**Day 3 Creating our first microservices using Java Spring boot and tools used IntelliJ and GitHub to add source code**  
  
open the browser search for start spring io(Framework) or use below link

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

Refer the SS configuration to select and create:

Project: Maven

Language: Java,

Springboot version: 3.310

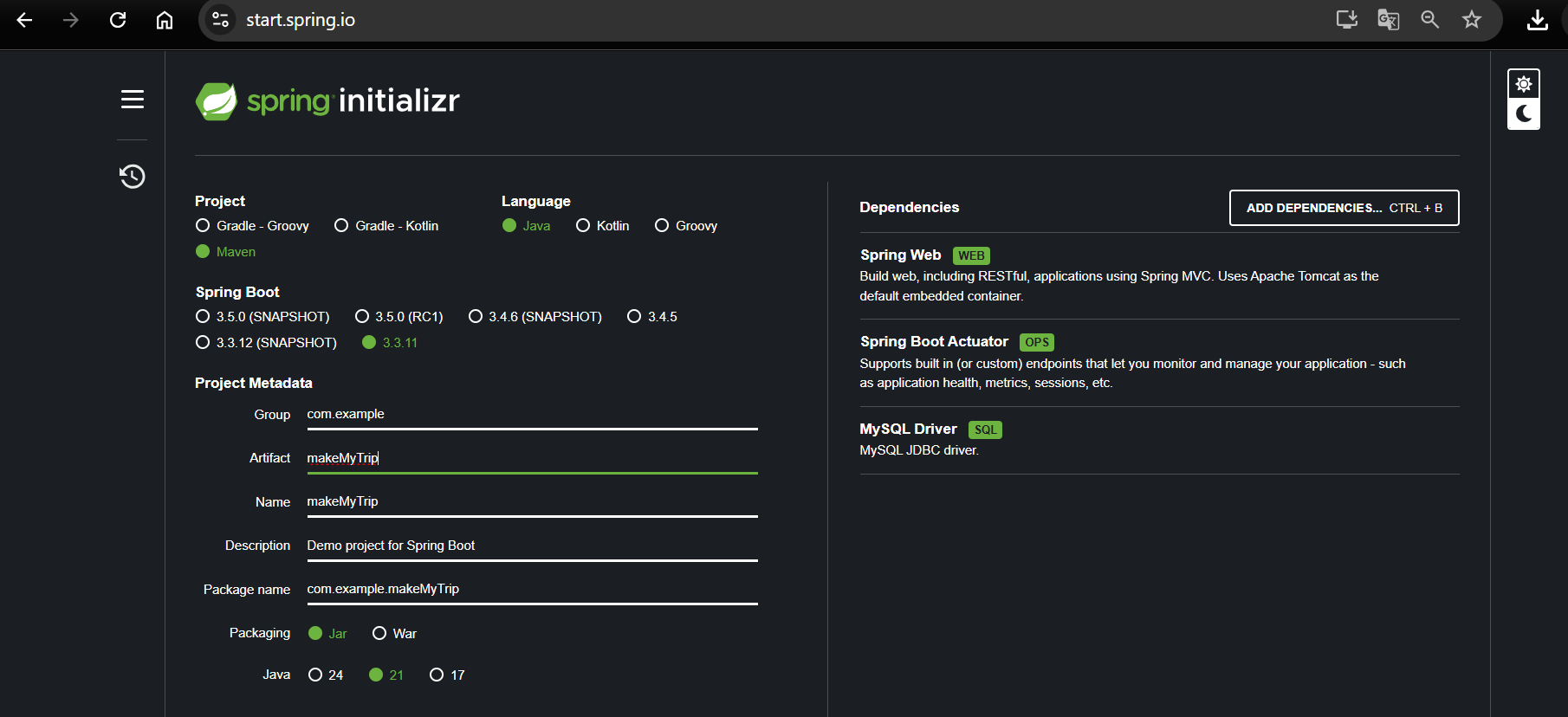
Artifact name: makemytrip

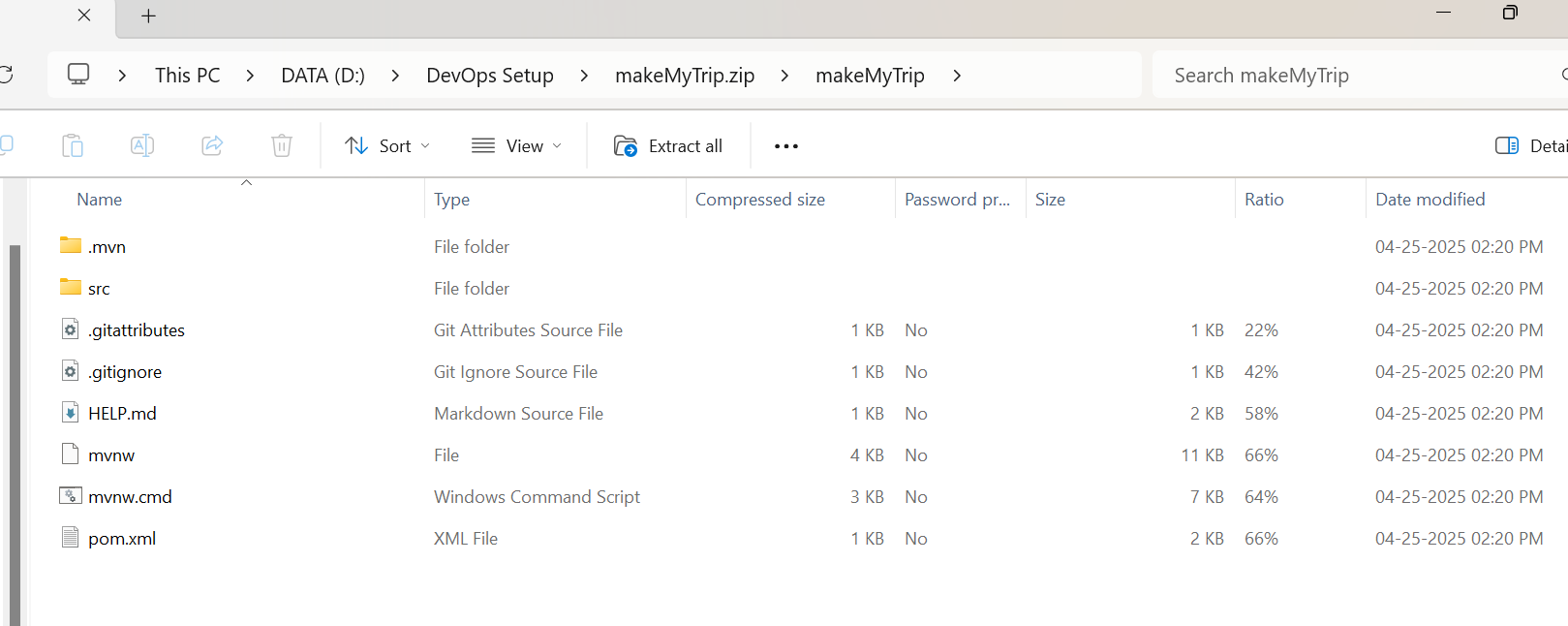
Packaging: Jar

Add the dependencies: Spring Web, String Boot Actuator(based on requirement we can add the new dependency)

Generate the zip file

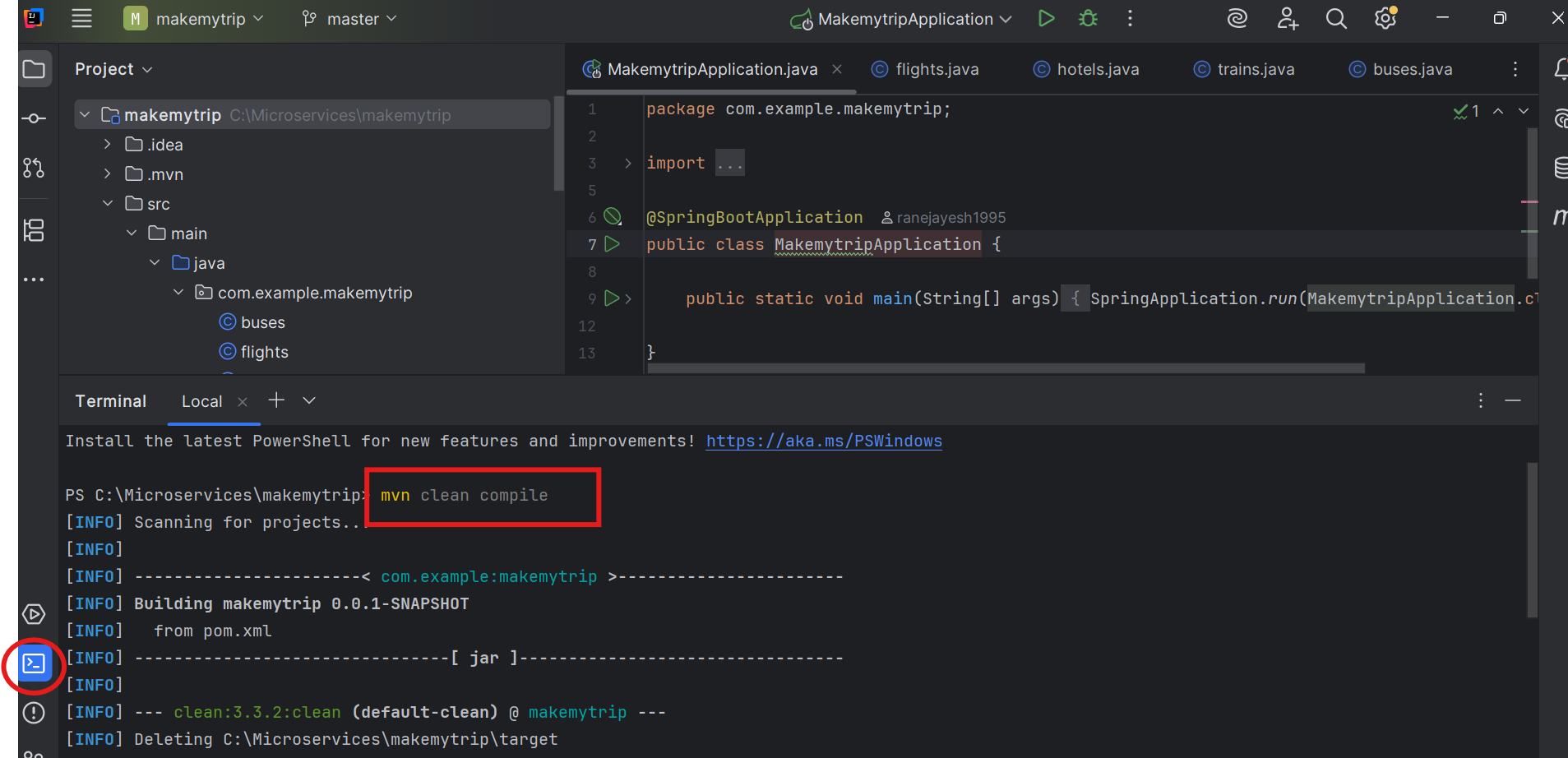
Extract the zip file





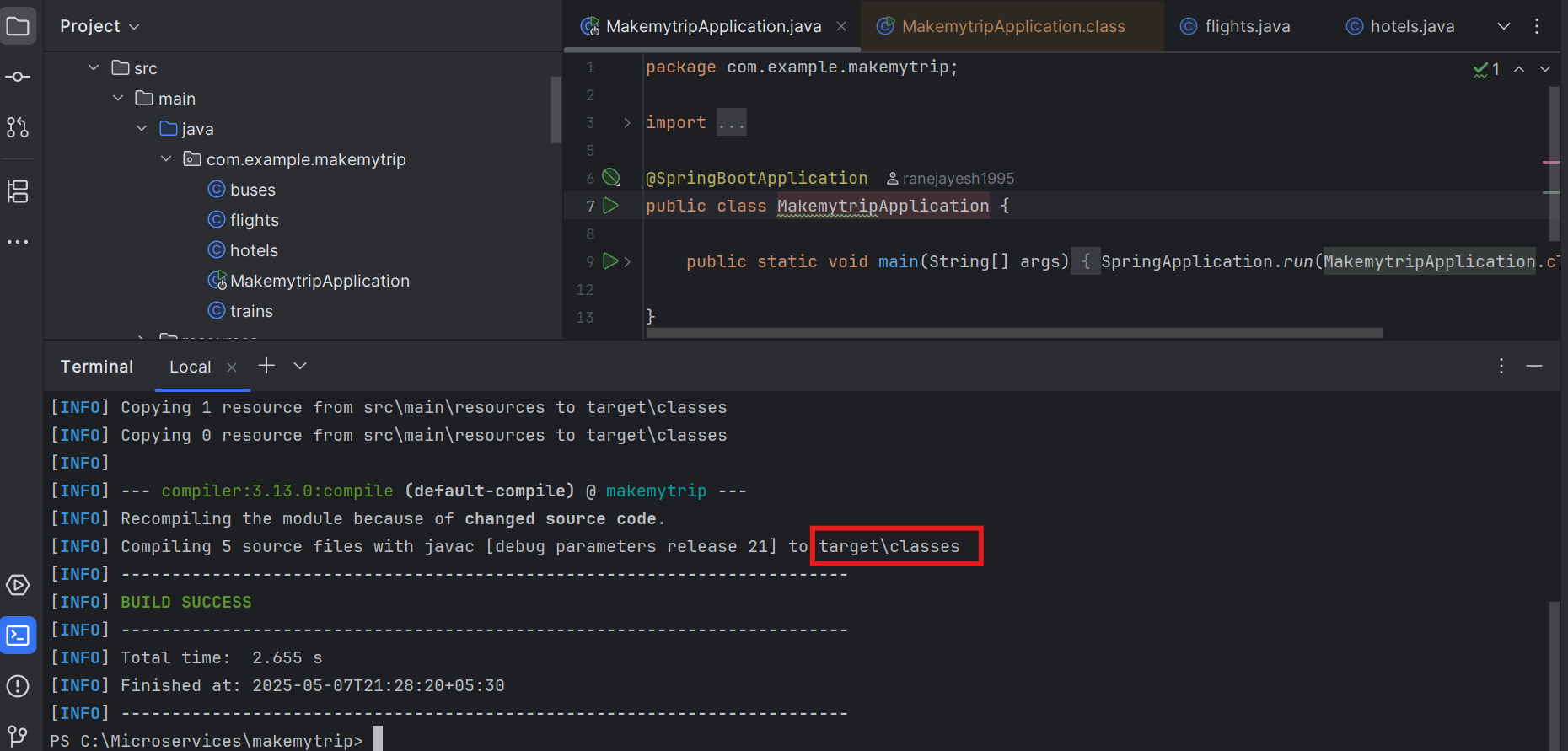
* Add this command under Terminal as below in below SS: - **mvn clean compile**

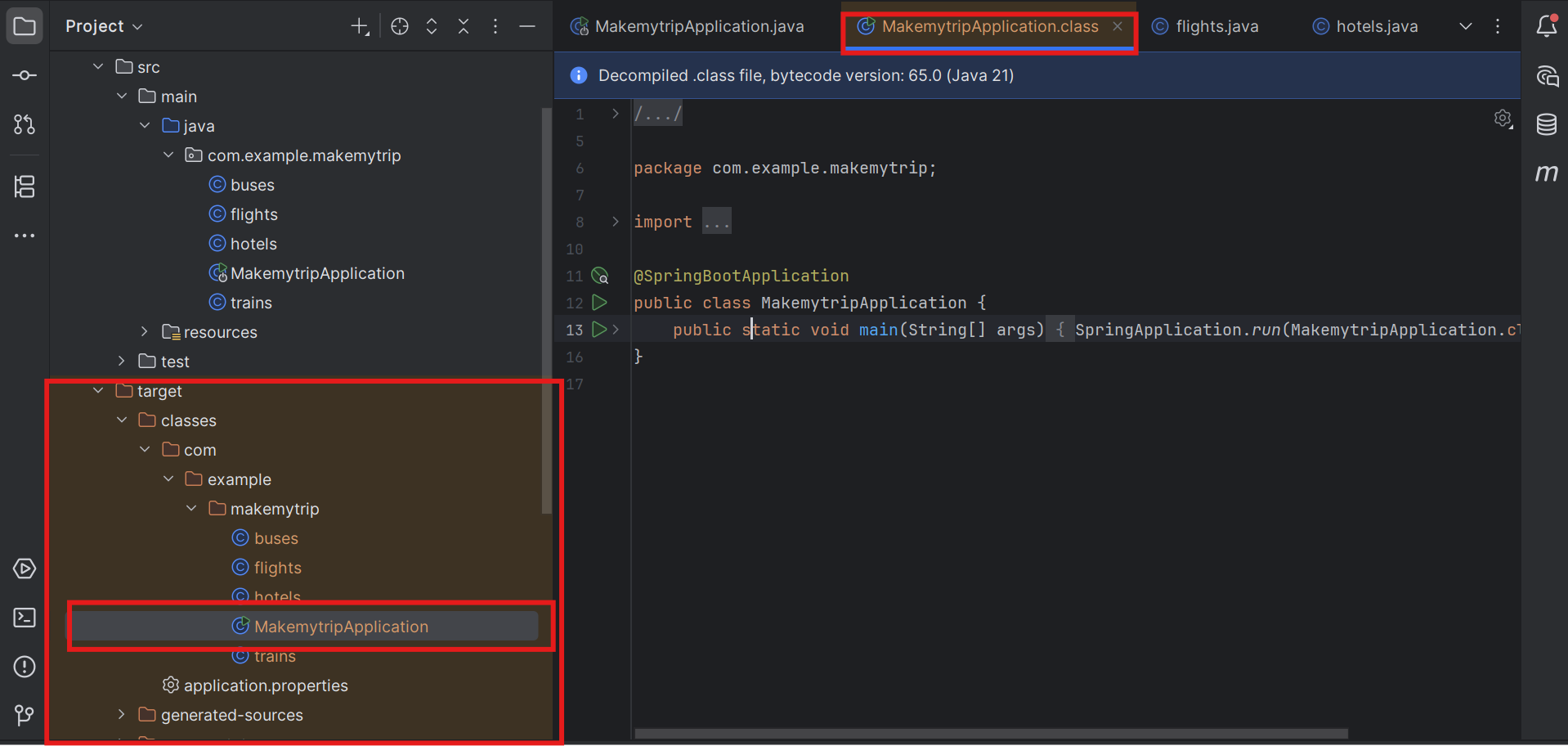
(to Creates .class file)



Above mvn clean compile should show build success like below ss

Location of .class file will be highlighted in below ss

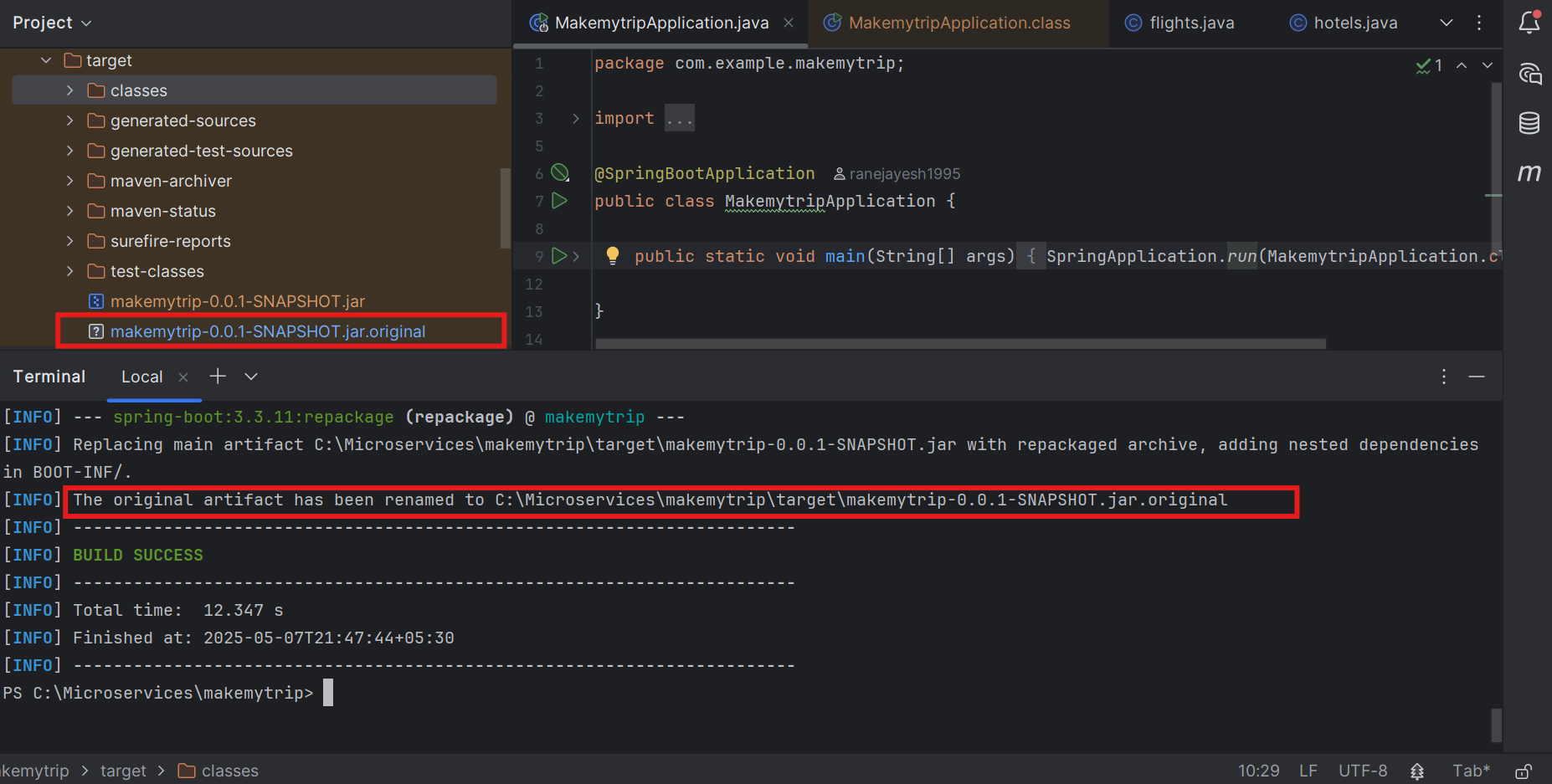


Now you can see below MakemytripApplication.class file is created when we run the **mvn clean compile**  


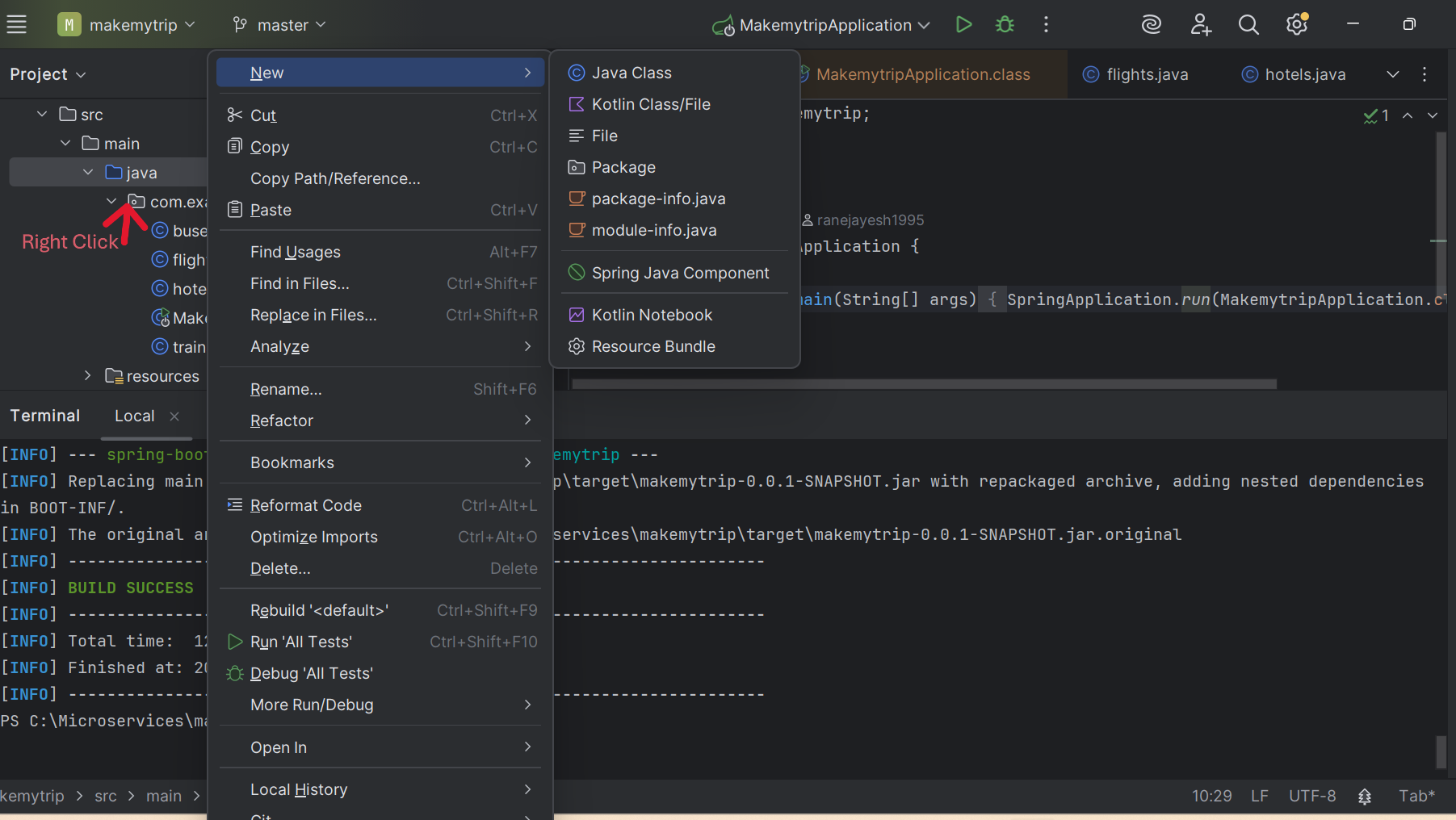
Now package:

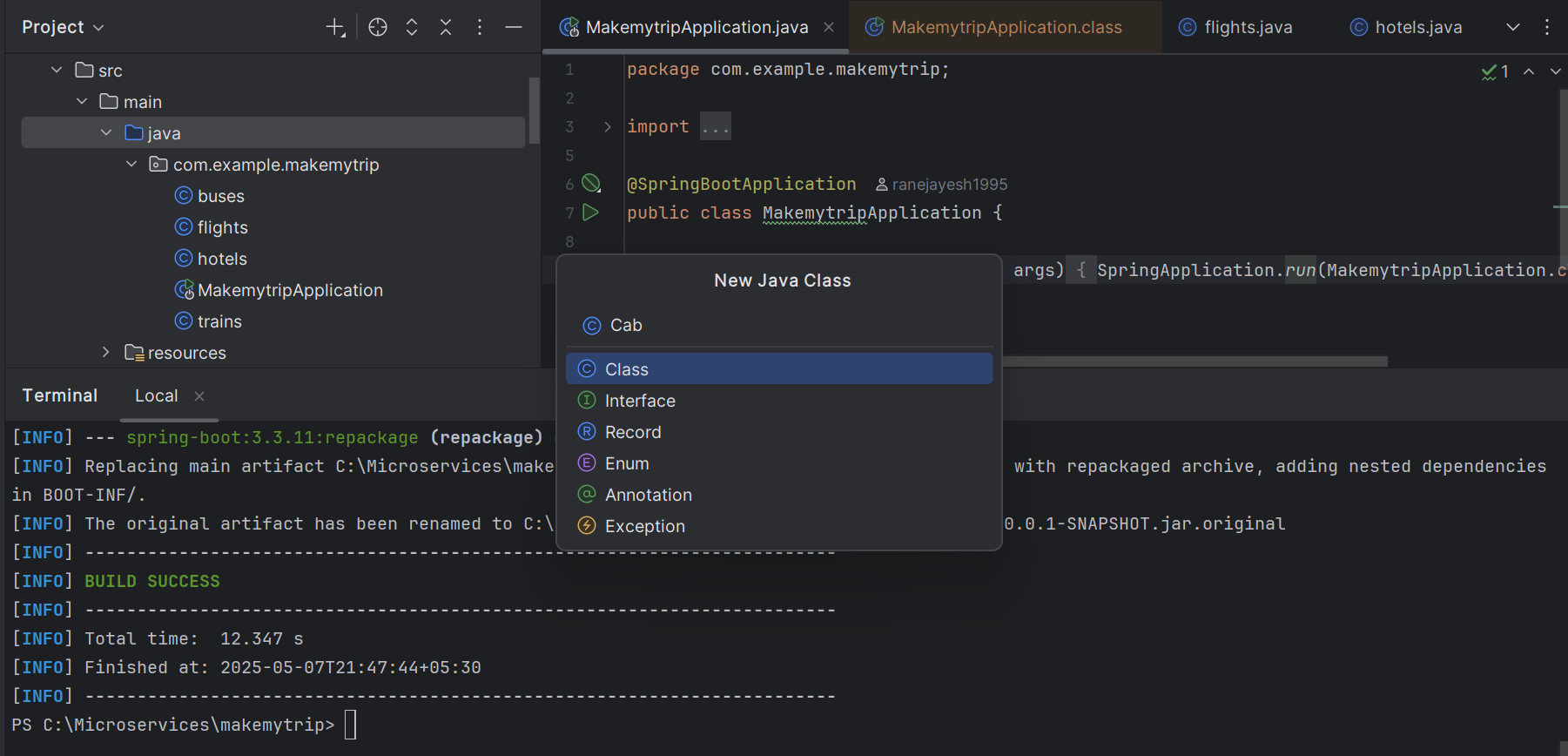
* Next Command: - **mvn clean package**: - Compiles your code, runs tests, and then packages the compiled code into a distributable format (like a JAR or WAR file), based on your project's configuration.

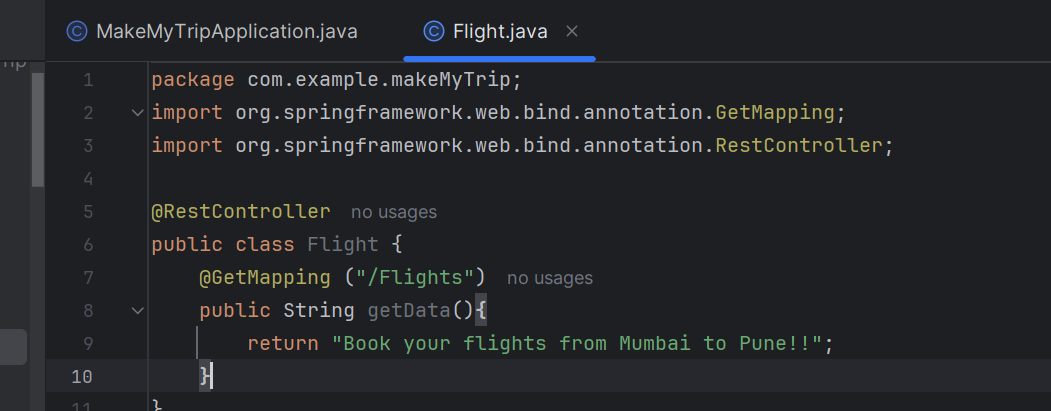
Default package is .jar



Now add a java file Right click on Below java .java File> new > java class > Cab



  
And add below code to this file flights.java:



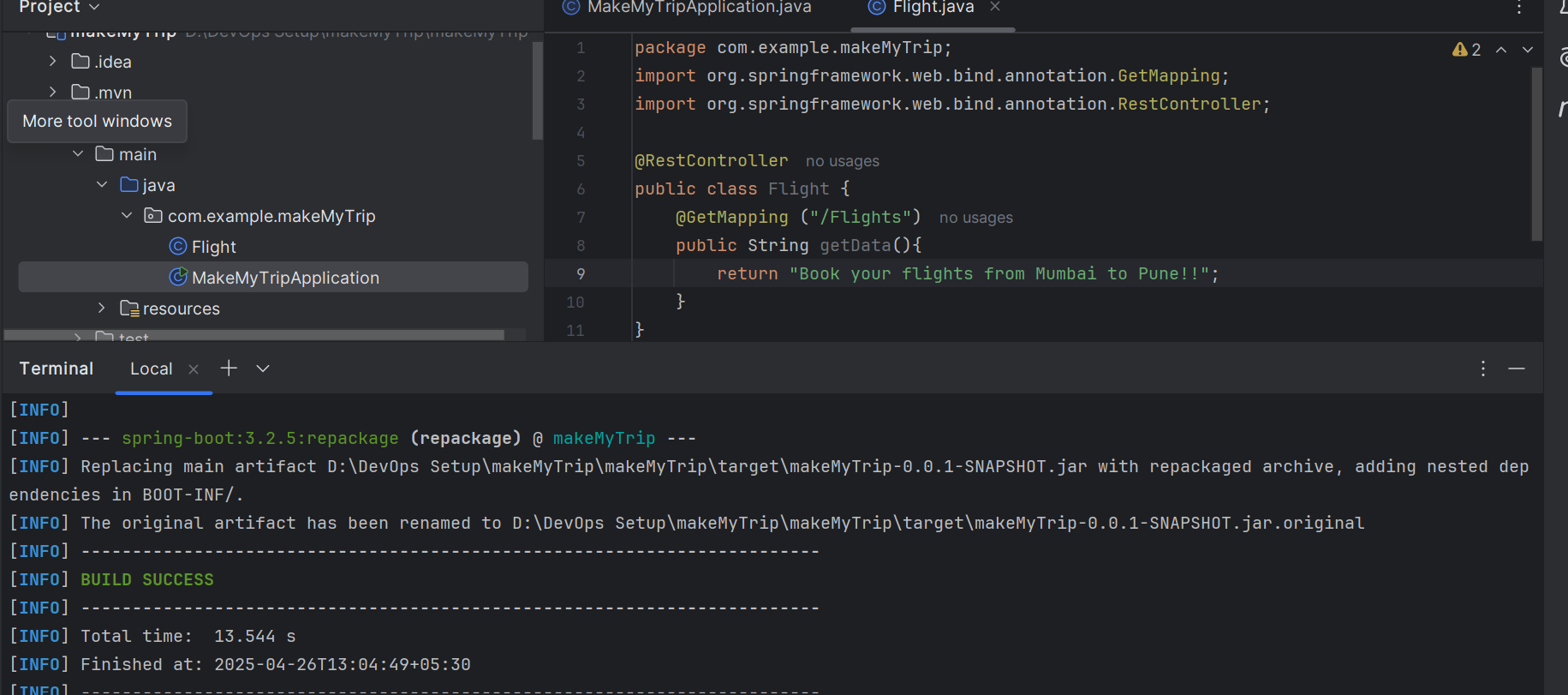
After ad ding new code - you need to compile and package it again.

> mvn clean compile

> mvn clean package

.class file should be generated for new flight.class file under target section

In case you want to create war file you have to make change on pom.xml file under <packaging> tag add war instead of jar

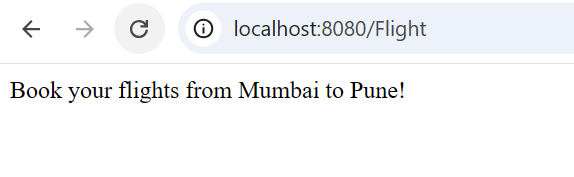


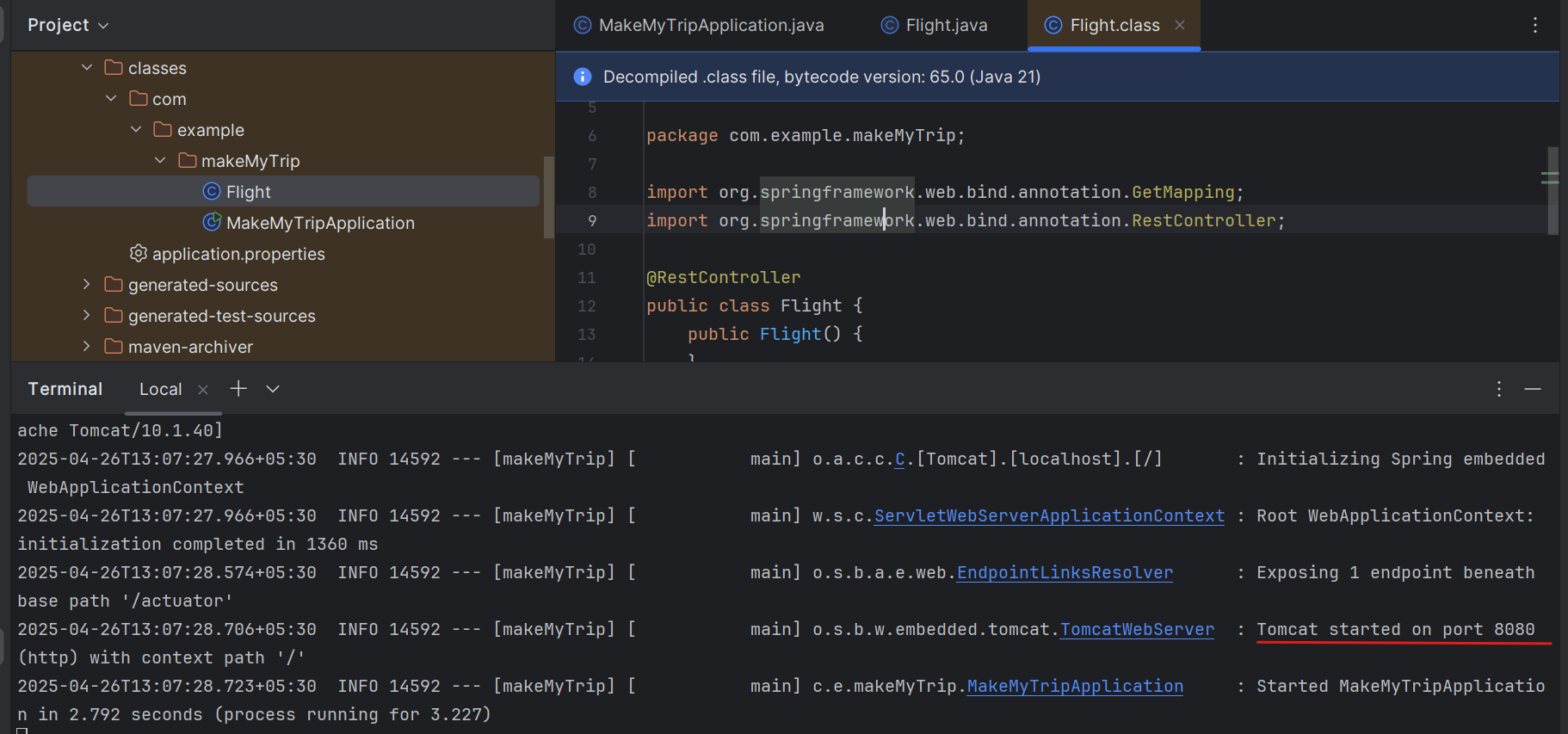
> mvn spring-boot:run

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

We can see the application is running on port 8080

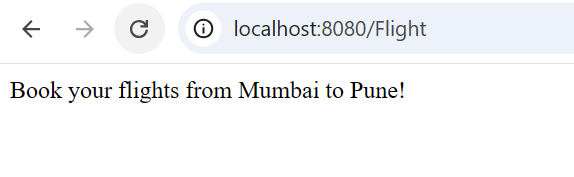
See the output response on [**http://localhost:8080/**flights](http://localhost:8080/flights) port number will be 8080



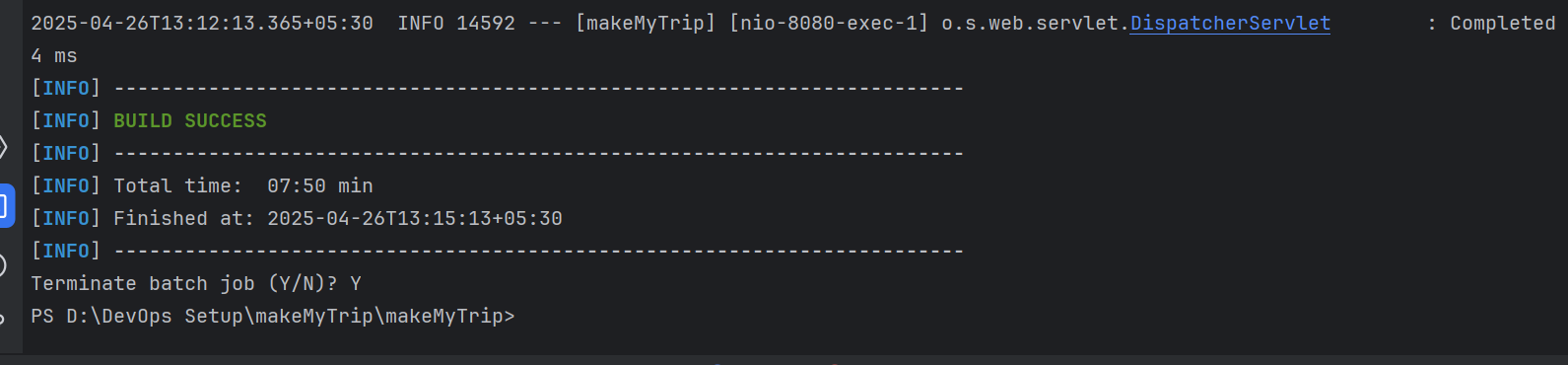


Now run the flights microservice on localhost

<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

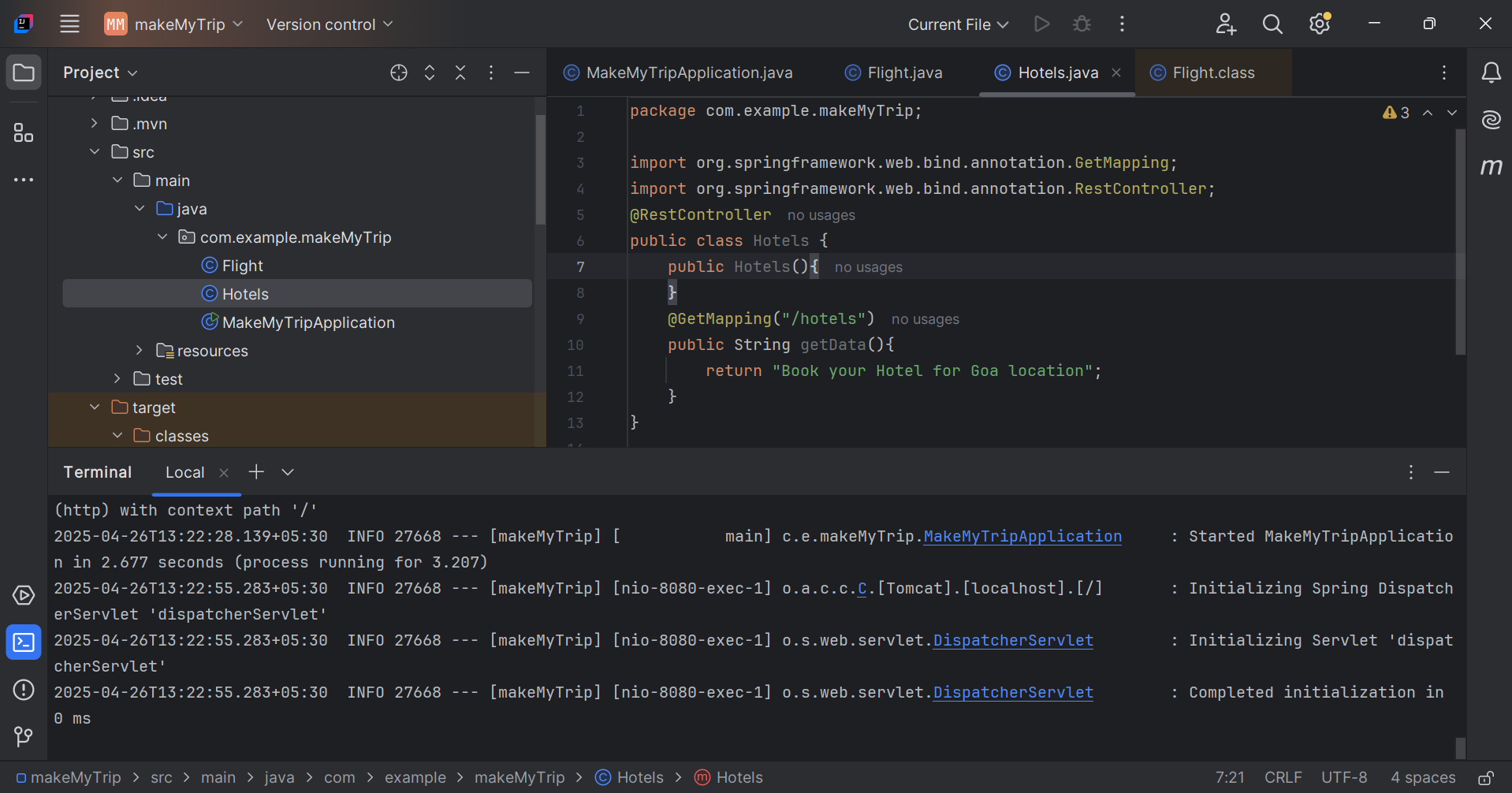
Now clean compile and package to make sure the new code is not spoiling the old working code.

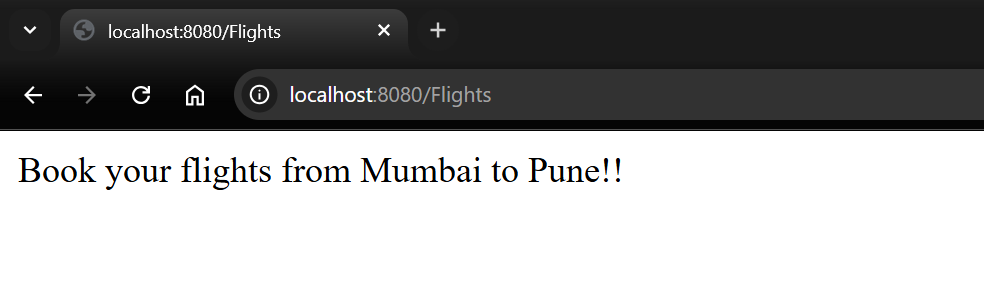
> mvn clean compile

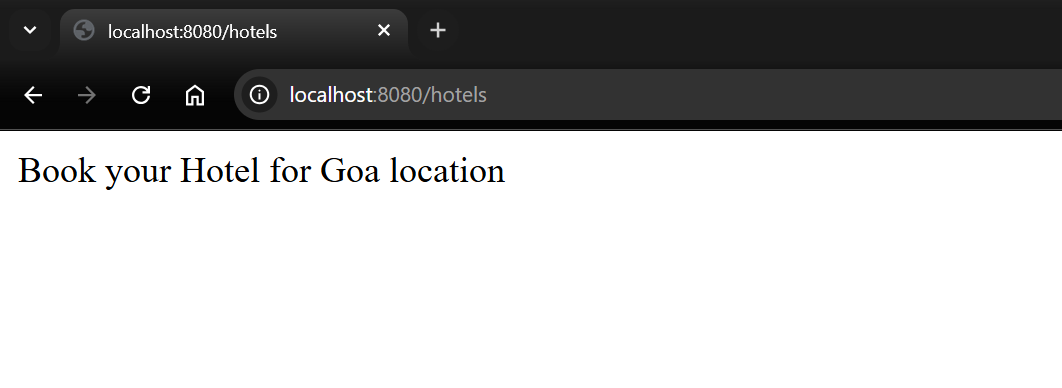
> mvn clean package

And then verify both the applications: hotels and flights

> mvn spring-boot:run





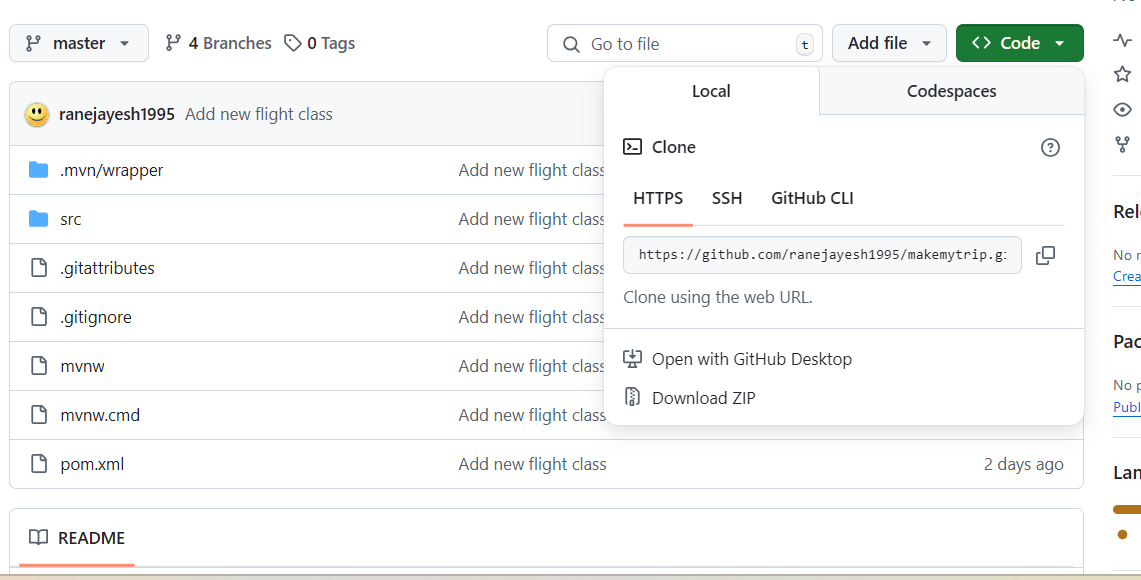


So we can see both features are working on local system.

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

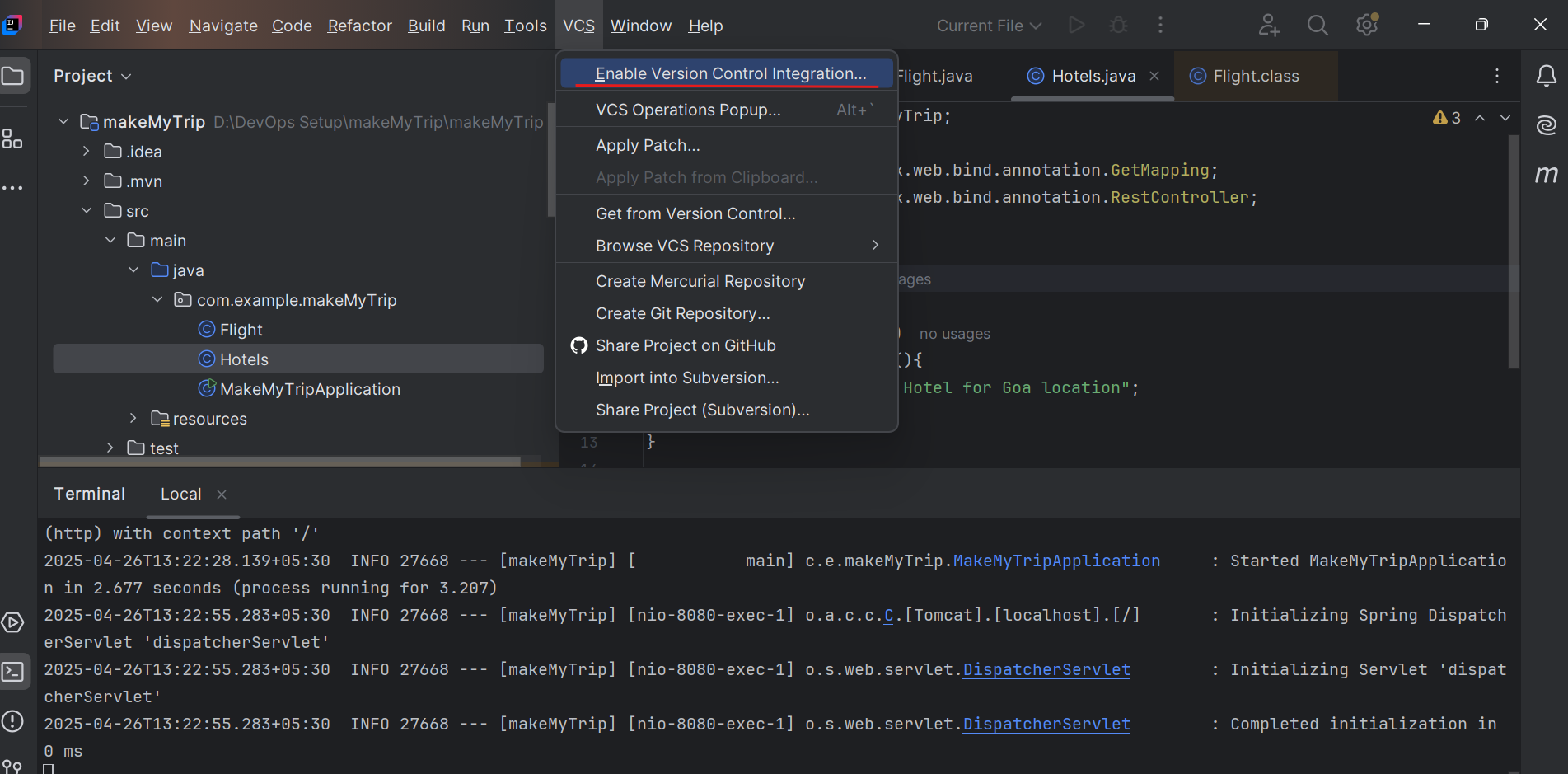




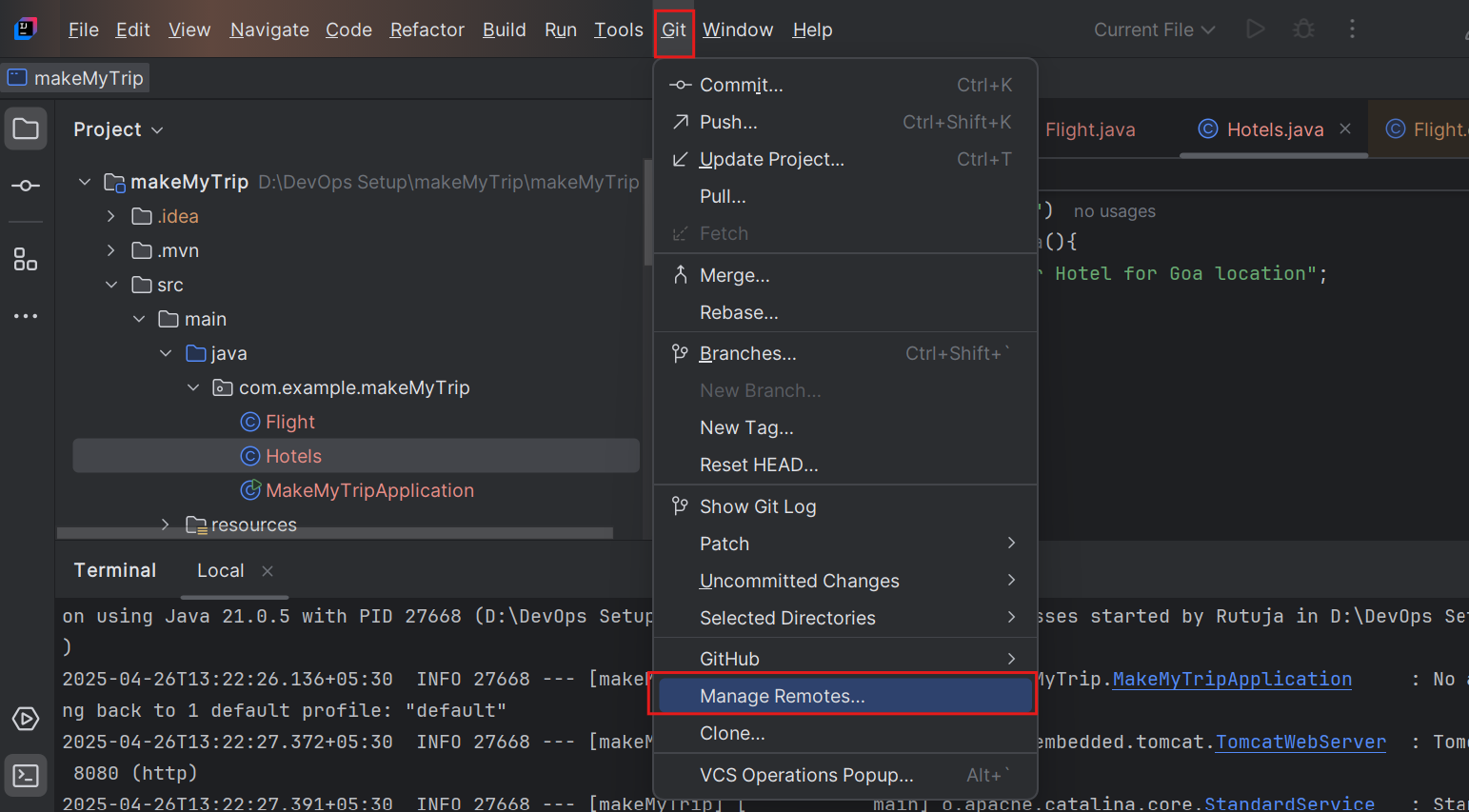
Push the code from local to remote repository

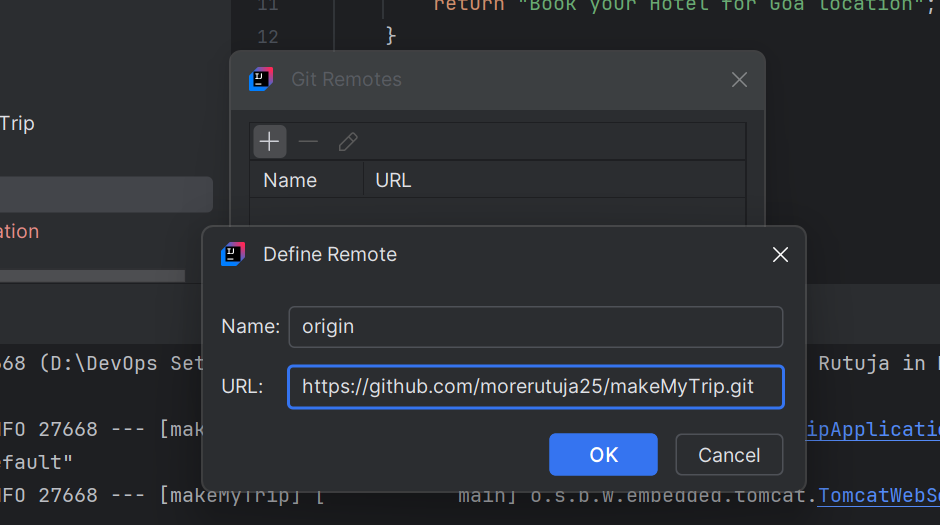
In IntelliJ IDEA enable version control integration:

VCS > Enable version control integration



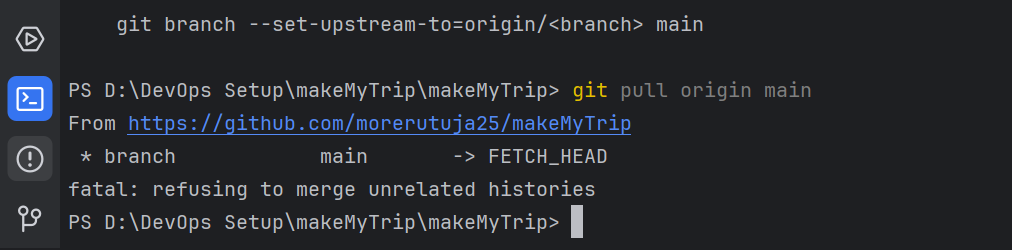
VCS > Manage remote > Enter URL of remote repository just copied





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

> git pull origin main



1st time you run this:

git pull

if after executing above and below command if still your repository link is not found as above ss while using **git pull orgin main** then might they will ask for

**git config -- global user.name "ranejayesh1995" (git username)**

**git config -- global user.email**  [**ranejayesh1995@gmail.com**](mailto:%20ranejayesh1995@gmail.com) **(git email ID)**

git pull origin master

git add --all

git commit -m "CodeRefactor"

git push origin master -f  
If above push command is also showing **erro:failed to push some refs to** something like this error then might you(user) have not permission to push so add the configuration from below **step D** **If denies to push**

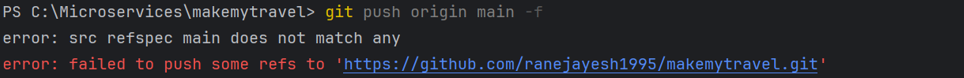
2nd and all times

git add --all

git commit -m "message"

git push origin master

**steps D**: when it denies to push because might you have not done the below setup please first setup the below following steps

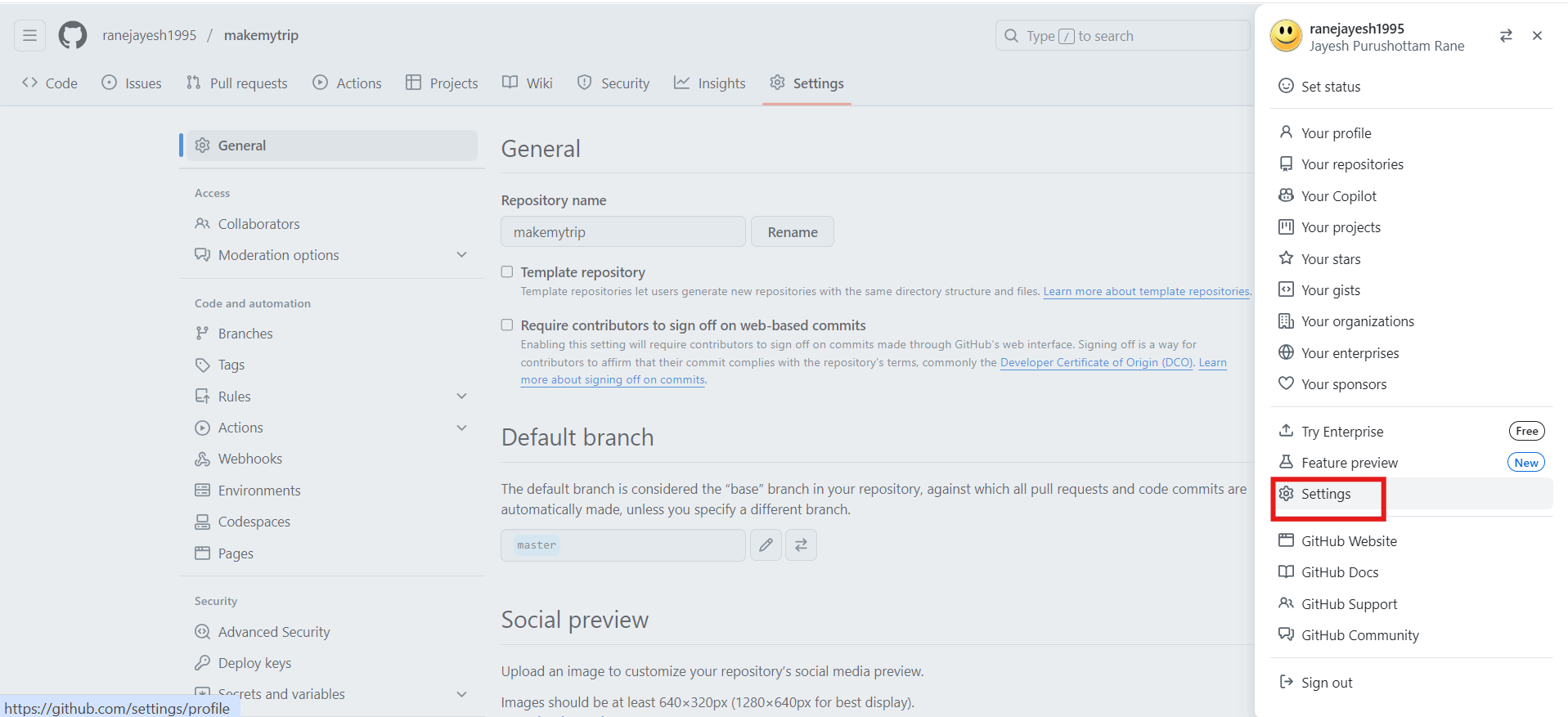


If github denies your your push request then it is because of your user doesn't have the necessary permissions to push to that repository.

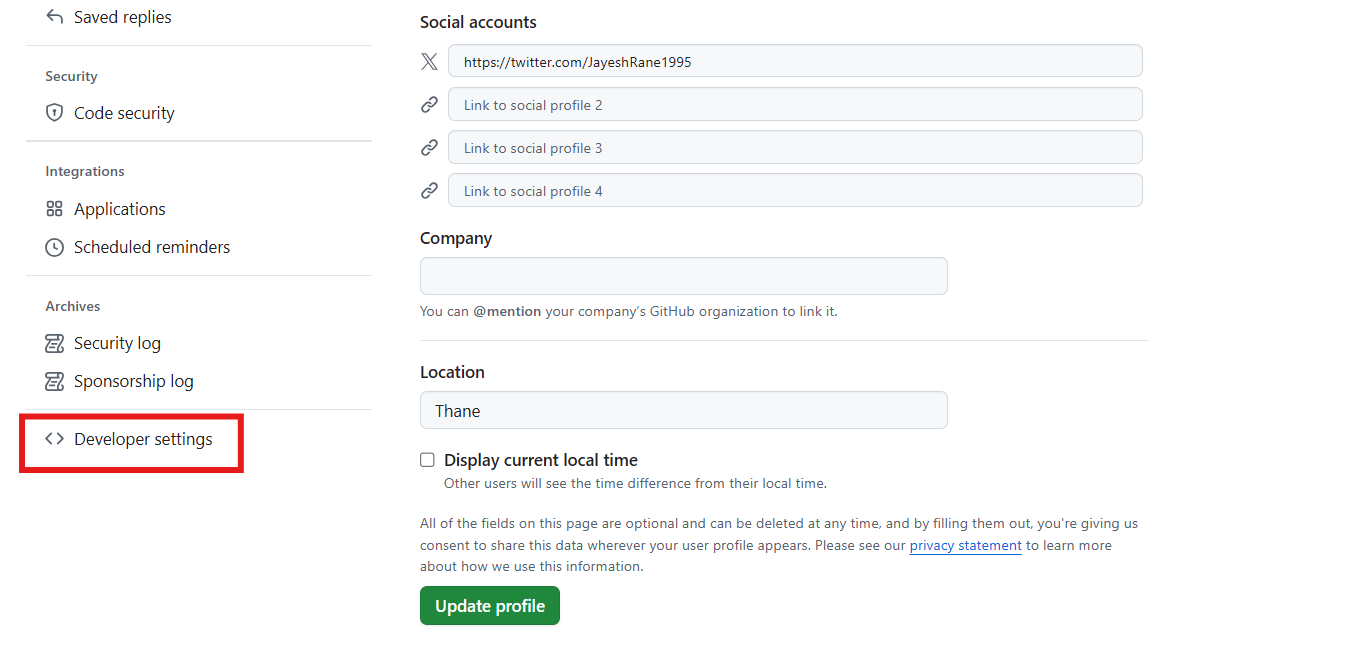
To fix this issue we need to follow below steps

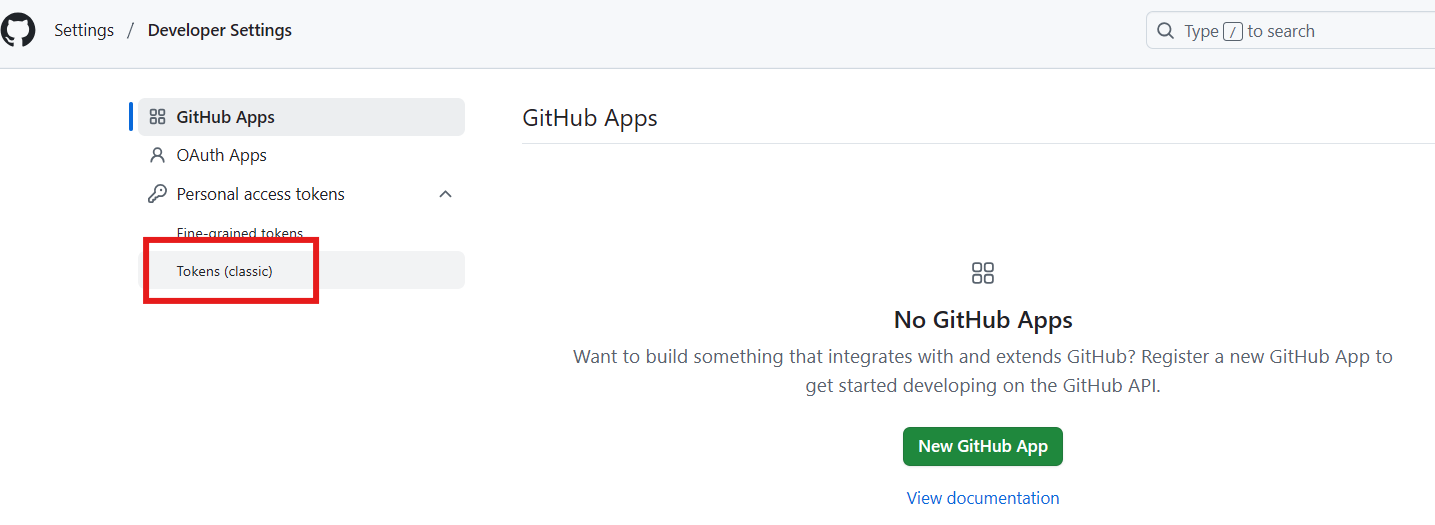
1. Go to GitHub → **Settings** → **Developer settings** → **Personal access tokens** → **Generate new token**.
2. Select the **repo** scope for full control of private repositories.
3. Copy the token and use it when prompted for a password.

1 .1

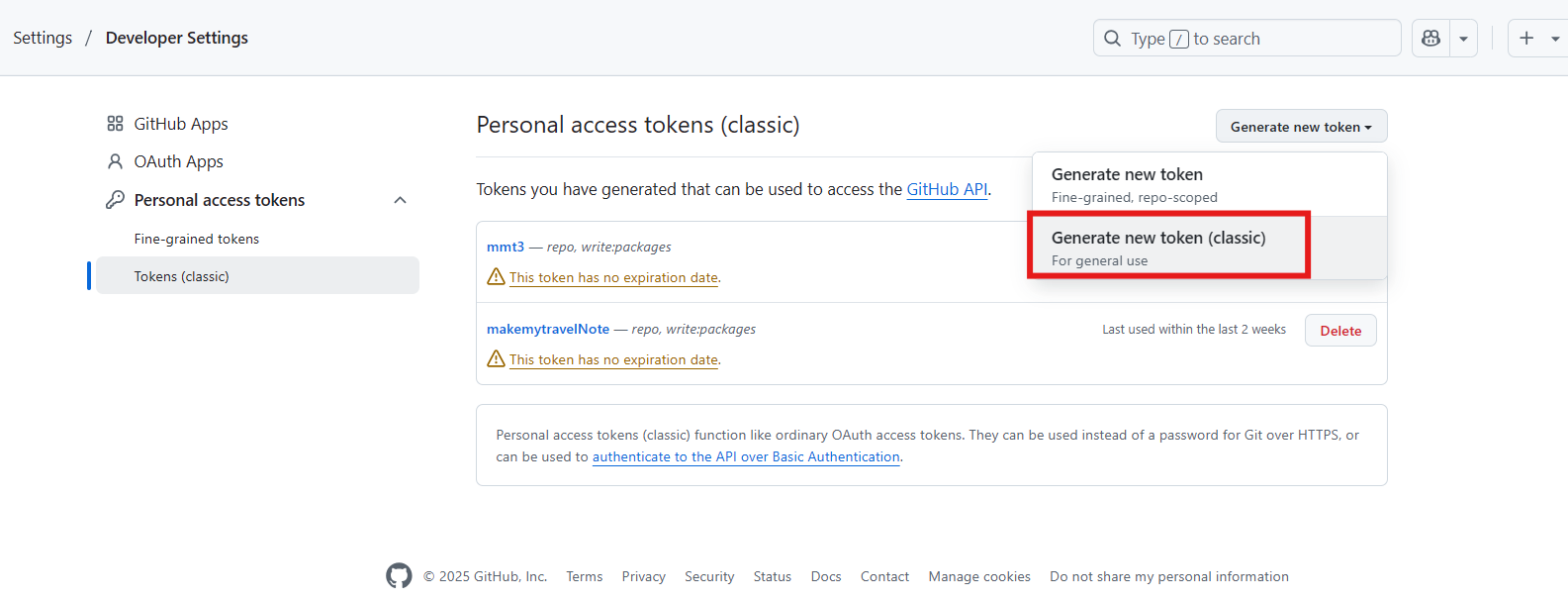


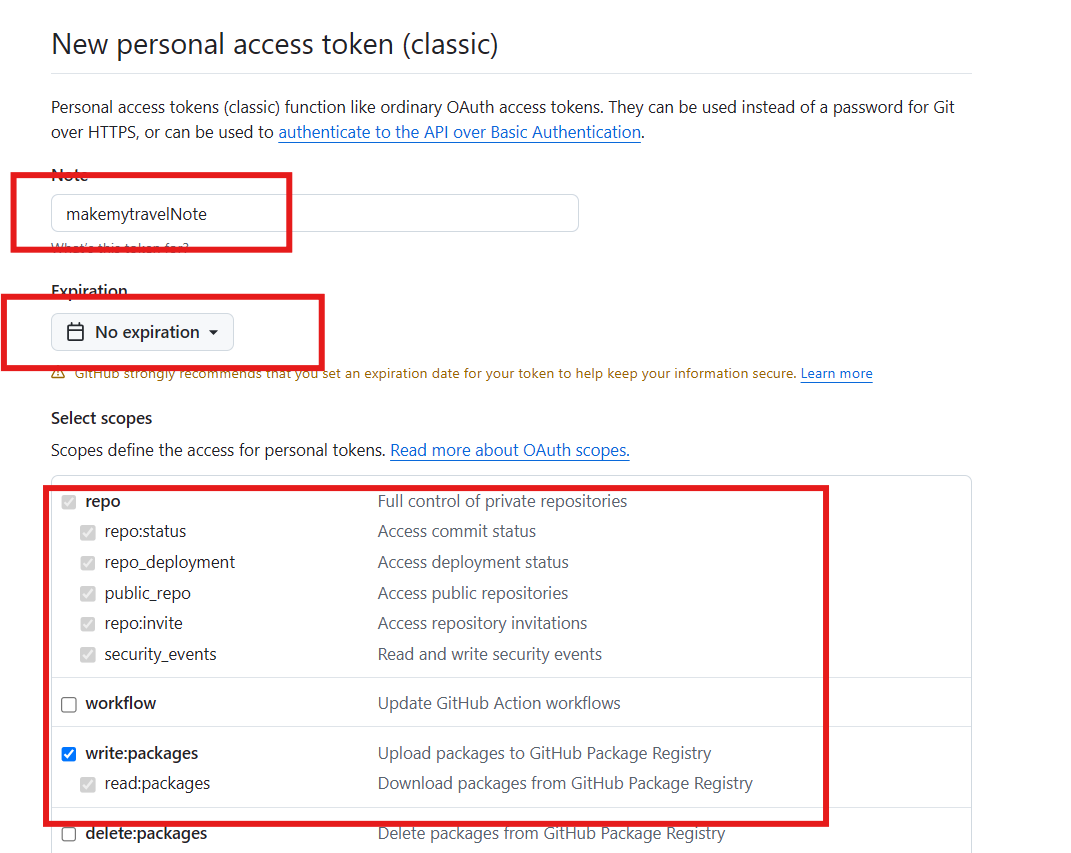
1.2

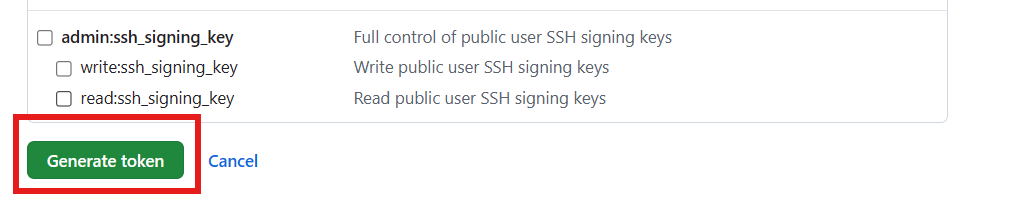


1.3

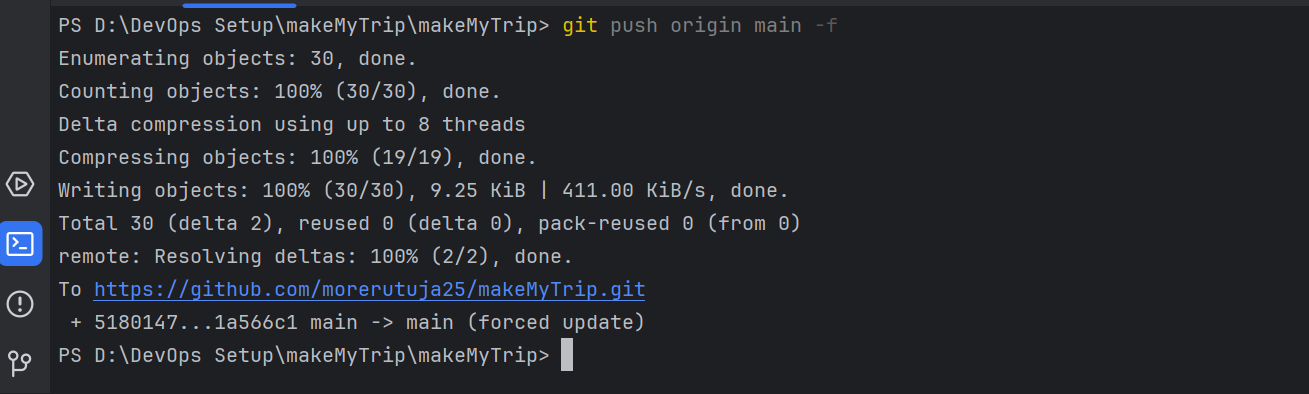
1.4

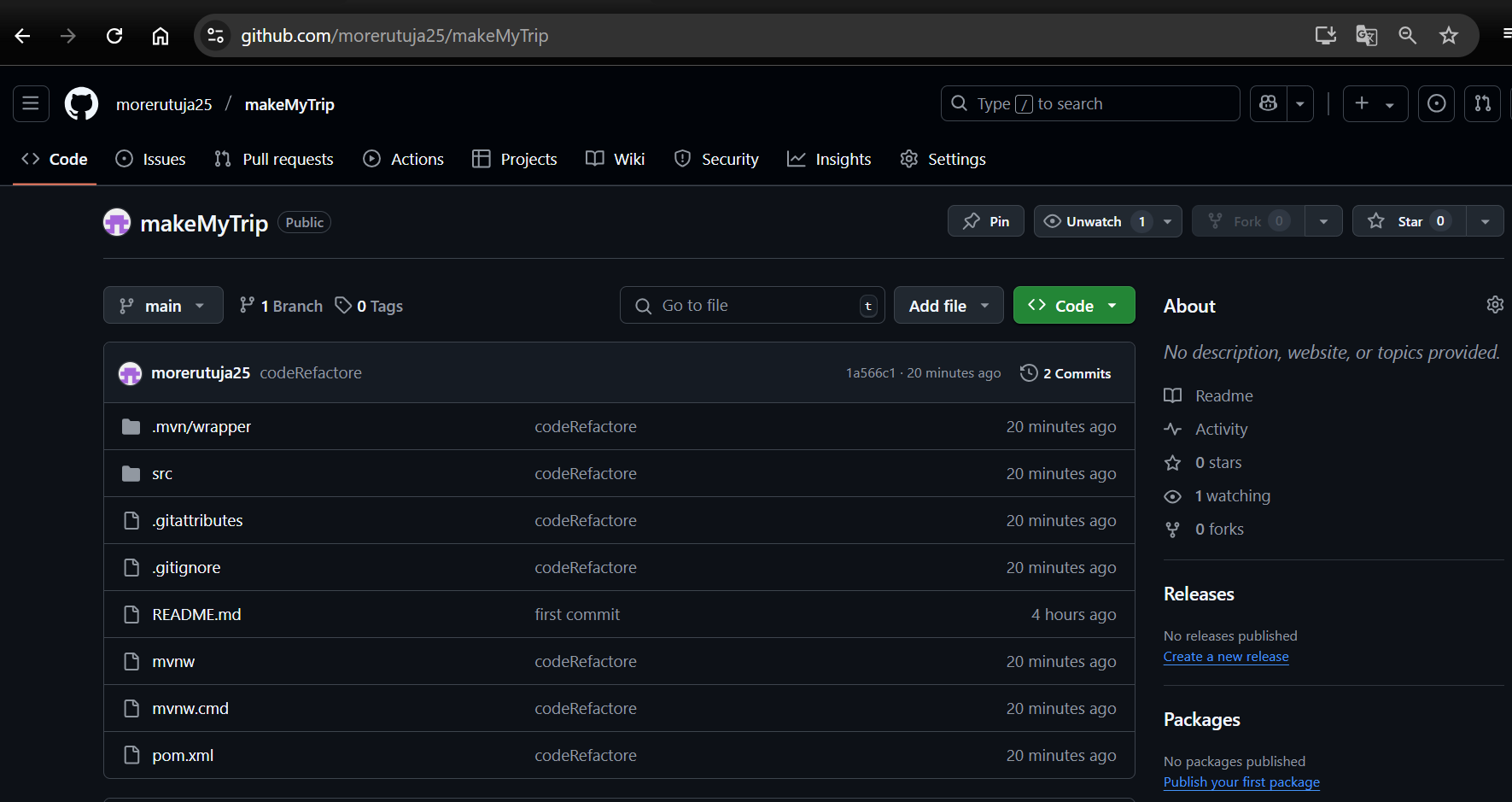


1.5

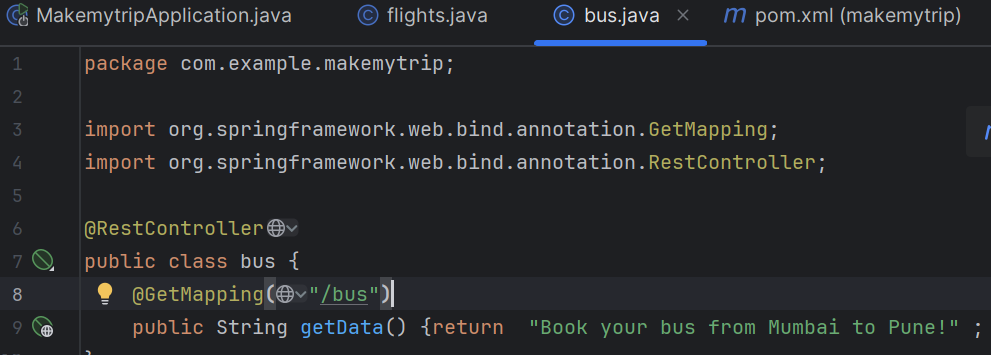


Now we can able to push our code on below ss you can see because read and write access we aren’t able to push





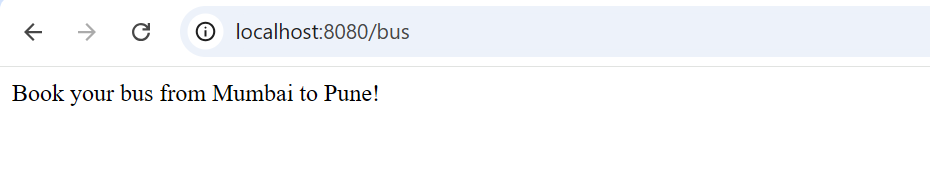
Add new class - bus.java



> mvn clean compile

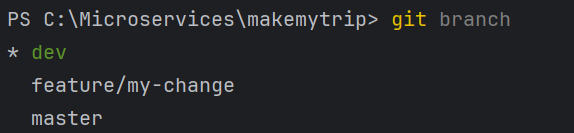
> mvn clean package

> mvn spring-boot:run

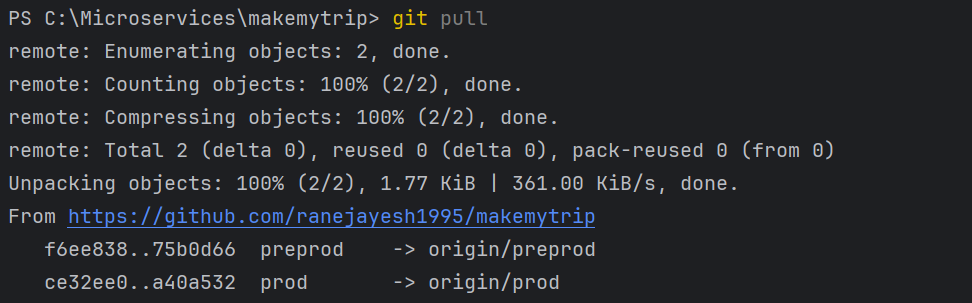


First verify which branch you are then

> git branch



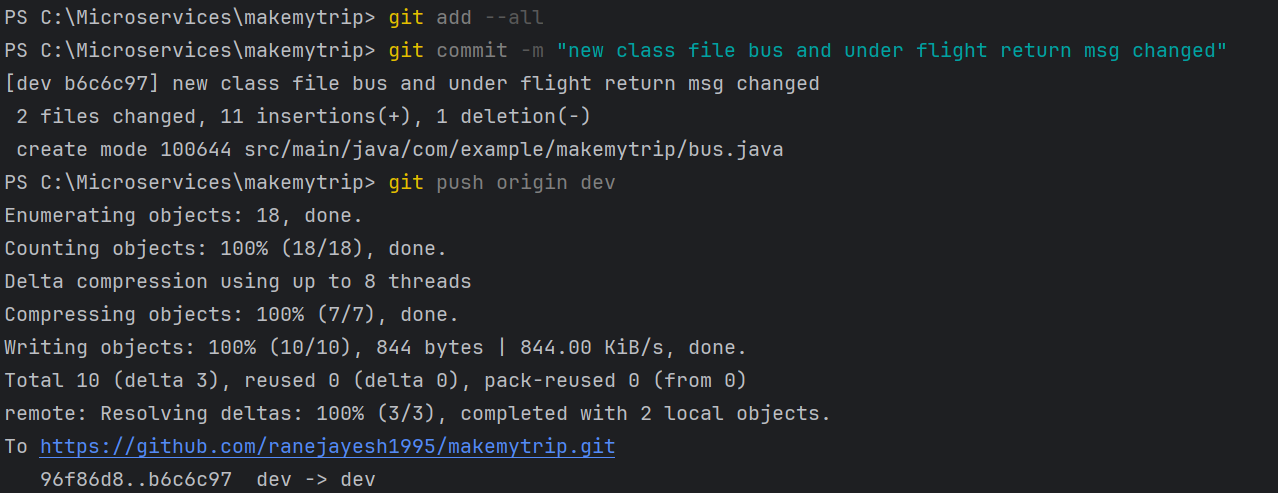
> git pull



> git add --all

> git commit -m "Adding new java class"

> git push origin master



File and Code is now uploaded

