Spring Boot – It’s build on top of popular Spring Framework.

Streams & Predicate.

Stream – parallel operation (functional programming)

1. Getting input from user using Scanner (wait for long time for user input) – blocking operation (20 ms)
2. Interact with the Database (600ms)
3. Print some data in the console (2ms)
4. Write the content to Log file (5 ms)

Sync – way & Async-way (parallel)

Live Video Streaming - Getting the data continuously (time dependent)

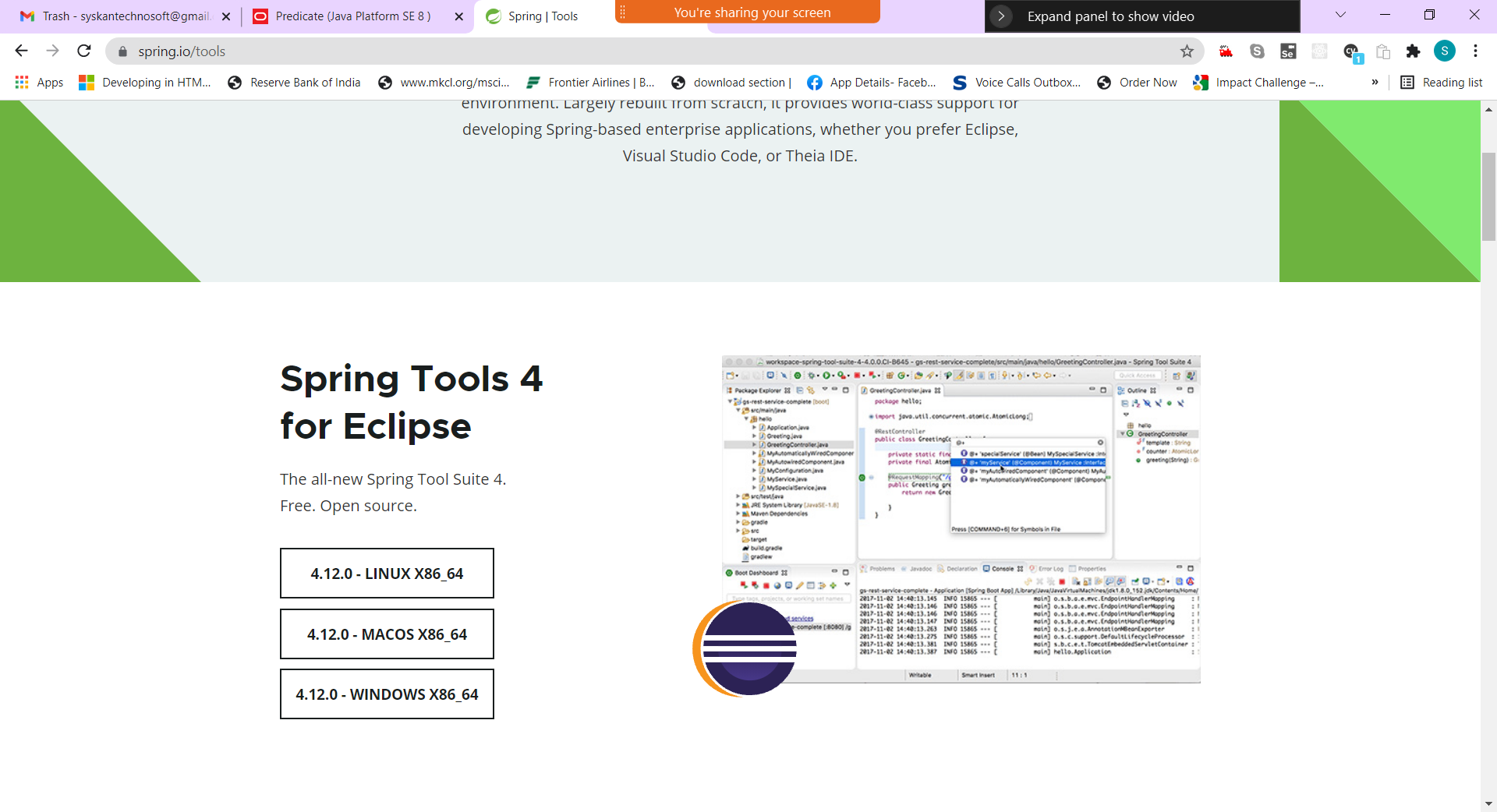
Streams -- filter, map, (Group of data- Collection API)

Lambda – anonymous function/nameless ()->statement;

Functional interface --- Lambda is used to give implementation to the abstract method of functional interface.

<https://www.geeksforgeeks.org/java-8-predicate-with-examples/>

STS – Spring Tools Suite (spring.io/tools)



WebService –

The term **Web service** (**WS**) is either:

* a service offered by an electronic device to another electronic device, communicating with each other via the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), (internet) or
* a server running on a computer device, listening for requests at a particular port over a network, serving web documents (HTML, [JSON](https://en.wikipedia.org/wiki/JSON), [XML](https://en.wikipedia.org/wiki/XML), images).

In a Web service a Web technology such as [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) is used for transferring machine-readable file formats such as [XML](https://en.wikipedia.org/wiki/XMLHttpRequest) and [JSON](https://en.wikipedia.org/wiki/JSON).

With the help of a URI (Endpoints) accessing the method defined using any programming lang.

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

ATM – (ABC Bank Debit Card)

1. ABC ATM for Withdrawal (withdraw, deposit, pin change, address/mobile/email update, print 5 transaction, apply for loan)
2. XYZ ATM for Withdrawal (withdraw, deposit, last 5 transaction)

API (End point) (JSON/XML)

JSON – JavaScript Object Notation

XML – eXtensible Markup Lang (Tag based lang)

Parser – Tool for extracting the data

Parsing – It’s an operation to extract the data

employee

id name email

100 test [test@gmail.com](mailto:test@gmail.com)

101 abc [abc@gmail.com](mailto:abc@gmail.com)

In JAVA

Public class Employee {

Private int id;

Private String name;

Private String email;

}

JSON format

Employees = [ {“id”:100,”name”:”test”,”email”:”test@gmail.com”}, {“id”:101,”name”:”abc”,”email”:”abc@gmail.com”}];

XML Format

<employees>

<employee>

<id>100</id>

<name>test</name>

<email>test@gmail.com</email>

</employee>

<employee>

<id>101</id>

<name>abc</name>

<email>abc@gmail.com</email>

</employee>

</employees>

Types of Web Service

1. SOAP Based (Simple Object Access Protocol) – WSDL (WebService Definition/Description Lang)
2. REST Based (Representational State Transfer) – It uses http methods

JAVA + Spring Boot

Javalin

Tools Needed

STS – <https://spring.io/tools>

JDK 1.8 SE – <https://www.oracle.com/java/technologies/downloads/#java8-windows>

Postman – API Testing Tool - <https://www.postman.com/downloads/>

Folder Structure of a Spring Boot Project (Maven Project)

Src 1) Main 2) Test

Target (output folder) jars/ classes

Main 1) java 2) Resources

Java – All Java Source Code

Resources – Static/templates & application.properties (Text file)

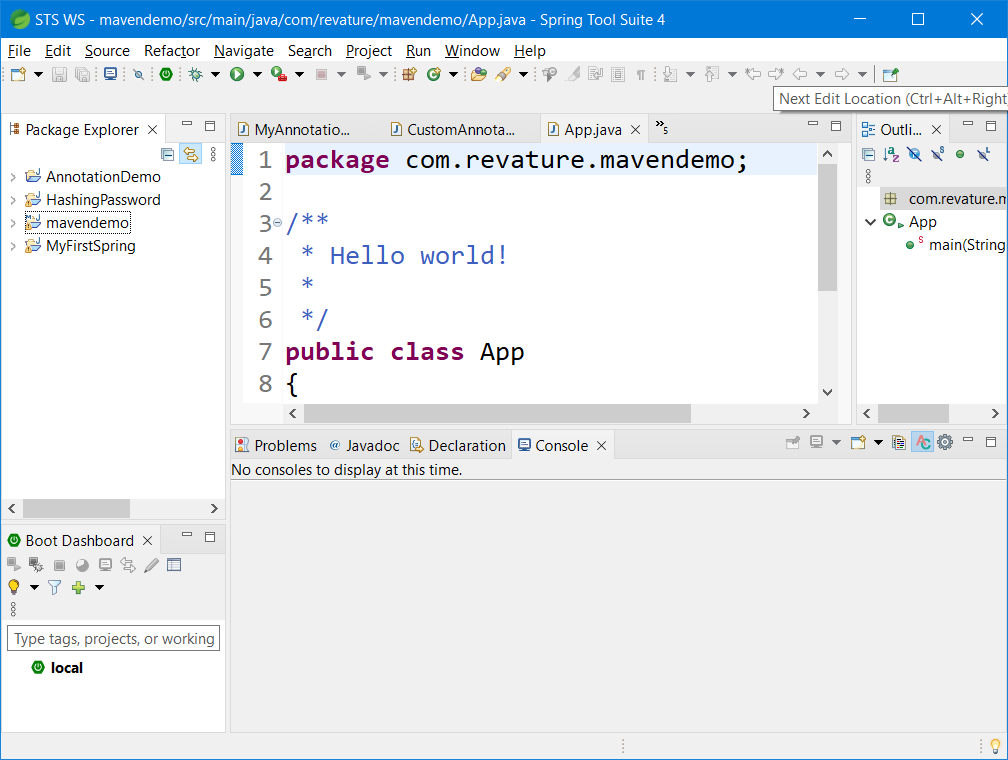
Test 1) java (All Unit Testing Files)

Pom.xml - Backbone of the Spring Boot Project

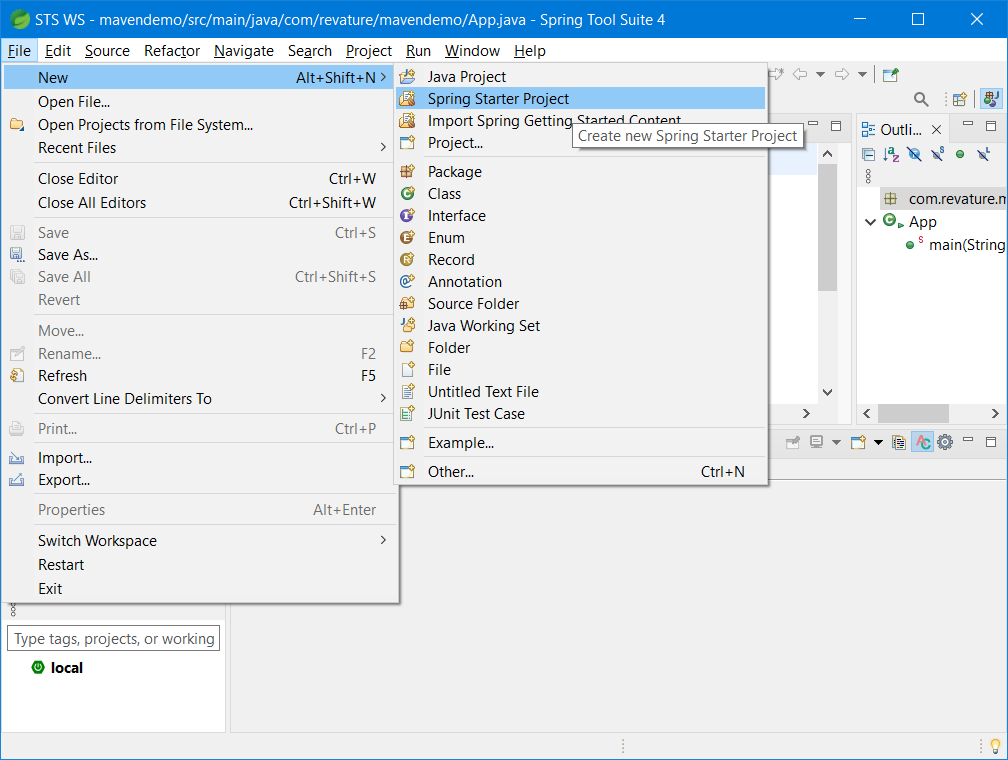
POM – Project Object Model

Creating a REST service using Spring Boot in STS

Step 1: Open STS

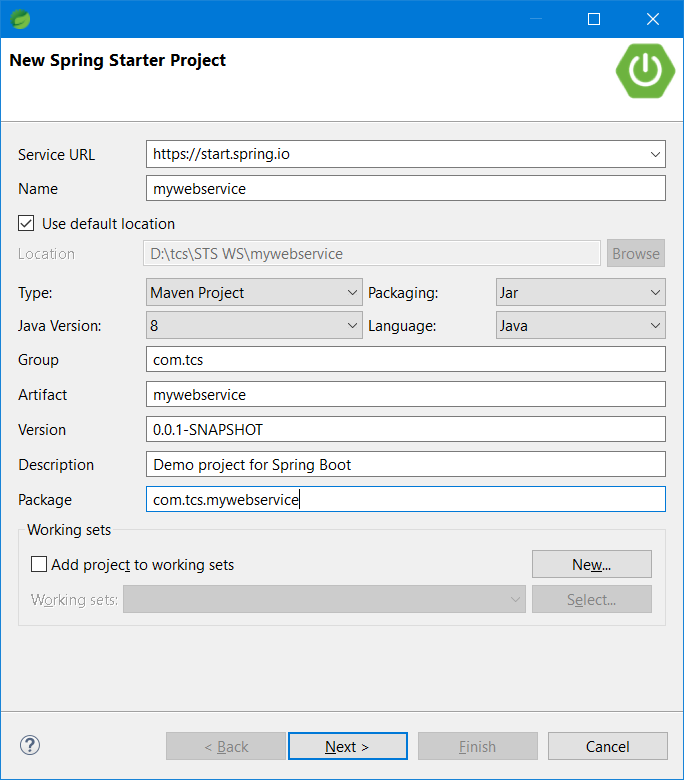


Step 2: File->New->Spring Starter Project



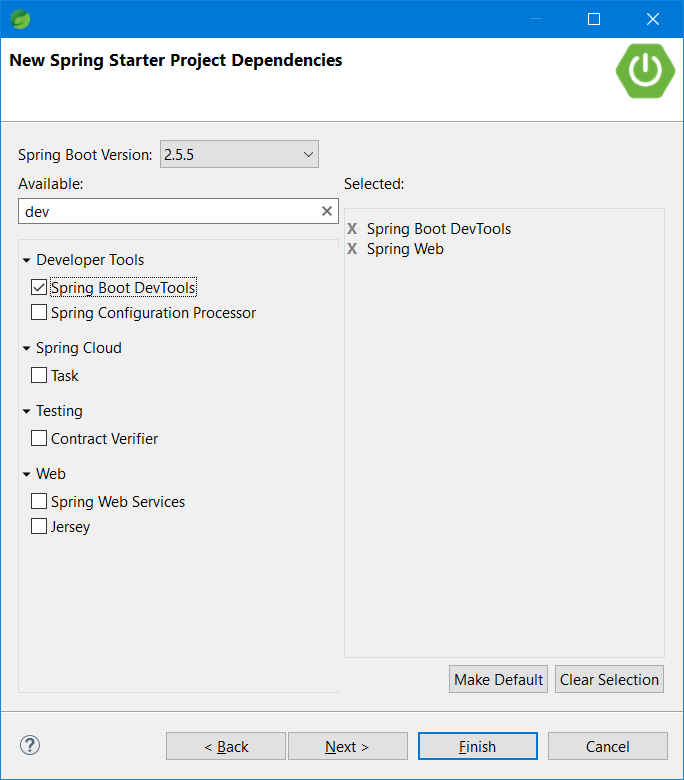
Step 3:

Service URL will be pointing to Spring Initializr URL (start.spring.io). Fill all the necessary details and click on “Next” Button.

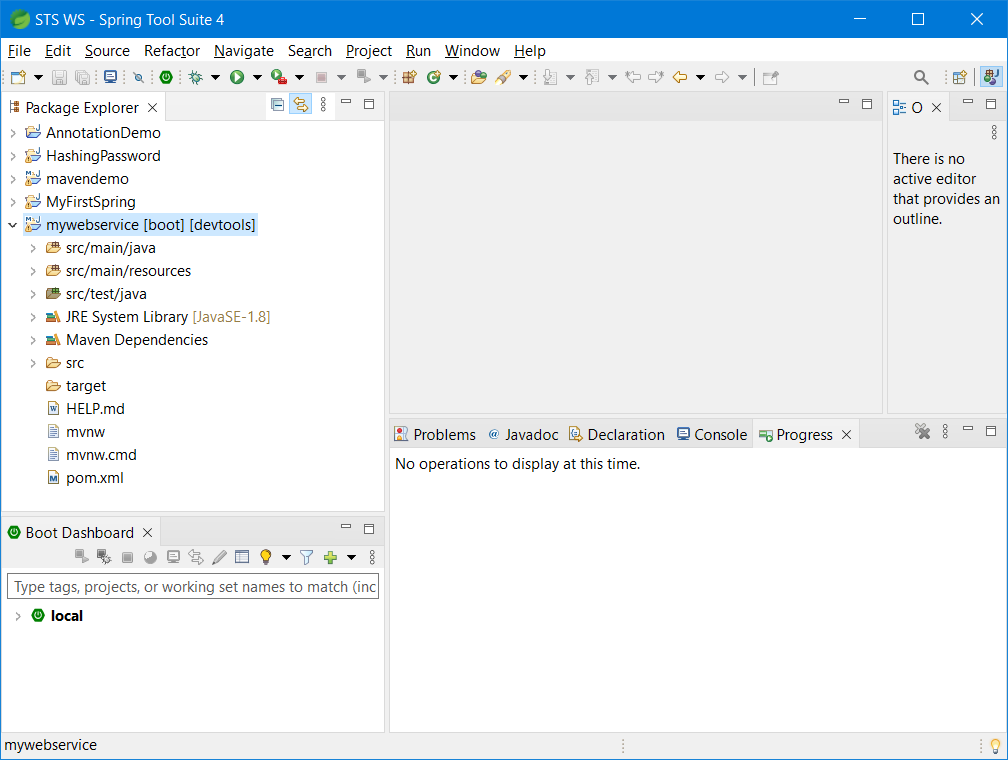


Step 4:

Select the required dependencies and click on “Finish” button. (devtools. Spring Web)

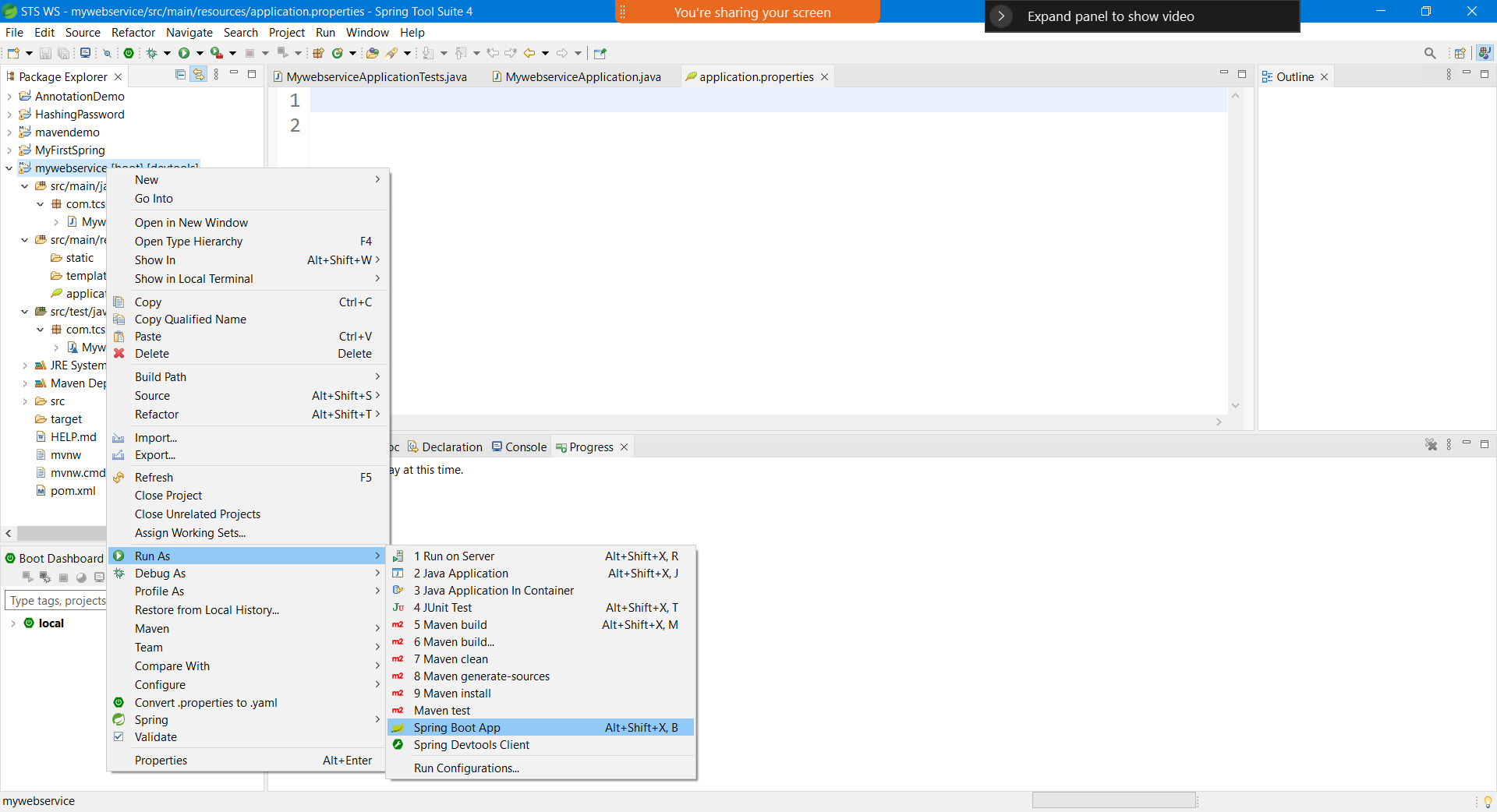


Step 5:

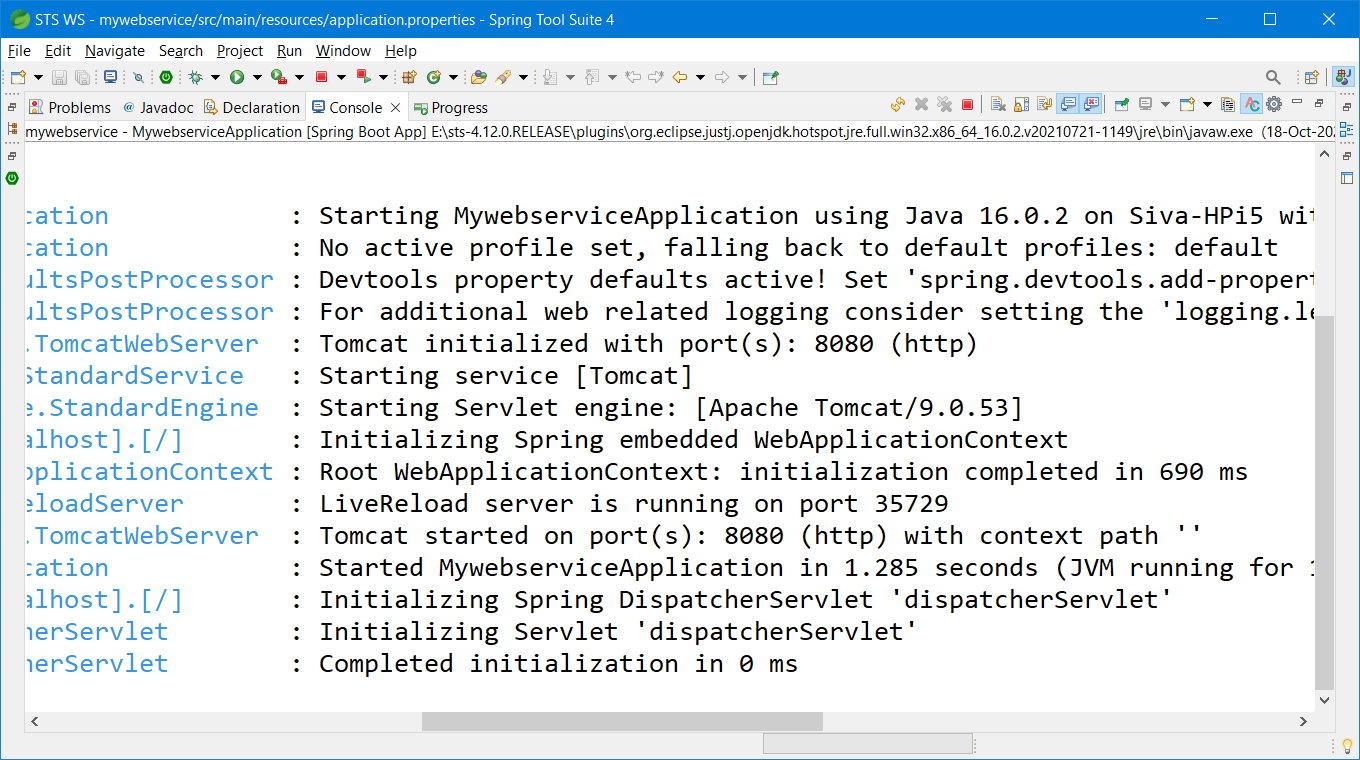


Step 6:

Right Click the project -> Run As -> Spring Boot App



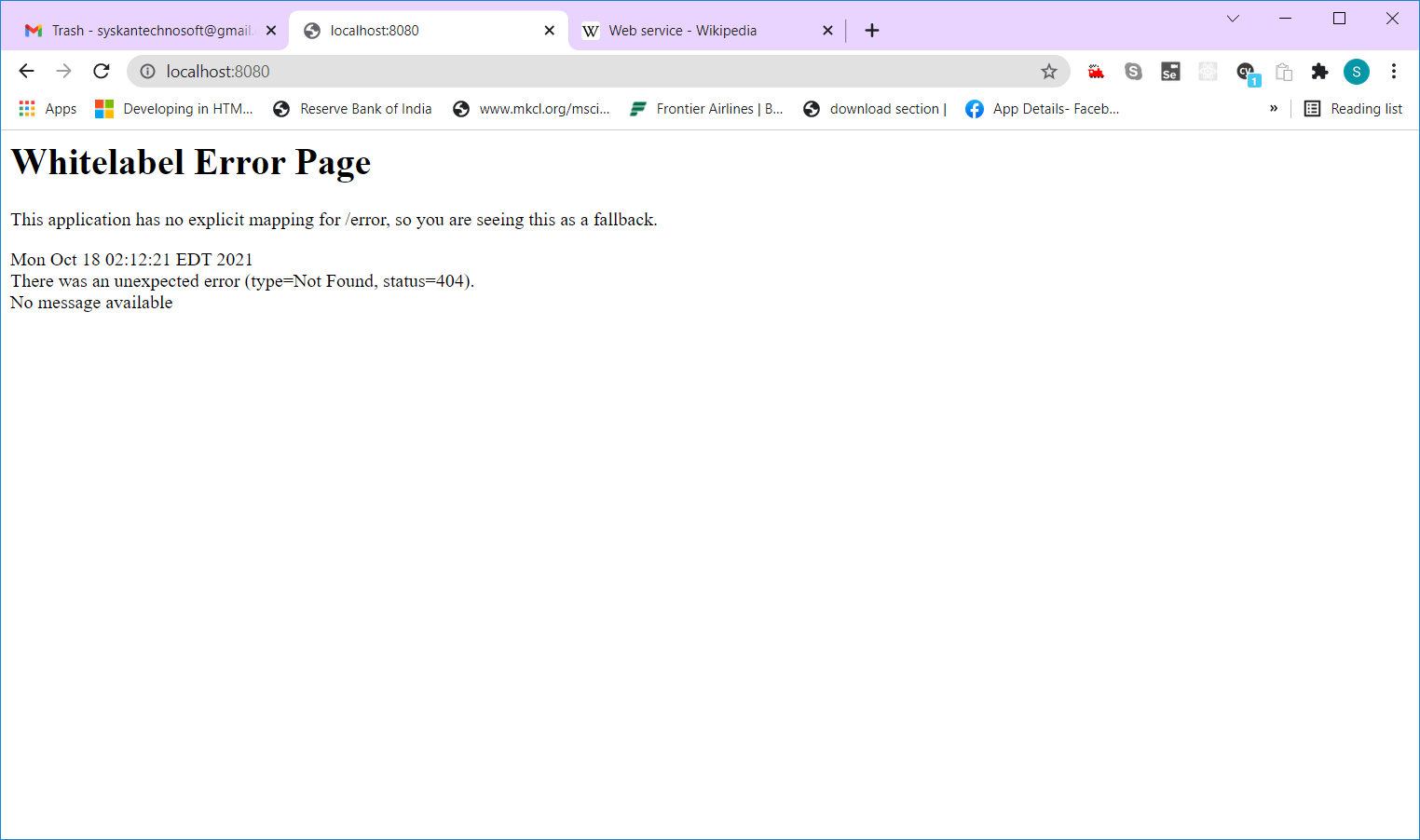
Step 7:



Step 8:

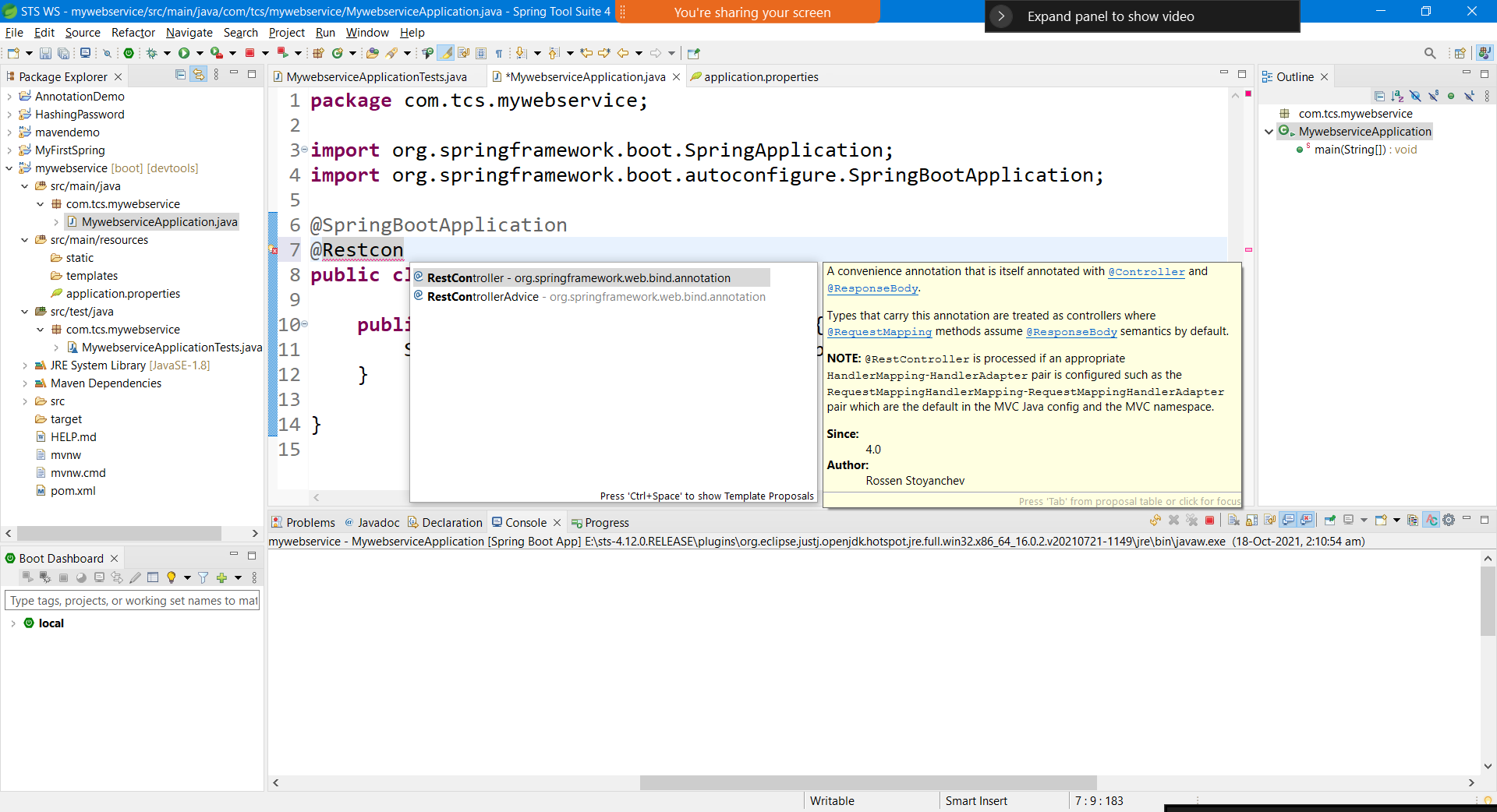
Open Browser (Chrome, Firefox, Edge….)

Type the following address in the address bar localhost:8080 and press enter



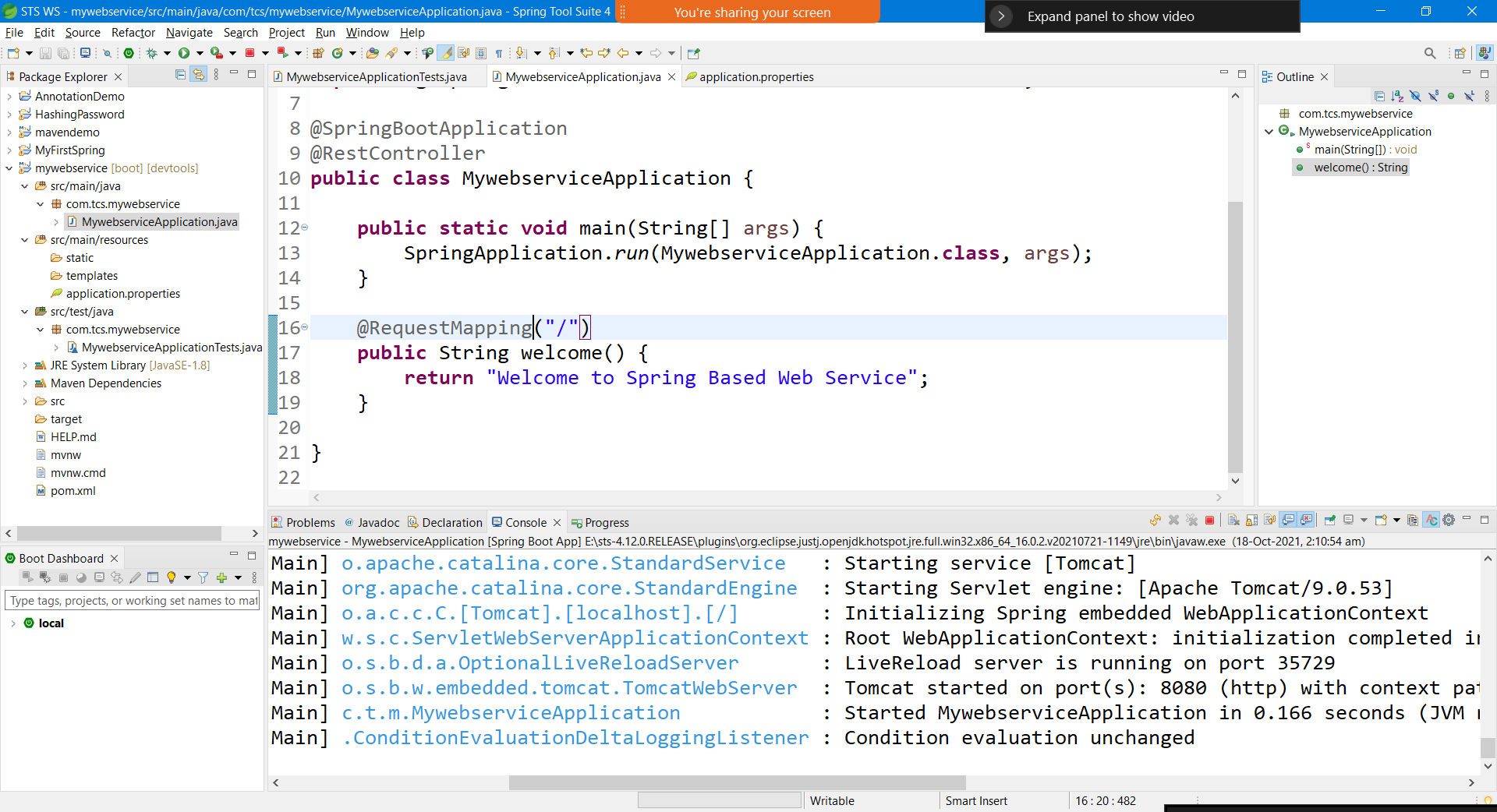
Step 9:

Add @RestController Annotation to the starter class as shown below



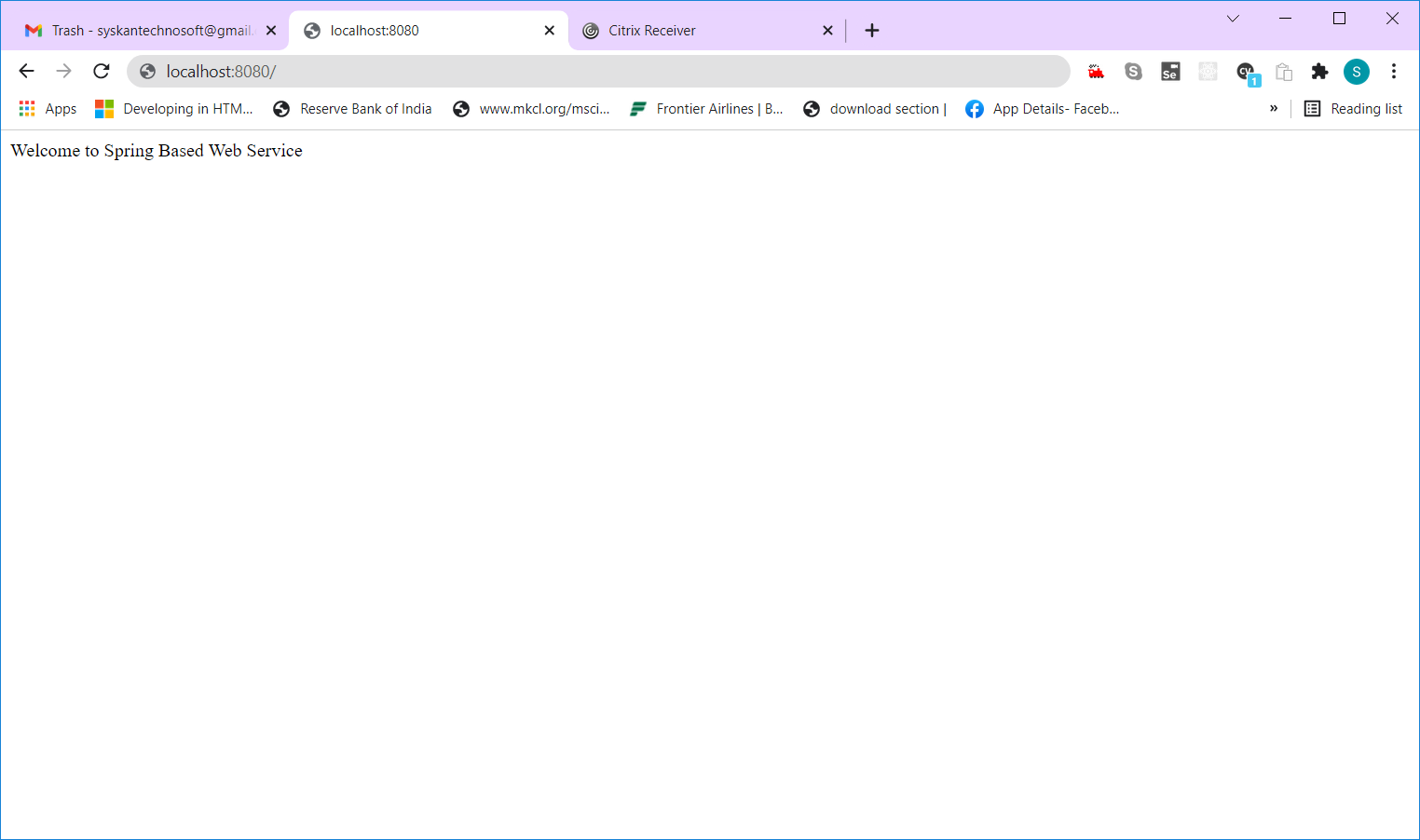
Step 10:

Add a new method with the annotation called @RequestMapping as shown below



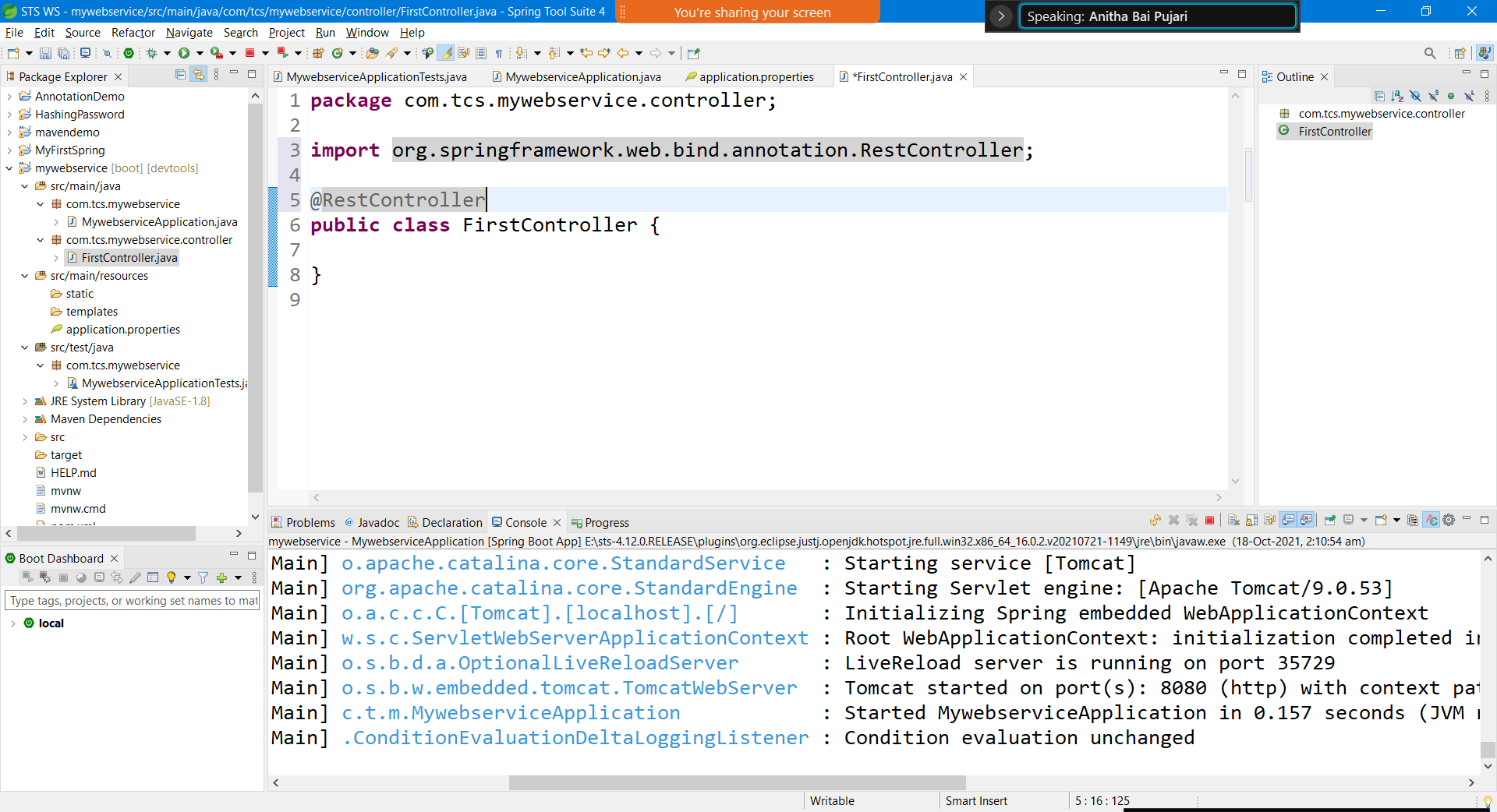
Step 11:

Open the browser and access the end point localhost:8080/ it will show the “Welcome to Spring Based Web Service” message in the window as shown below



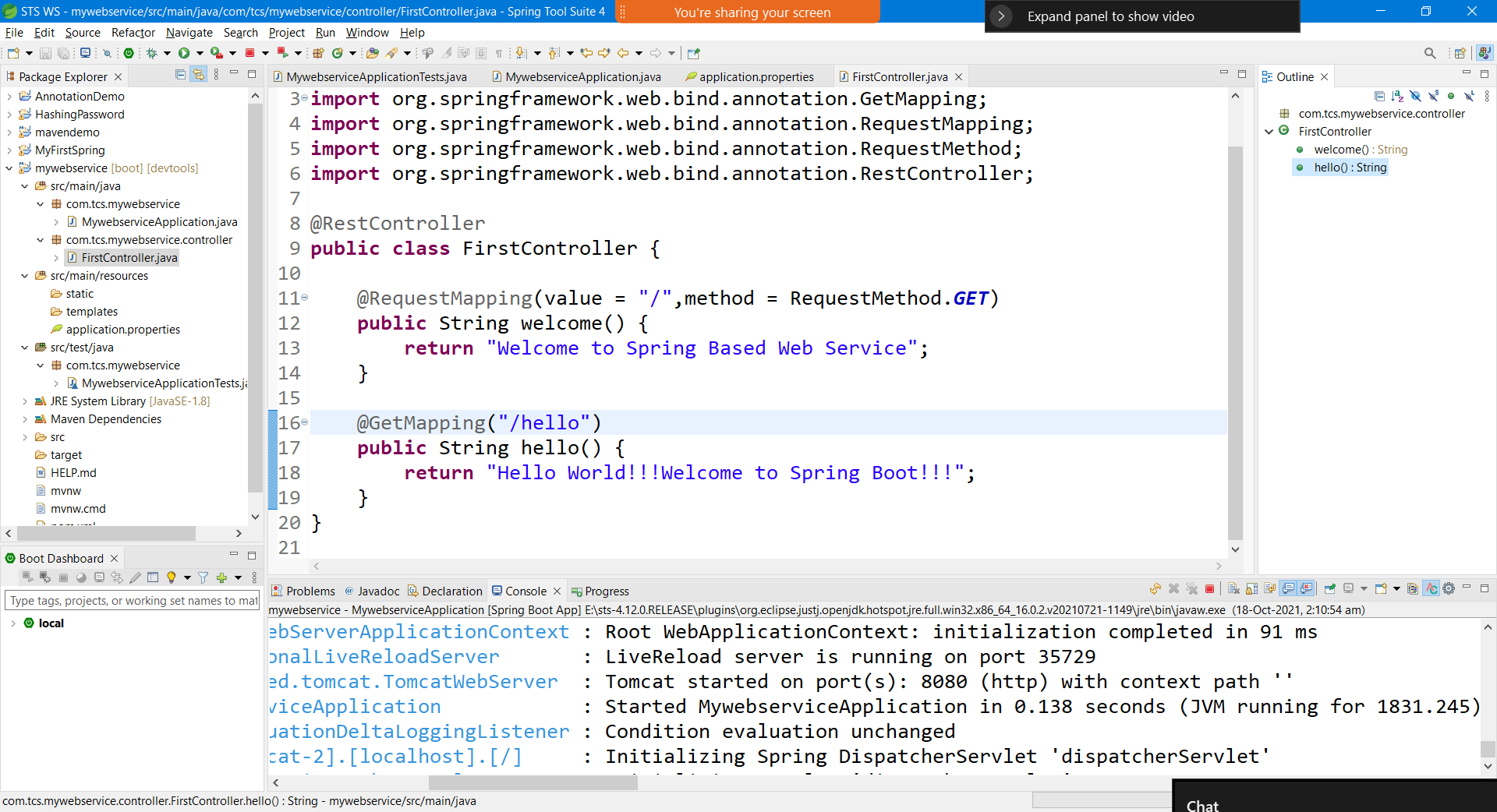
Step 12:

Add a new Class “FirstController” inside a new sub package “controller” as shown below



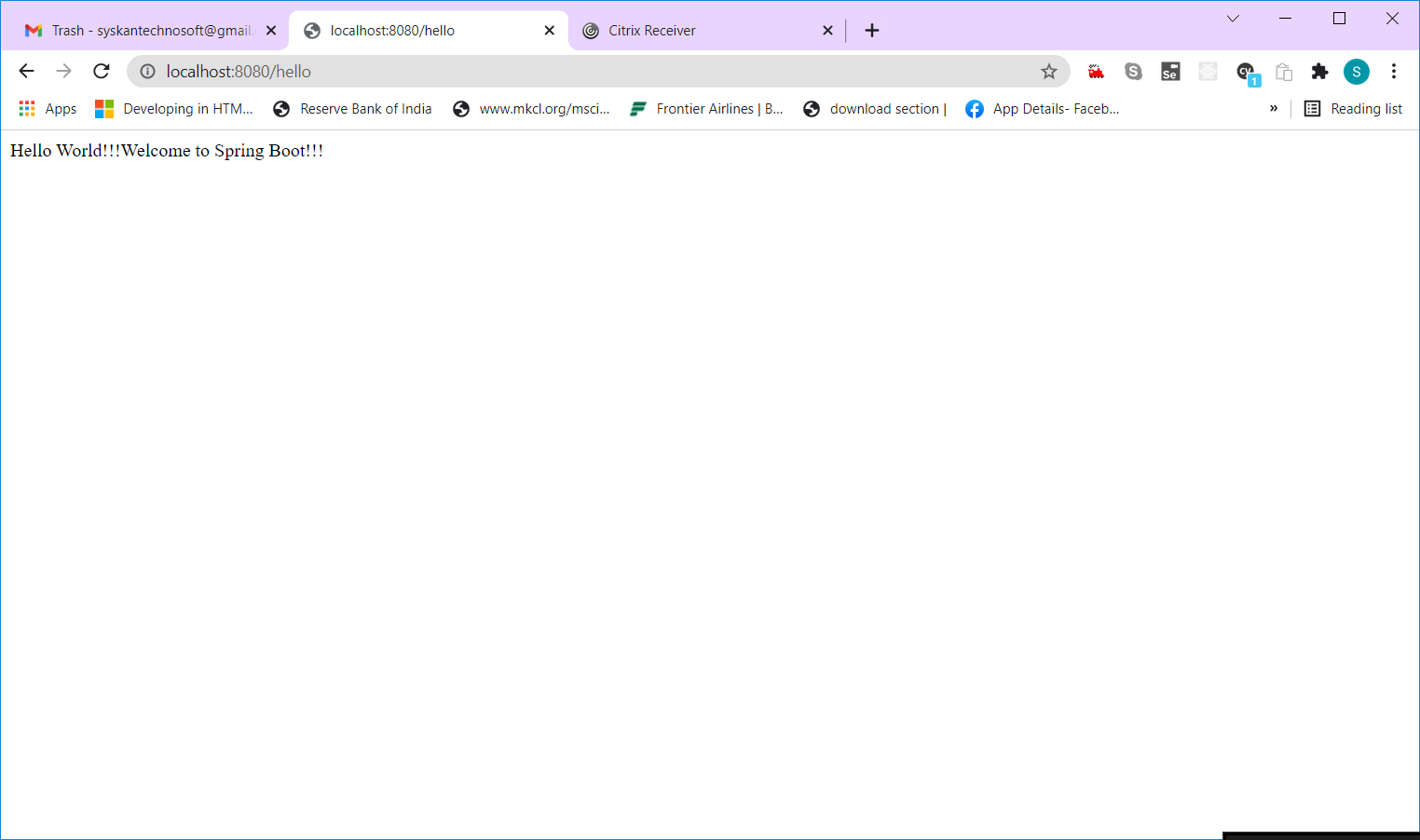
Step 13:

Move the welcome() method from the main class and add one more method as shown below.



Step 14:

Access the newly added end point /hello from your browser.



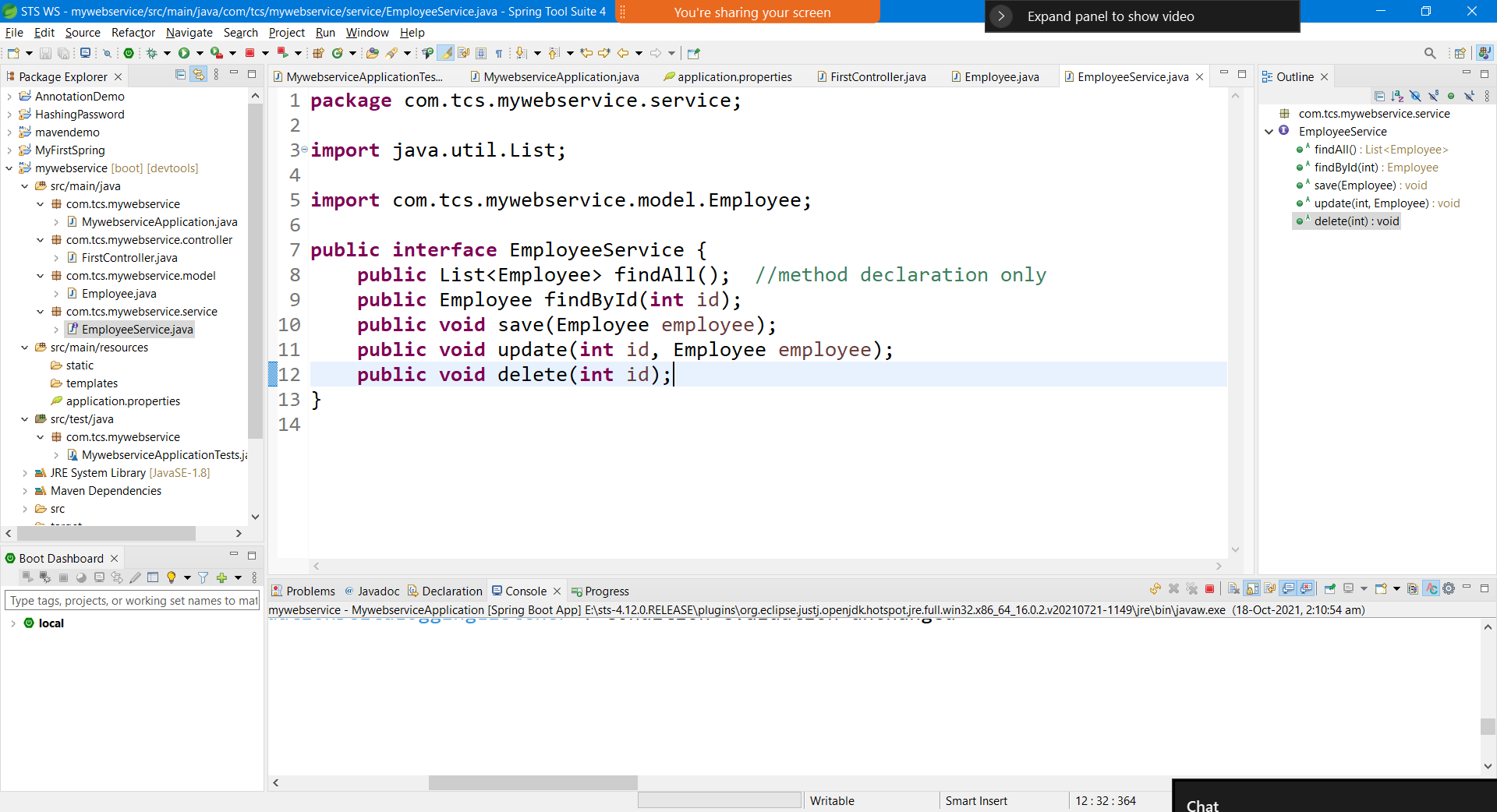
Step 15:

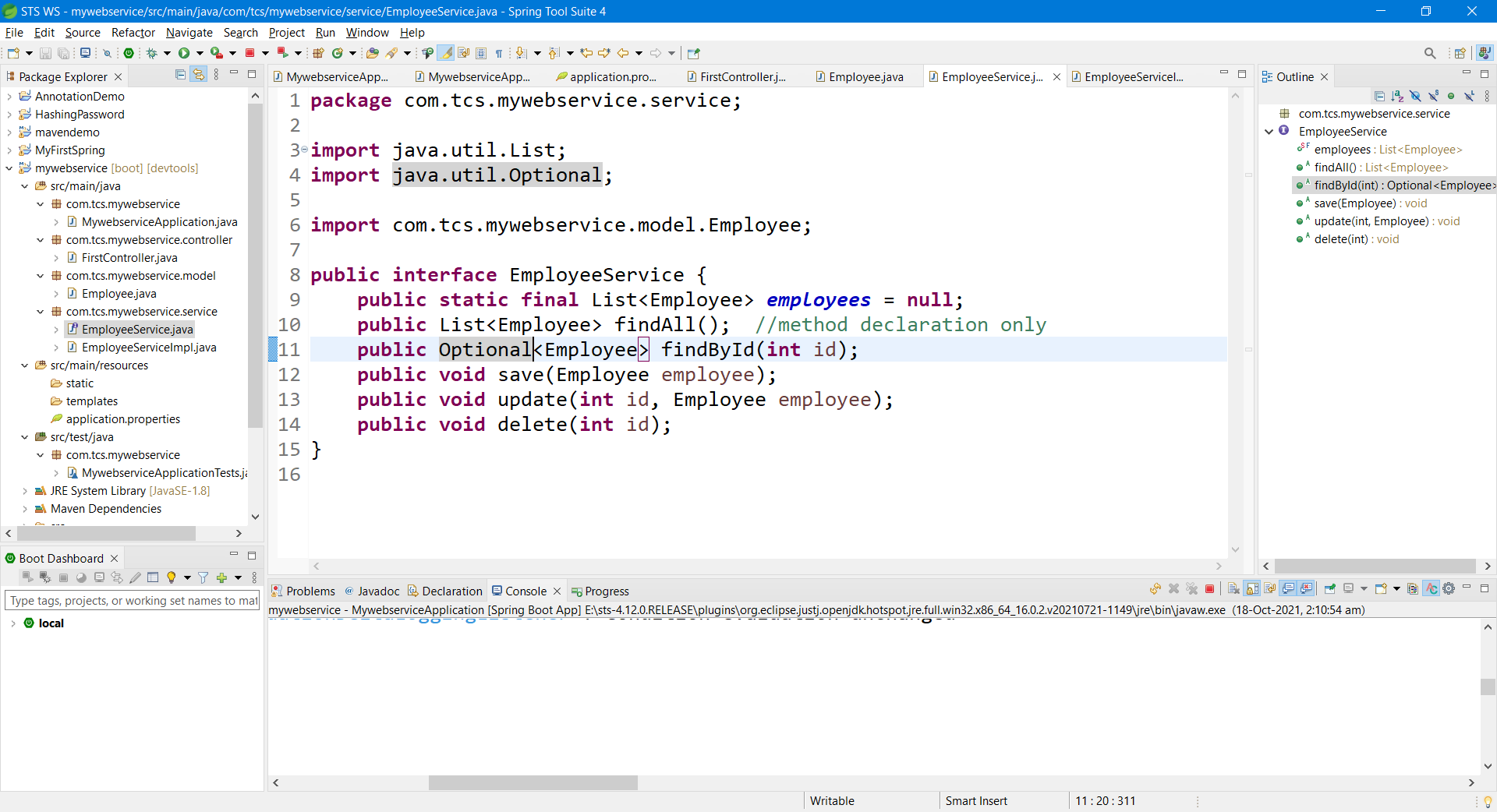
Create a Bean class called “Employee” in the sub package “model” as shown below



Step 16:

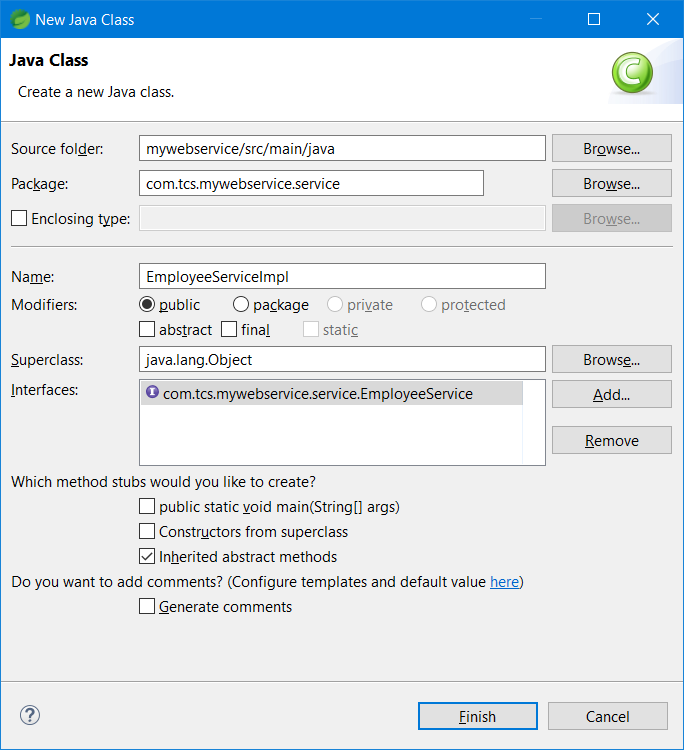
Add a new Interface called “EmployeeService” in the sub package “service”.





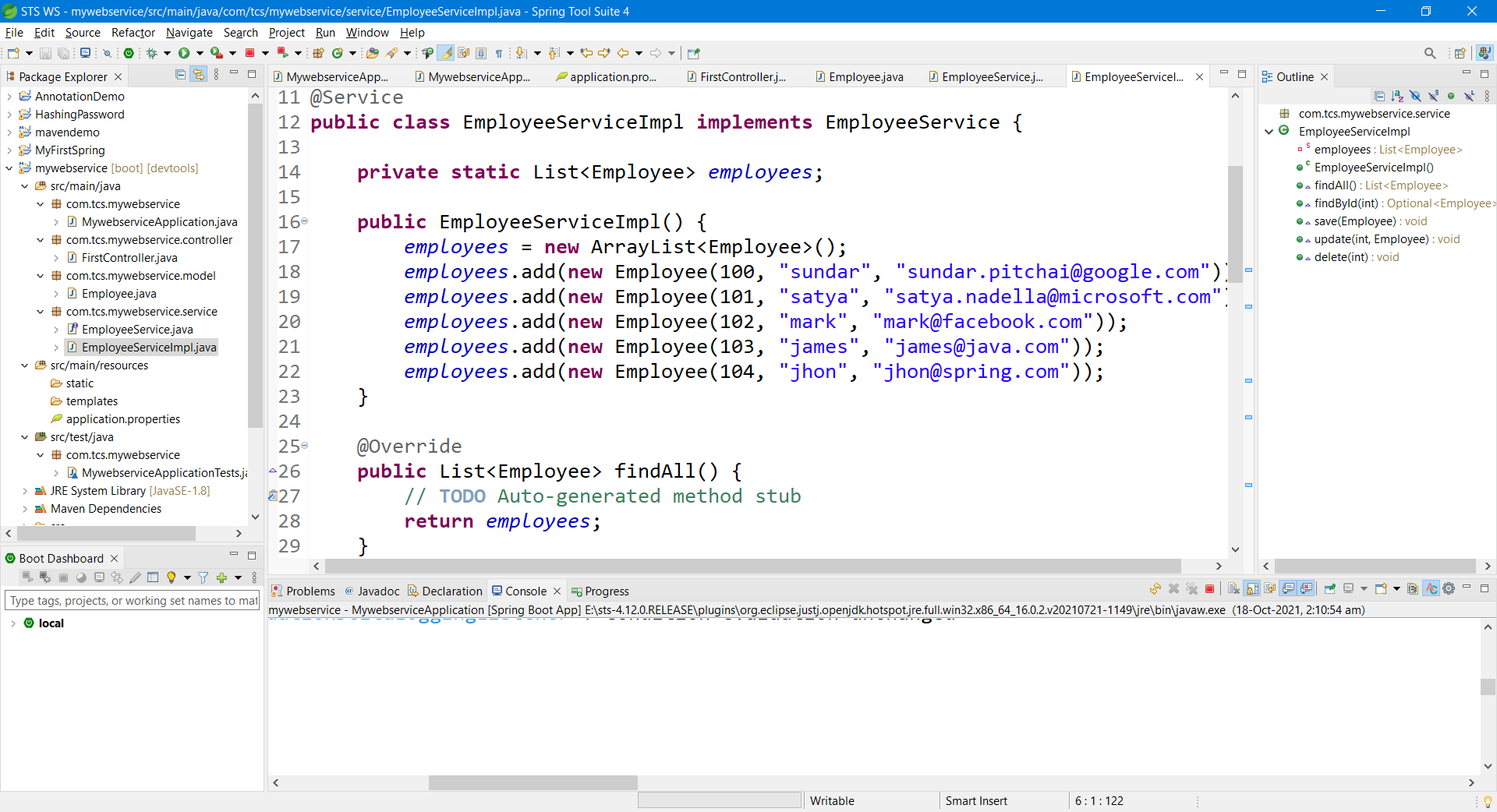
Step 17:

Add a new Class named “EmployeeServiceImpl” in the same sub package “service” as shown below



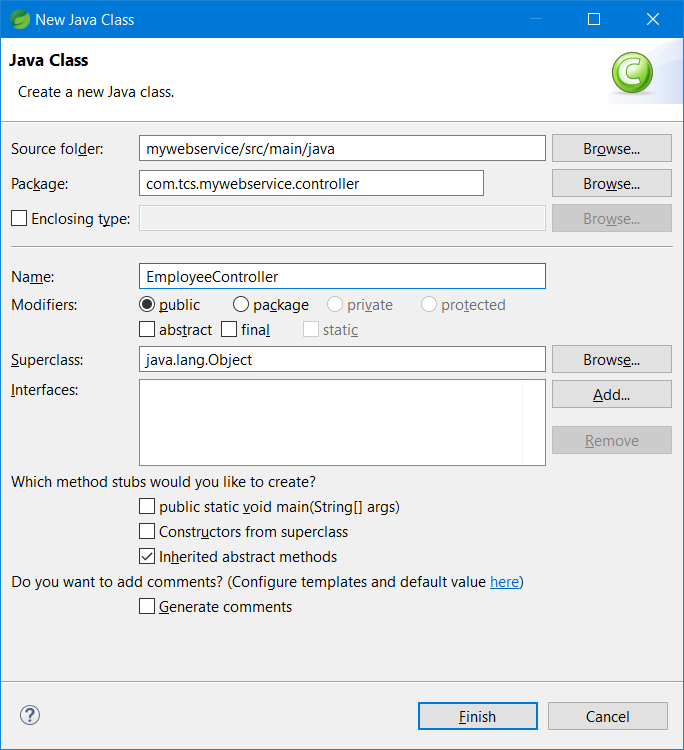
Step 18:

Provide implementation of all the abstract methods defined in “EmployeeService” interface.



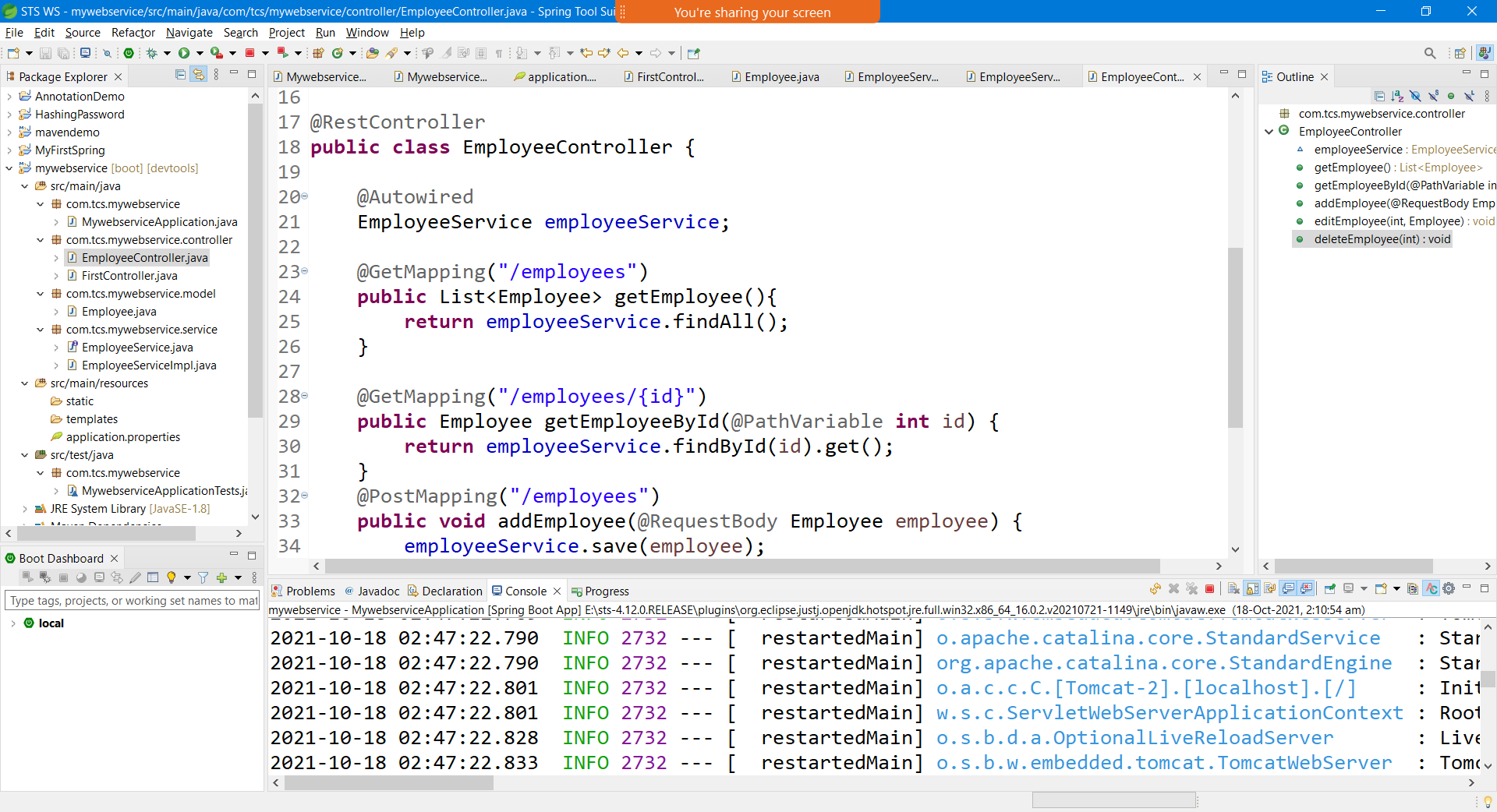
Step 19:

Add a new controller “EmployeeController” in the controller sub package as shown below.



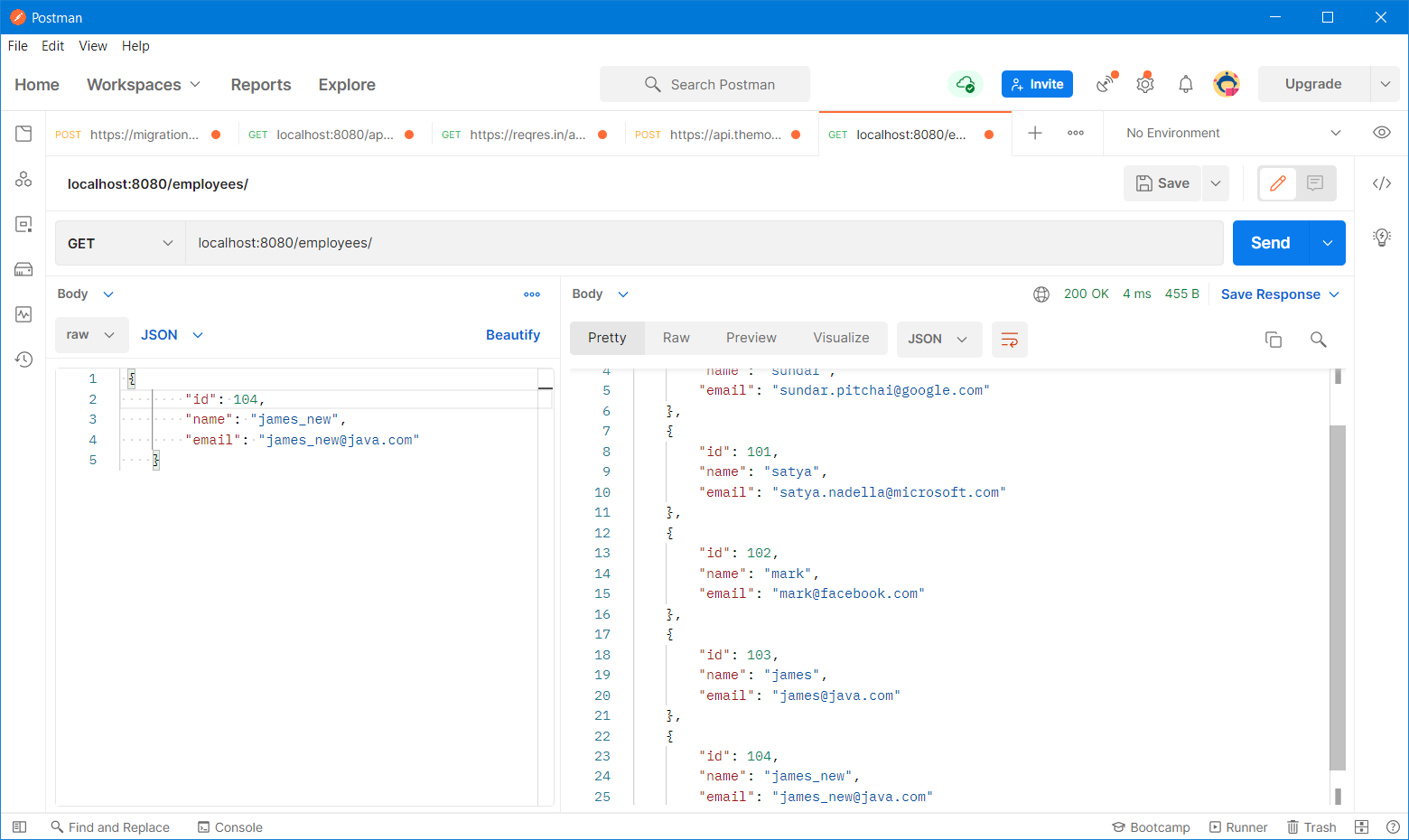
Step 20:

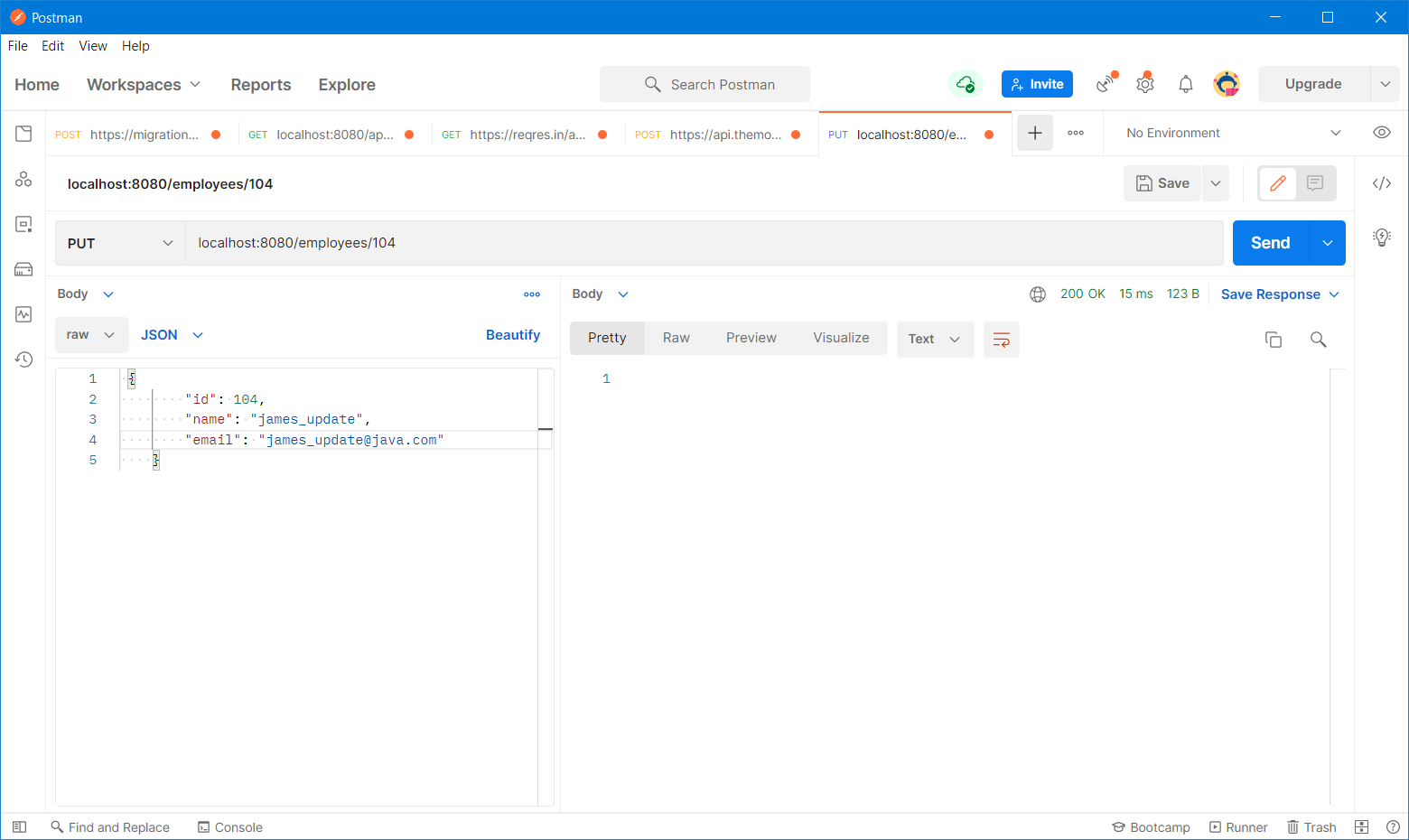
Add @RestController Annotation and all the End points for CRUD operation.



Step 21:

Test all the end points using postman.





End point - http method

/ - root

http methods

get – default method (getting/reading data from the service)

post – creating a new record (insert operation)

put – updating the existing record

delete – deleting the existing record

CRUD – Create(Insert)/Read/Update/Delete

Endpoints should be unique

Two end points with the same name and same http method – Not allowed

Two end points with the same name but different http method – Allowed

@Request Mapping – Specify method (Generic Annotation – use it for any http methods)

Specific Annotation

@GetMapping

@PostMapping

@PutMapping

@DeleteMapping