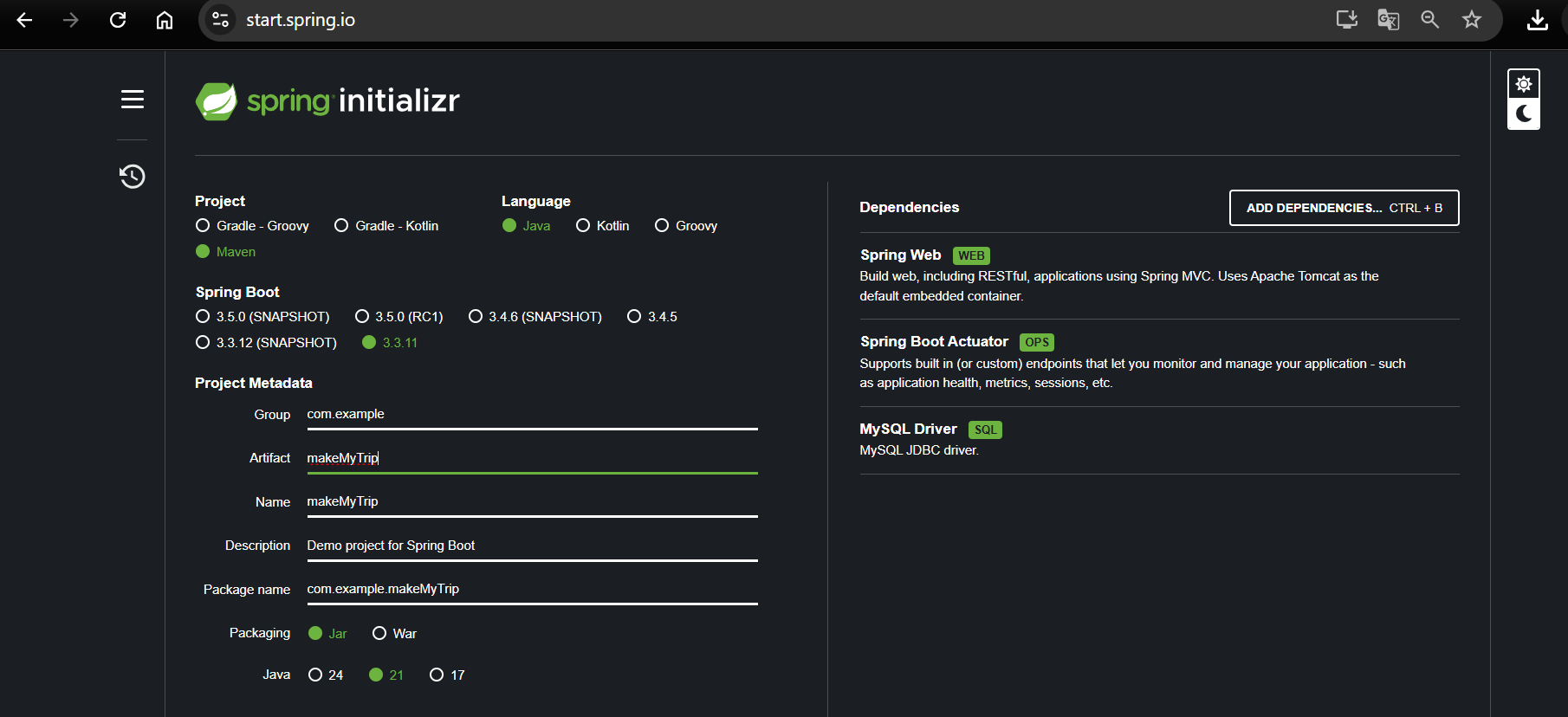
Artifact name: makemytrip

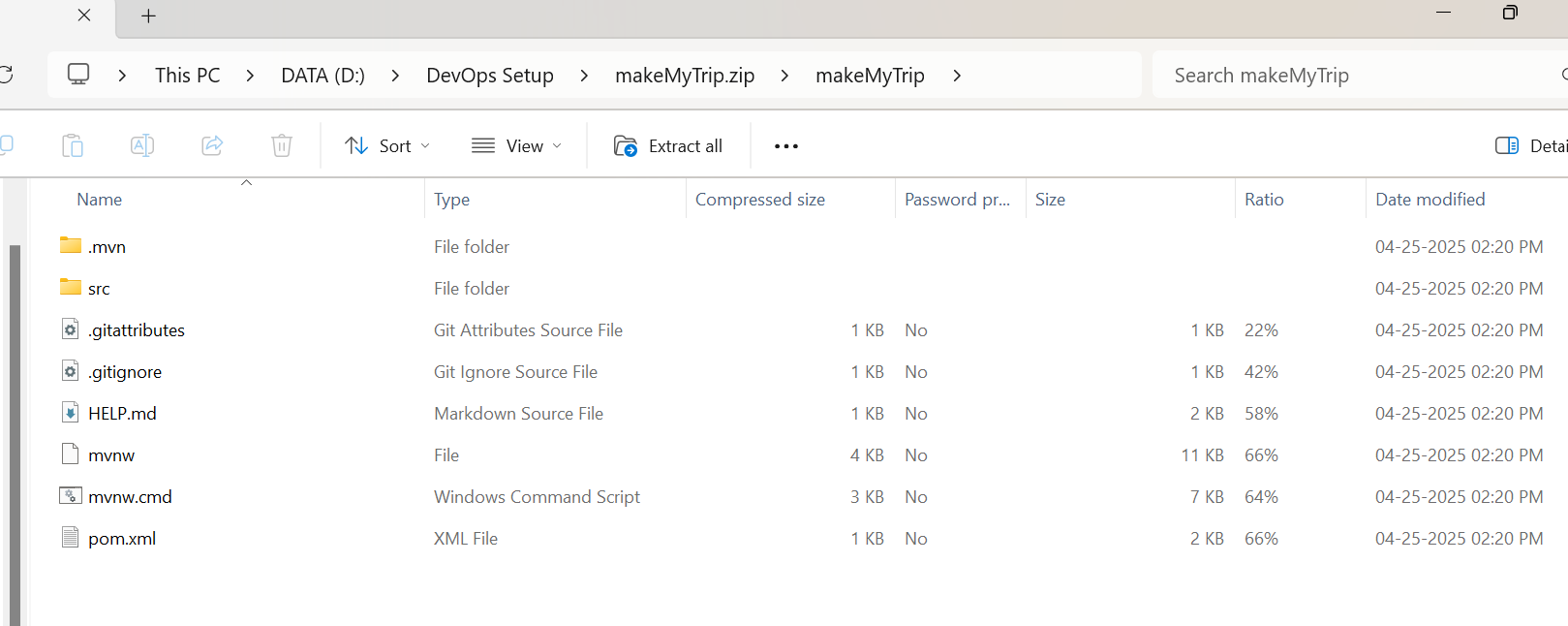
Packaging: Jar

Add the dependencies: Spring Web, String Boot Actuator

Generate the zip file

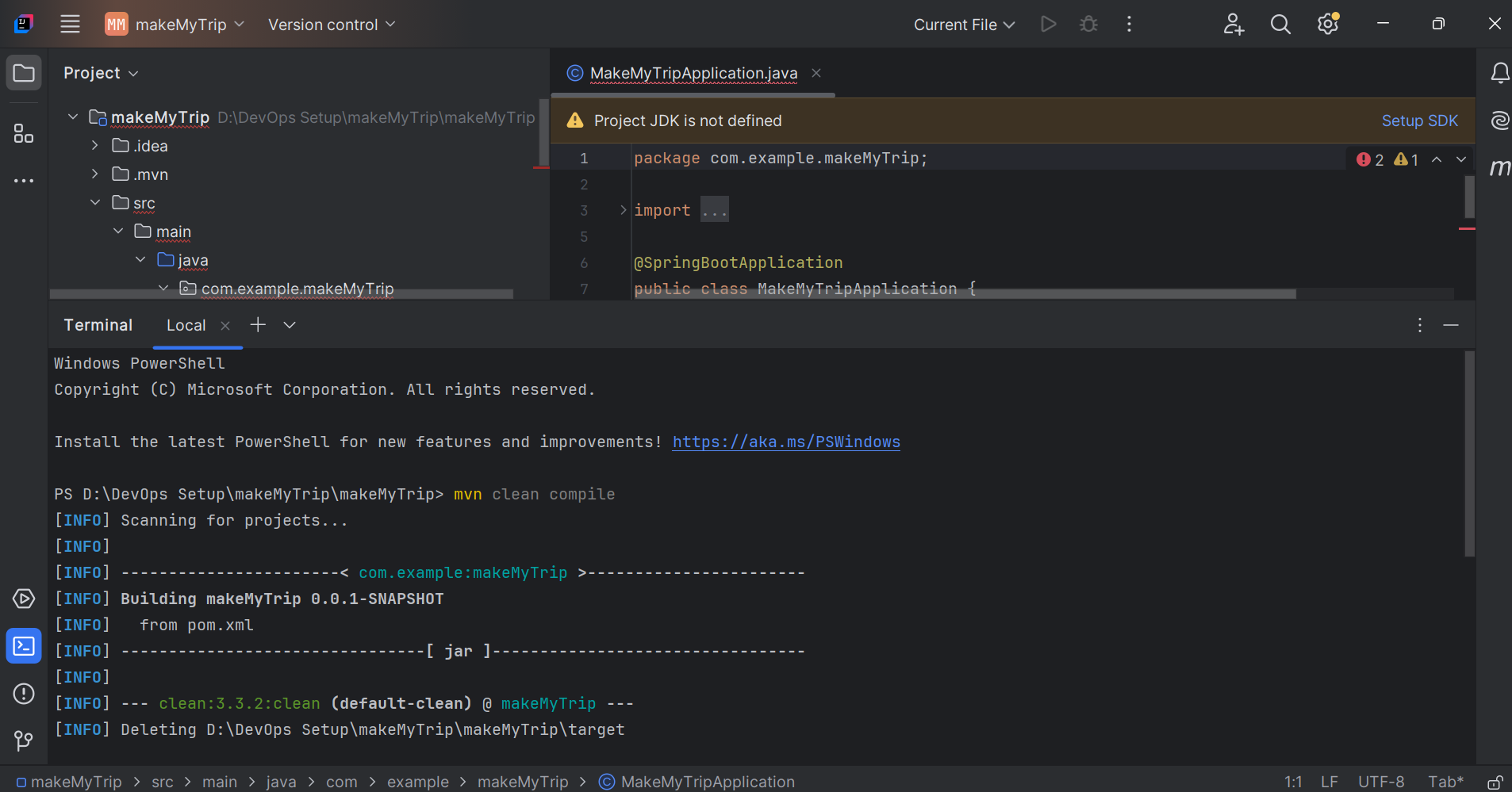
Extract the zip file



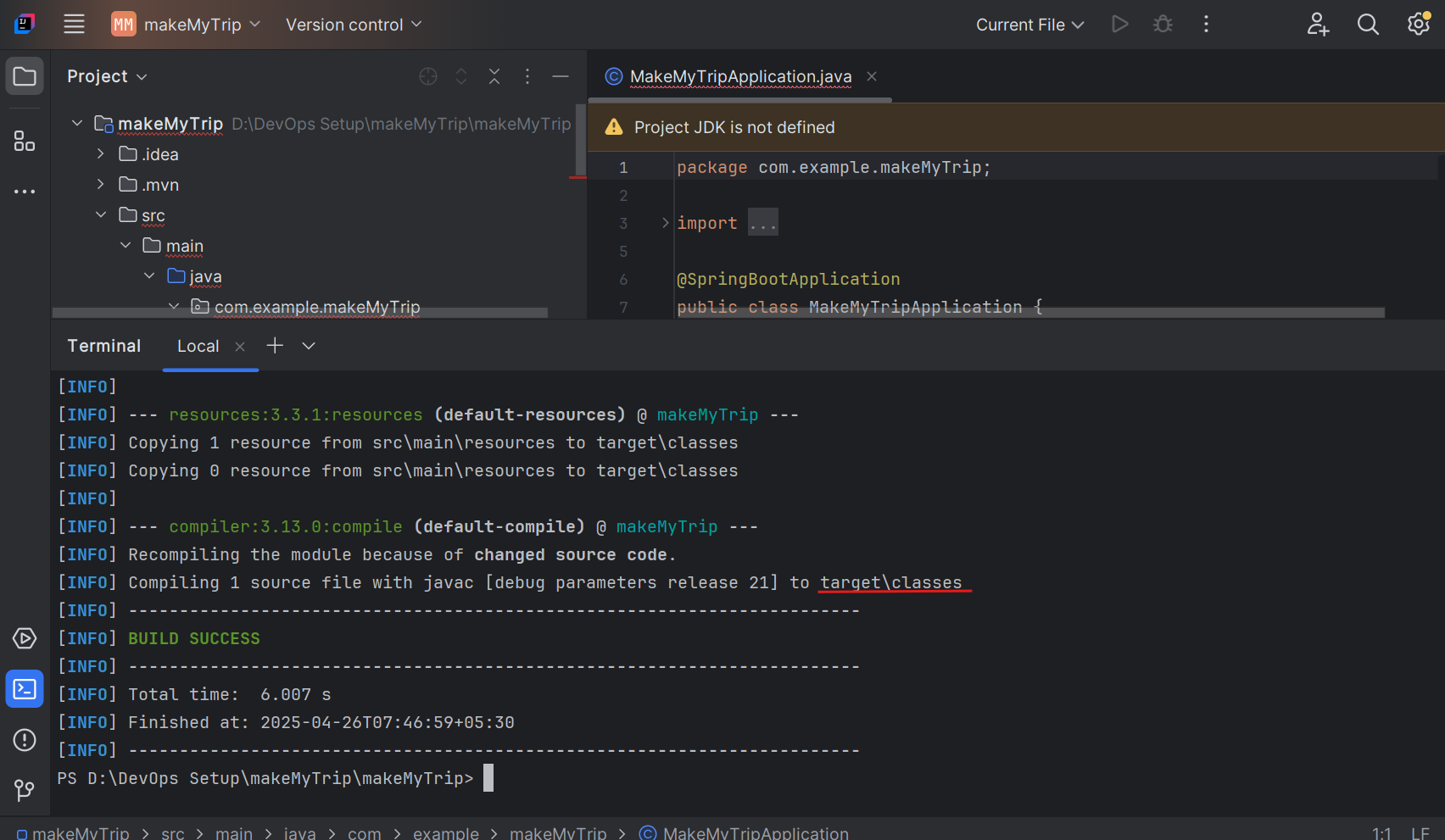


> mvn clean compile

(Creates .class file)



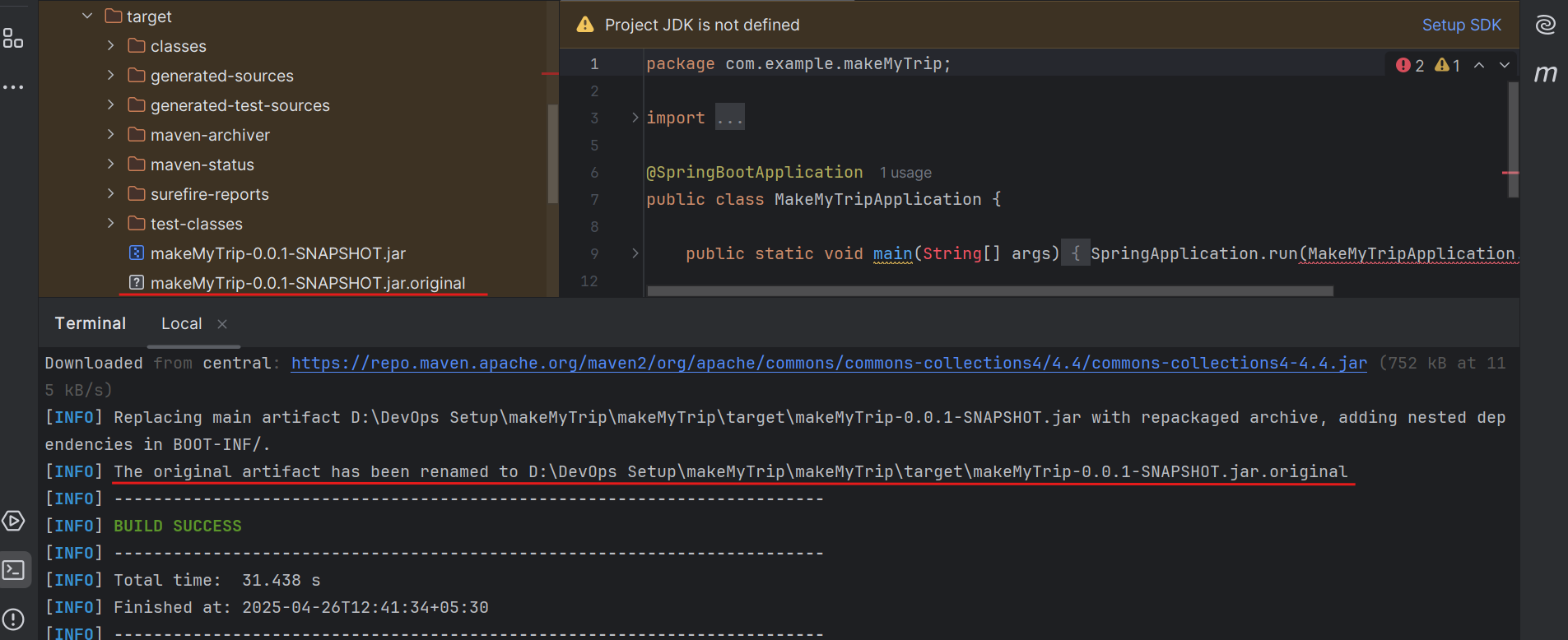
Location of .class file



Now package:

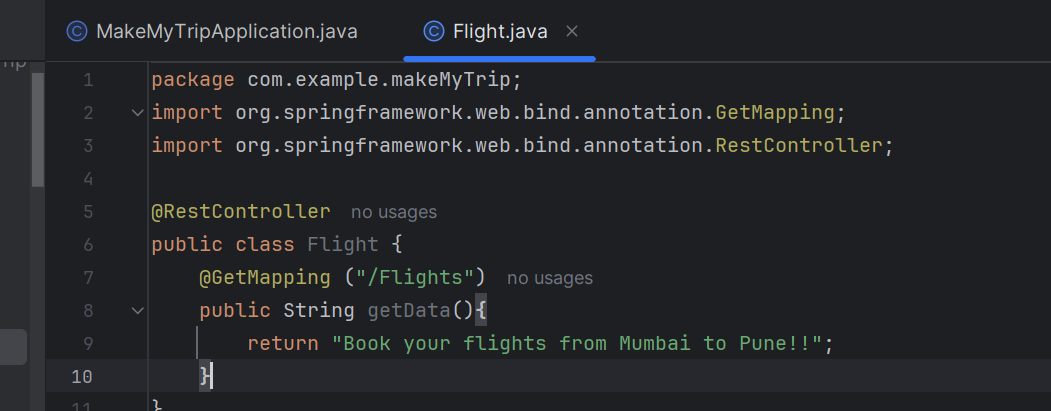
> mvn clean package : - Compiles your code, runs tests, and then packages the compiled code into a distributable format (like a JAR or WAR file), based on your project's configuration.

Default package is .jar



Now add a java file Right click on java > new > java class > flights

And add below code to this file flights.java:

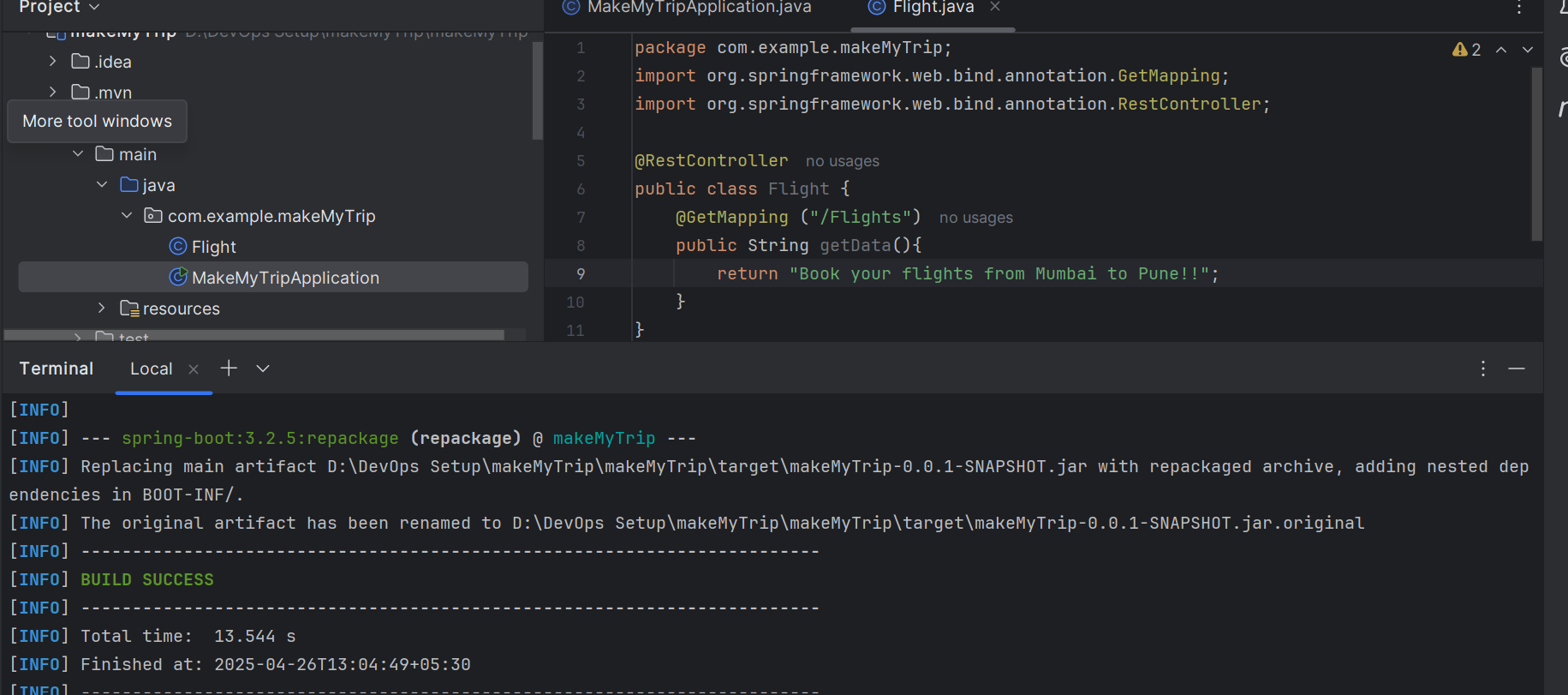


After ading new code - you need to compile and package it again.

> mvn clean compile

> mvn clean package

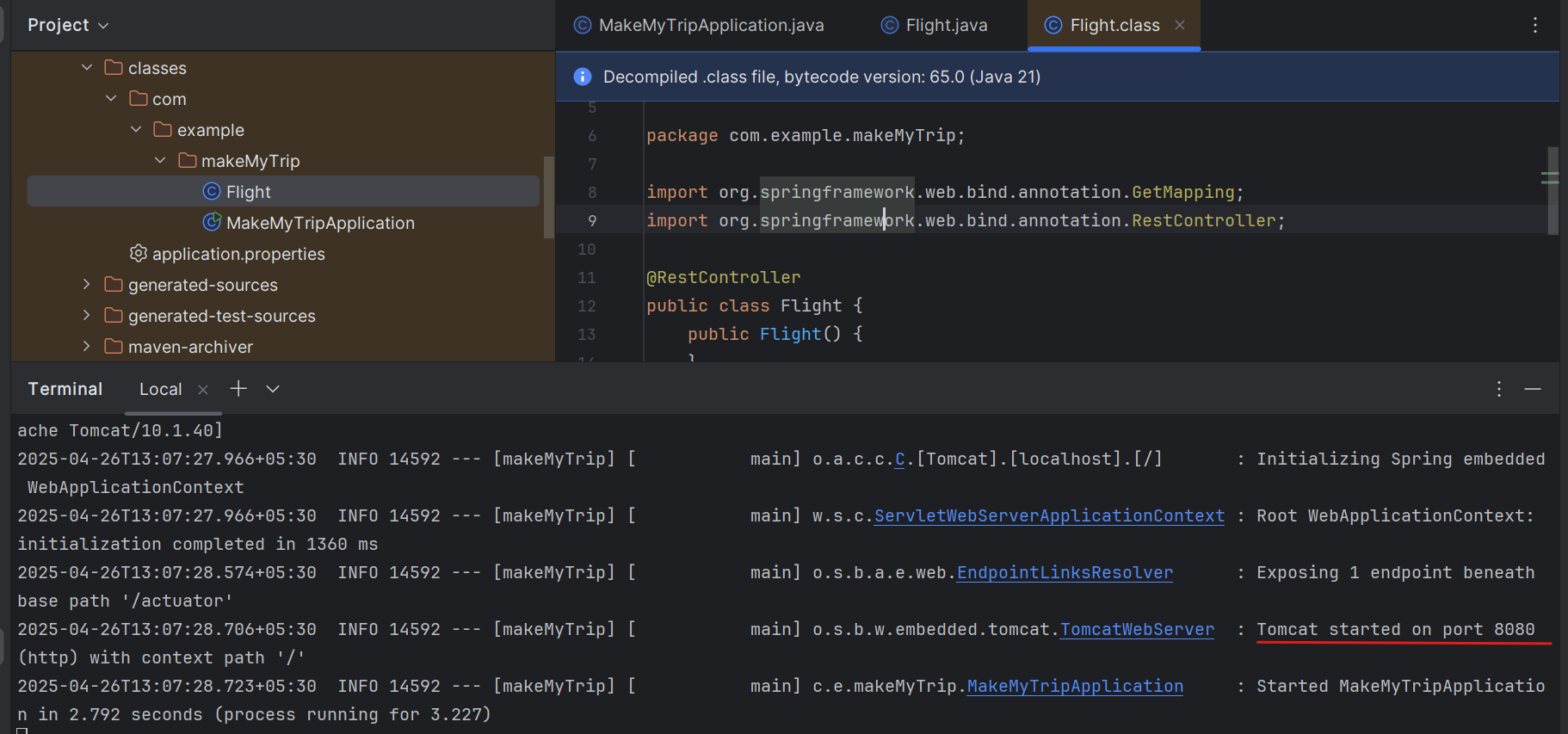
.class and .jar file should be generated



> mvn spring-boot:run

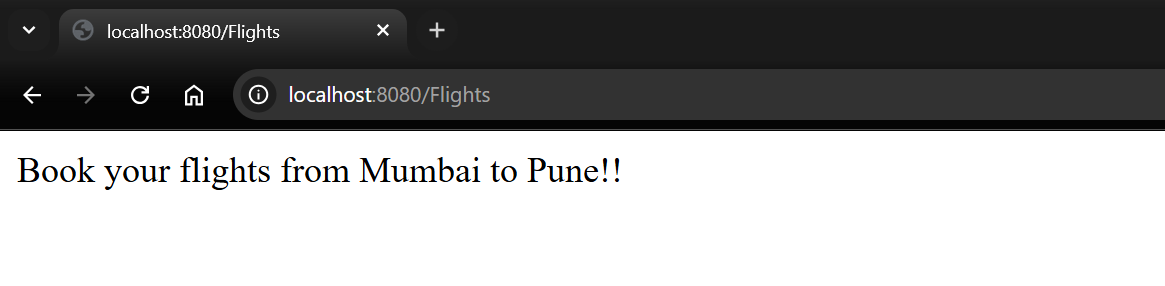
With help of this we will run the application on local system.

We can see the application is running on port 8080

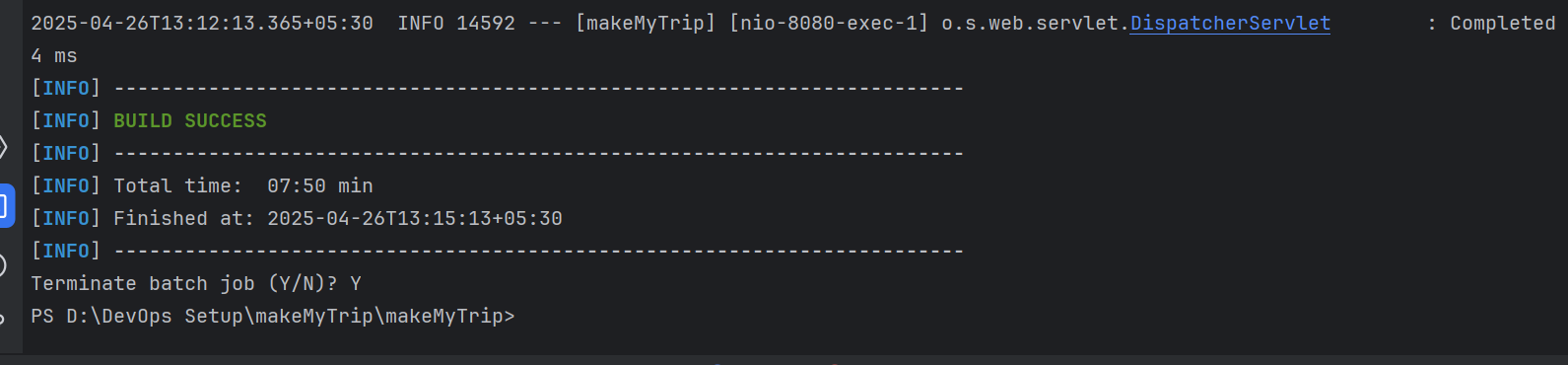


Now run the flights microservice on localhost

<http://localhost:8080/flights>



Press control C on terminal and the application will stop. If you check the link again it wont work anymore



Add new feature now: hotels.java

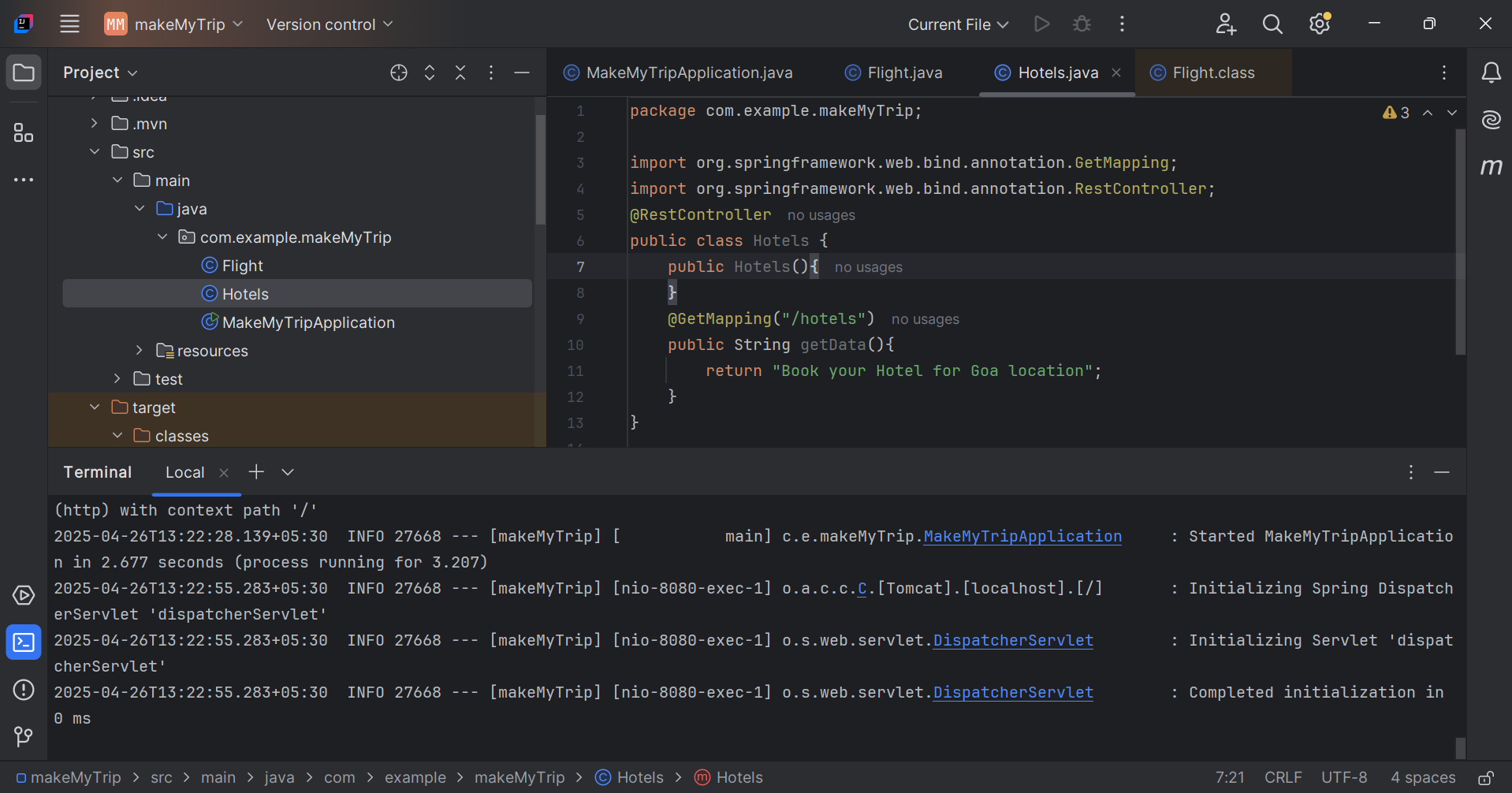
Now clean compile and package to make sure the new code is not spoiling the old working code.

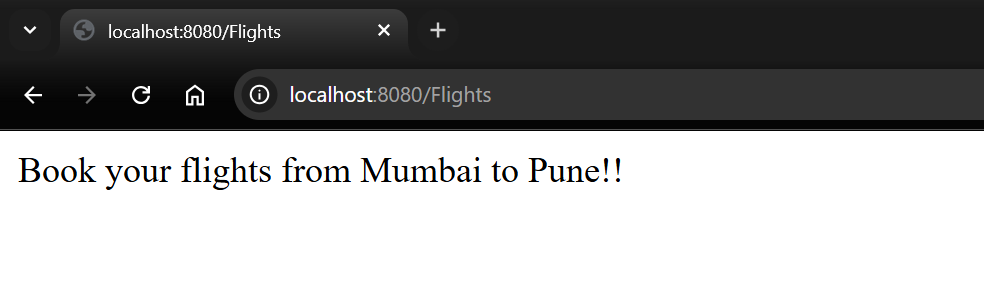
> mvn clean compile

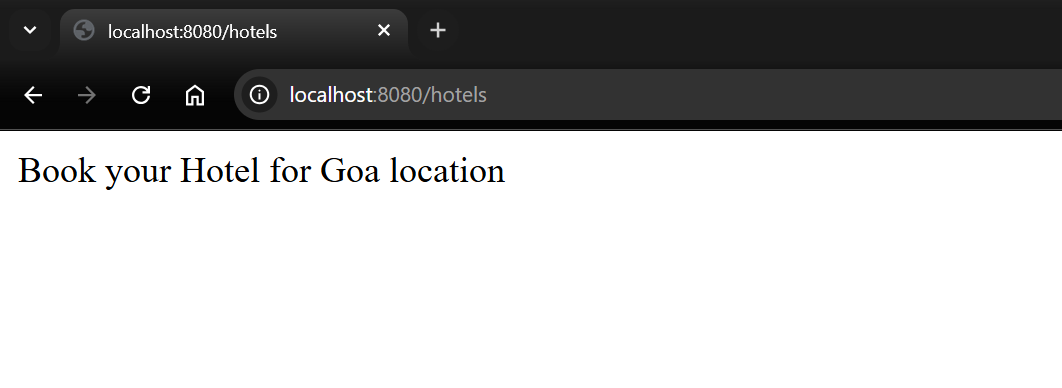
> mvn clean package

And then verify both the applications: hotels and flights

> mvn spring-boot:run



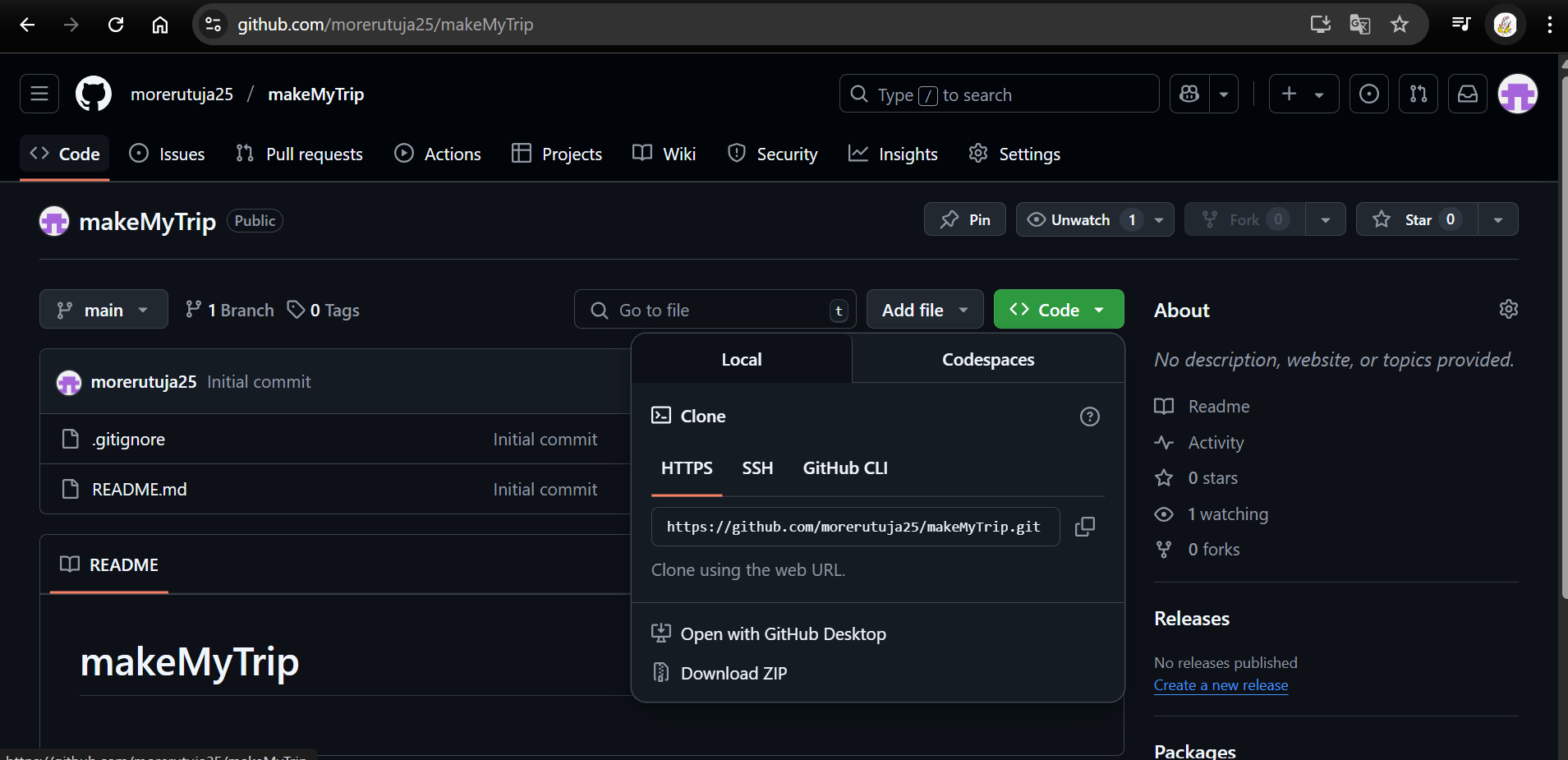




So we can see both features are working on local system.

Once the functionalities are verified and code is working - the code is supposed to be pushed on **github - shared remote code repository**.

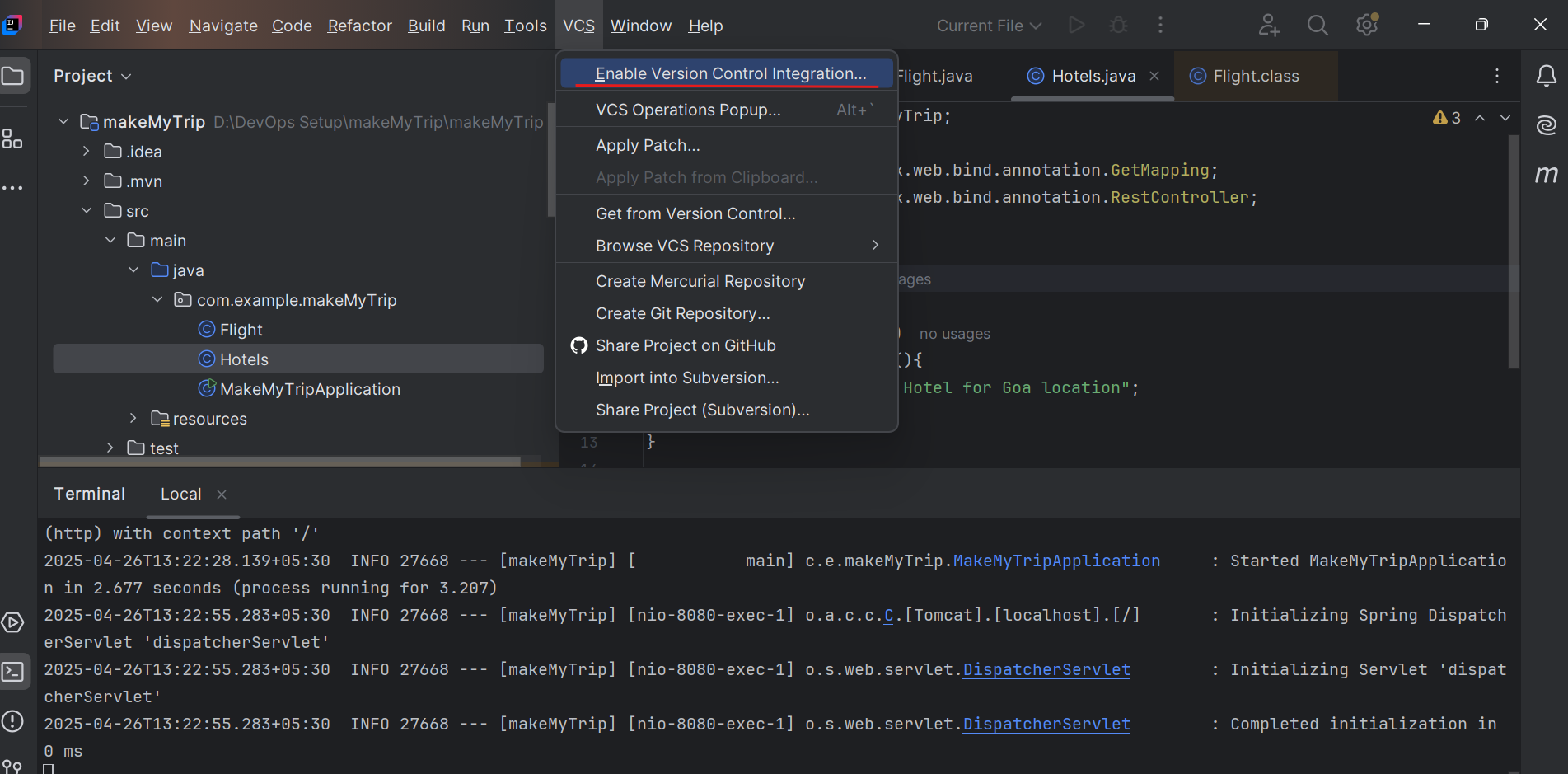
Create new repository on Github: makeMyTrip



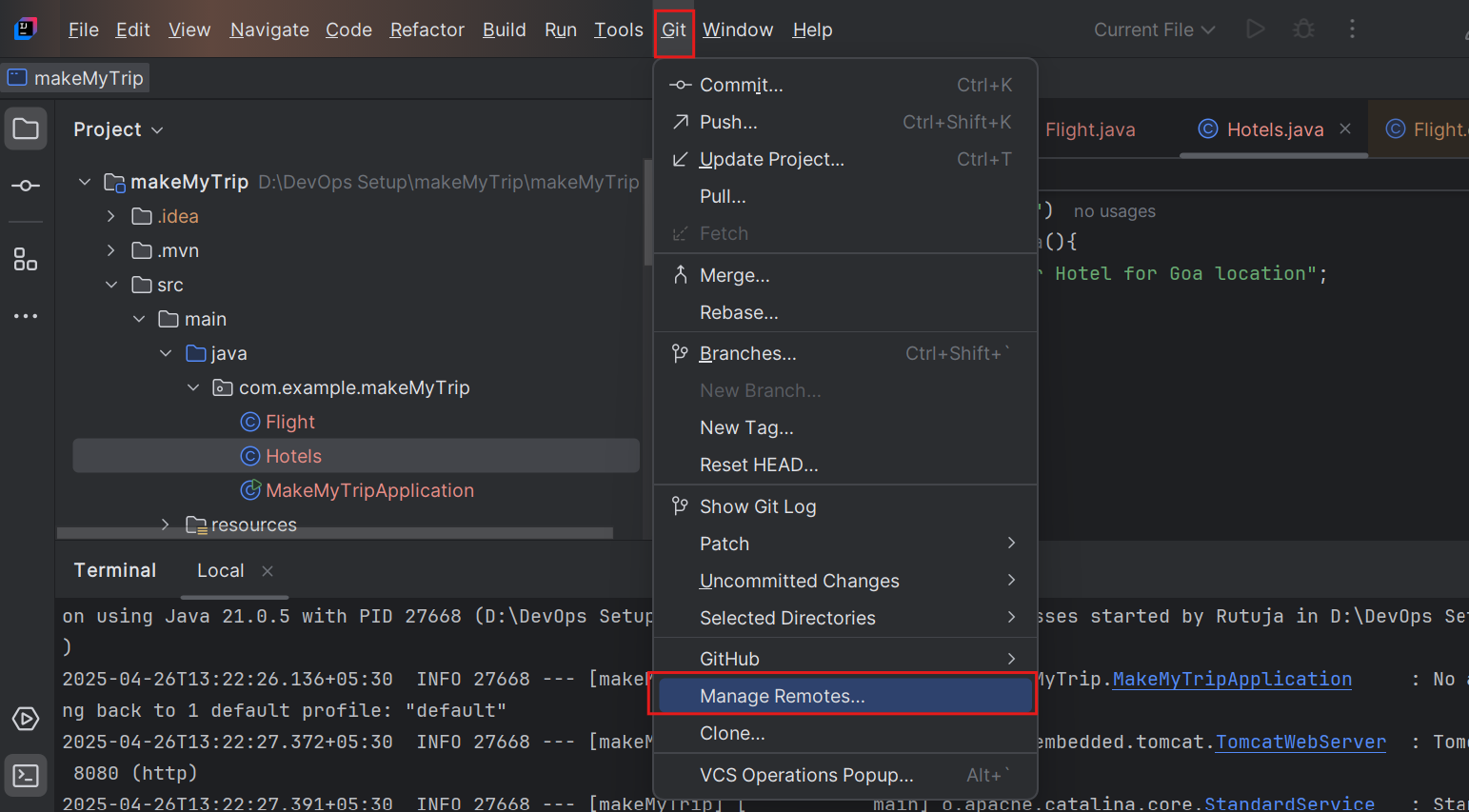
Push the code from local to remote repository

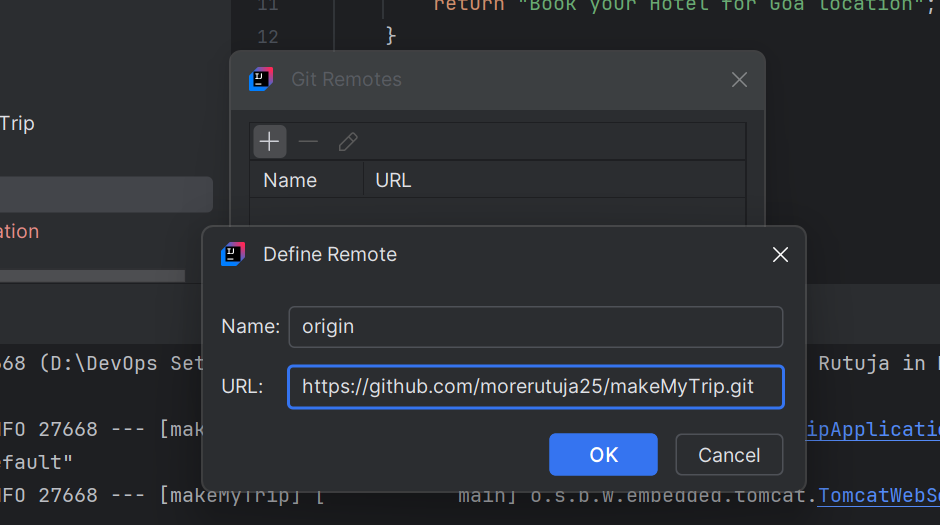
In IntelliJ IDEA enable version control integration:

VCS > Enable version control integration



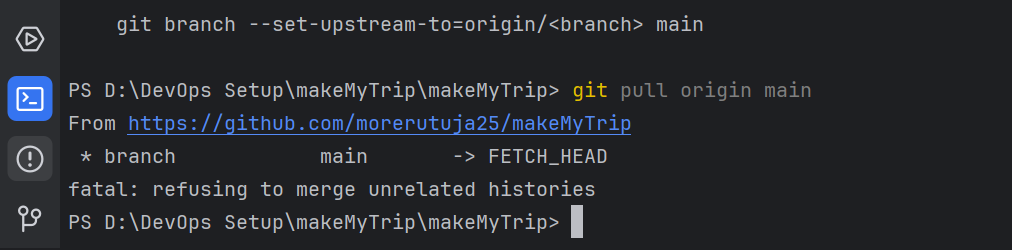
VCS > Manage remote > Enter URL of remote repository just copied





In my case - my default branch name is main and not master

> git pull origin main



1st time you run this:

git pull

git pull origin master

git add --all

git commit -m "CodeRefactor"

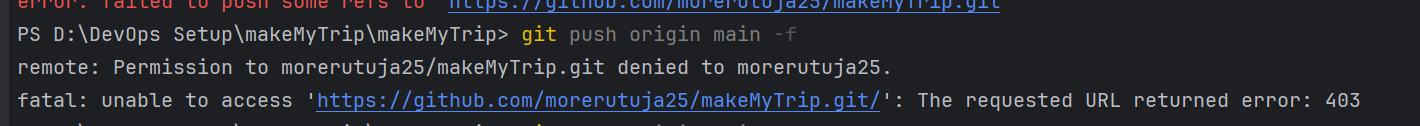
git push origin master -f

2nd and all times

git add --all

git commit -m "message"

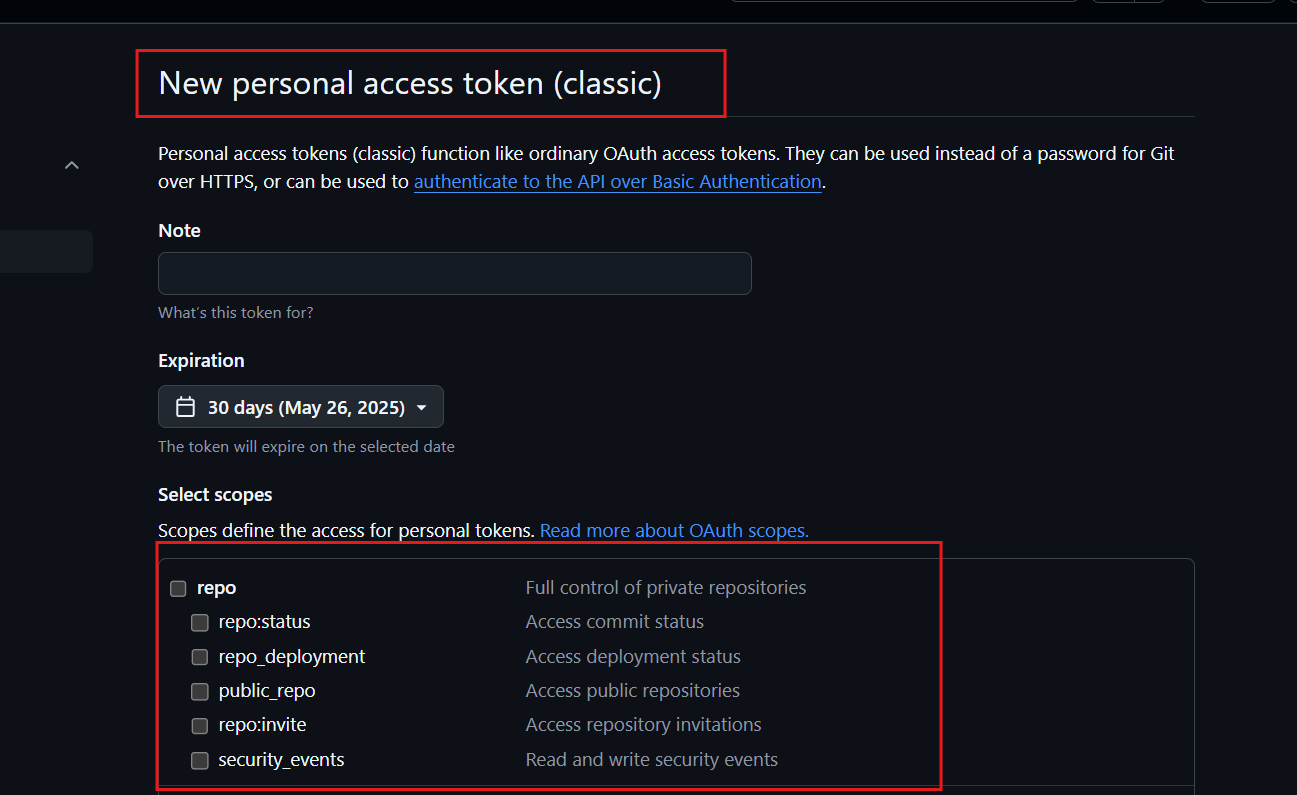
git push origin master



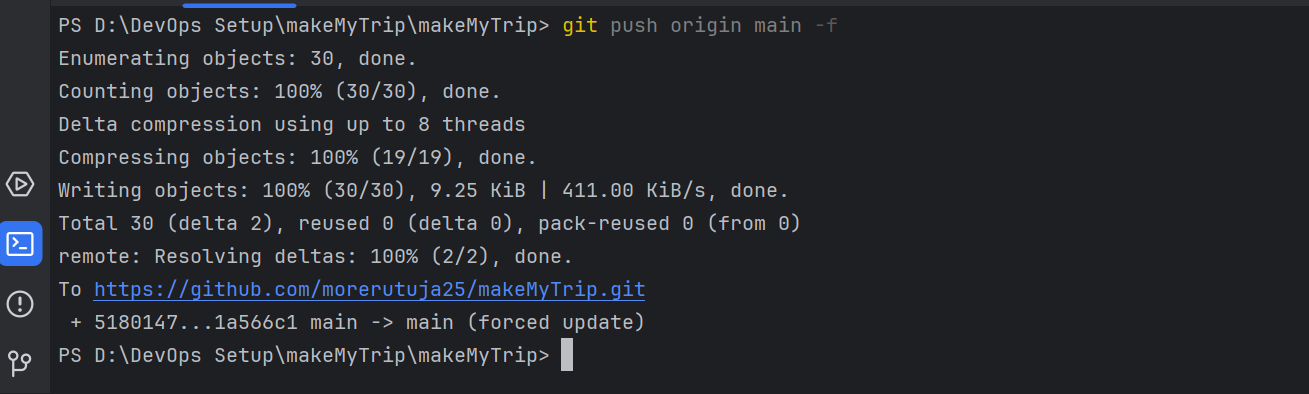
If github denies your your push request then it is because of your user doesn't have the necessary permissions to push to that repository.

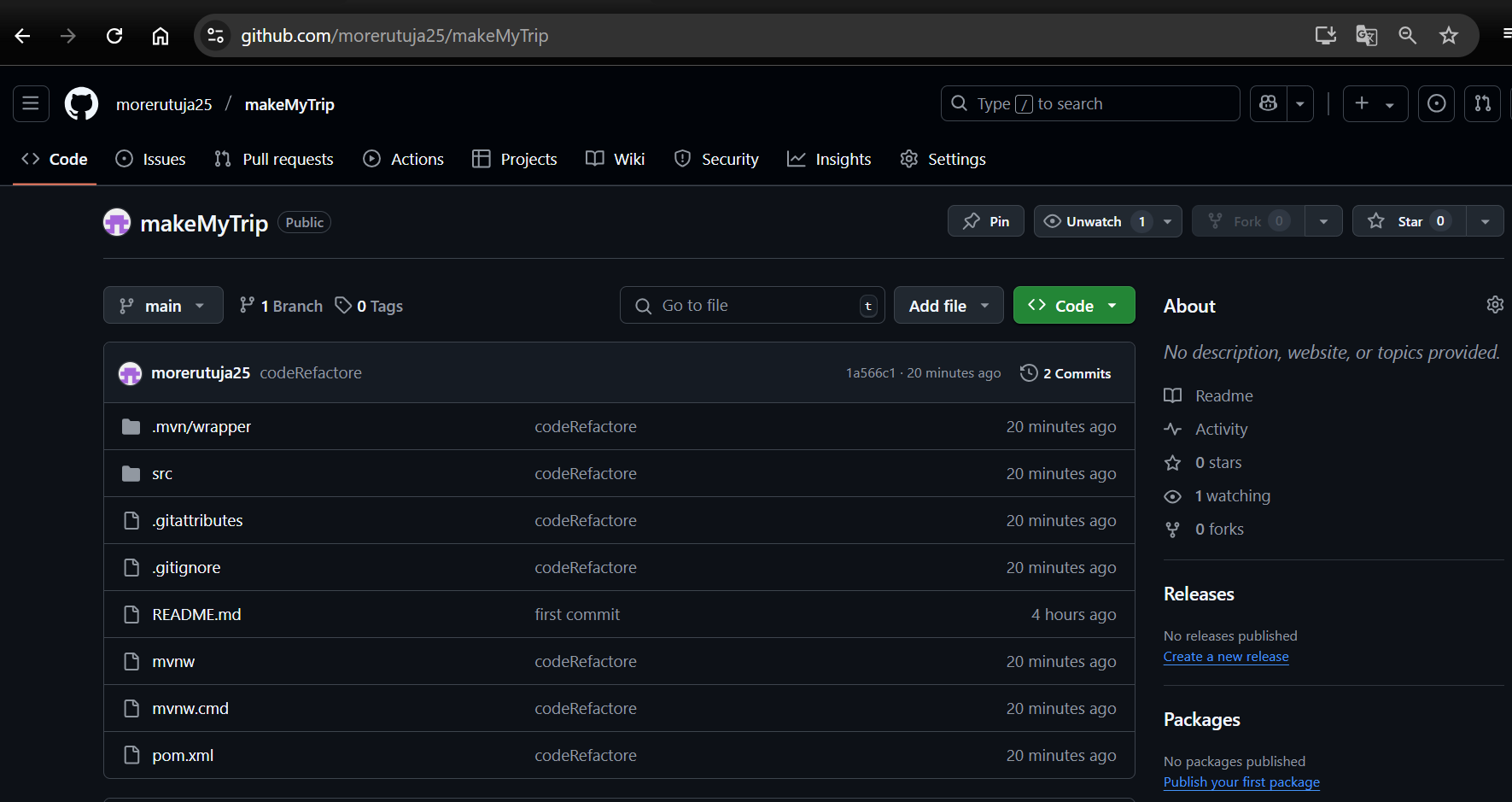
To fix this issue we need to follow below steps

1. Go to GitHub → **Settings** → **Developer settings** → **Personal access tokens** → **Generate new token**.
2. Select the **repo** scope for full control of private repositories.
3. Copy the token and use it when prompted for a password.



Now we can able to push our code





Add new class - bus.java

package com.example.makemytrip;

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

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

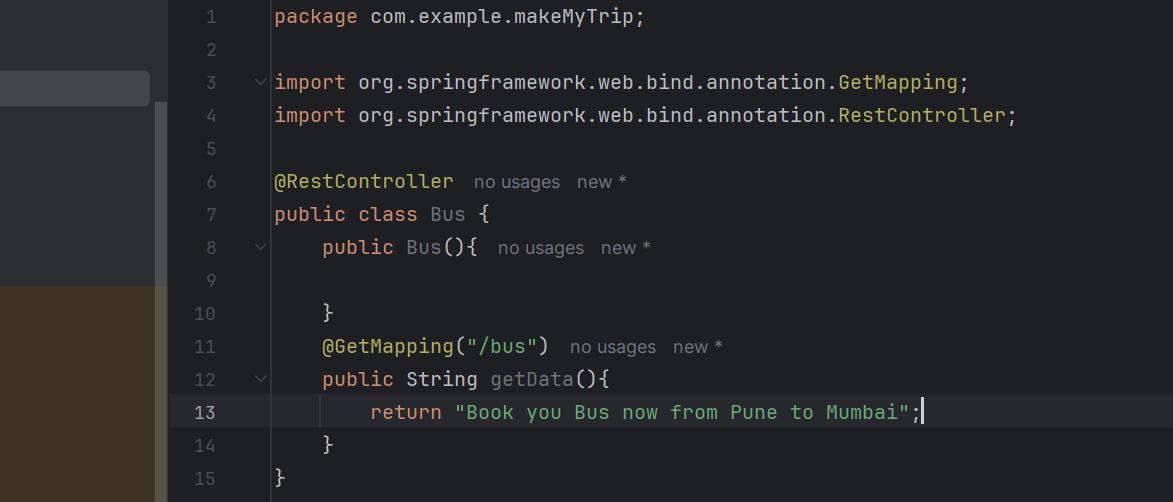
@RestController

public class bus {

@GetMapping("/bus")

public String getData() {return "Book your BUS from Kolkata to Pune at 10% OFF!";}

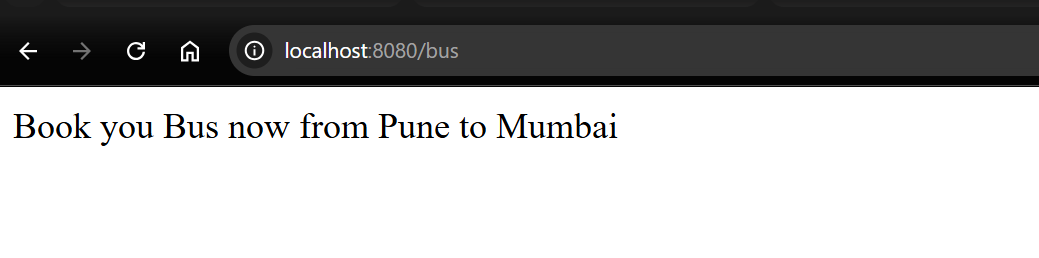
}



> mvn clean compile

> mvn clean package

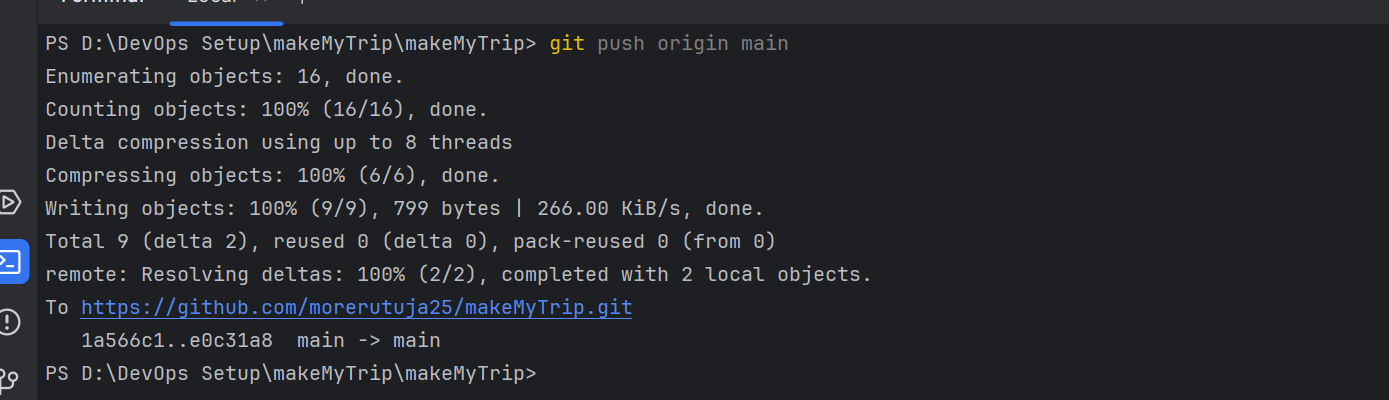
> mvn spring-boot:run



> git add --all

> git commit -m "Adding new java class"

> git push origin master



Code is now uploaded

