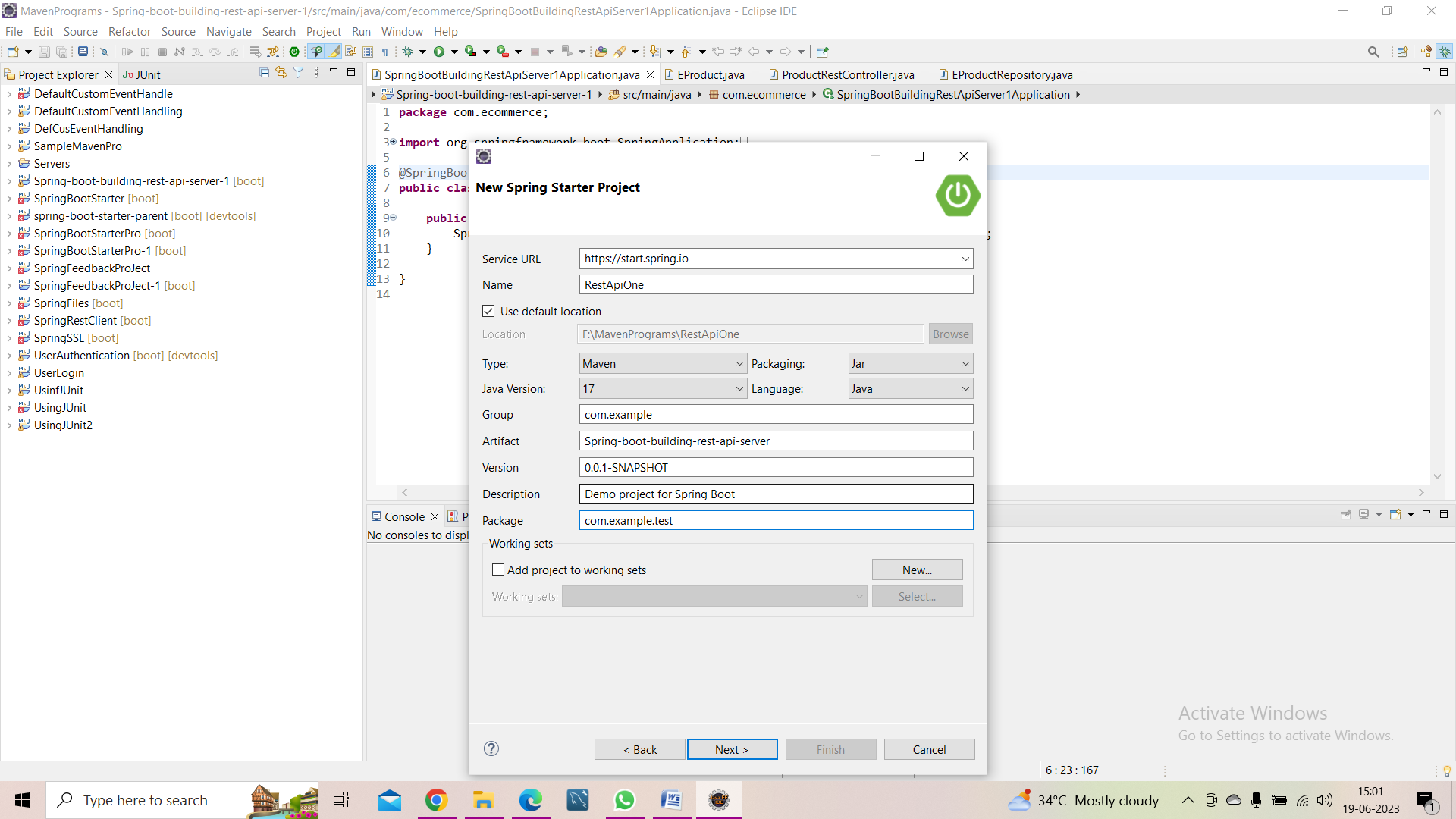
**Create a Project to Demonstrate Microservices with Spring Boot.**

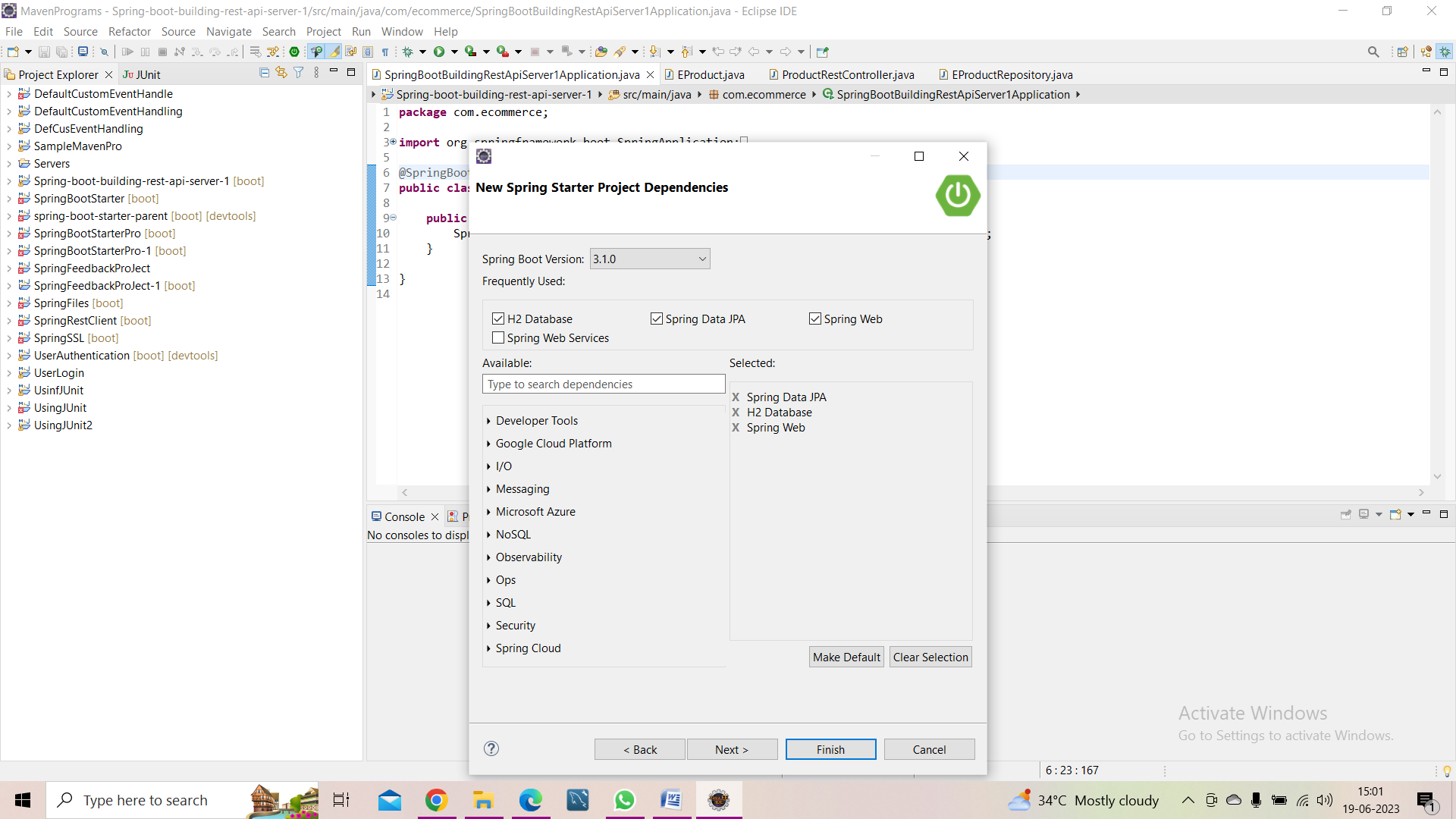
**1. Opening Spring Tool Suite and creating a new project using Spring Initializer**

● Fill in the required fields and call it ‘RestApiOne’.



**2.** Selecting the required project dependencies Spring Web, H2 Database, and Spring Data JPA.

Now, click on Finish.

****

* **It will creates the Main class: -**

package com.example.test;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestApiOneApplication {

public static void main(String[] args) {

SpringApplication.run(RestApiOneApplication.class, args);

}

}

**3. Creating an Entity Class: -**

package com.example.test;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

@Entity

public class PersonEntity {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

@Column(name = "id", updatable = false, nullable = false)

private Integer personId;

@Column

private String name;

@Column

private Integer age;

public PersonEntity() {

super();

}

public PersonEntity(Integer personId, String name, Integer age) {

super();

this.personId = personId;

this.name = name;

this.age = age;

}

public Integer getPersonId() {

return personId;

}

public void setPersonId(Integer personId) {

this.personId = personId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Integer getAge() {

return age;

}

public void setAge(Integer age) {

this.age = age;

}

}

**4.Creating a Repository Class**

package com.example.test;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface PersonRepository extends JpaRepository<PersonEntity, Integer> {

}

**5.** **Creating a Service Class: -**

package com.example.test;

import java.util.HashMap;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.web.client.RestTemplate;

@Service

public class PersonService {

@Autowired

PersonRepository personRepository;

RestTemplate restTemplate = new RestTemplate();

public PersonResponse getPerson(int personId){

final String uri = "http://localhost:8082/webapitwo/hobby/{personId}";

Map<String, Integer> params = new HashMap<String, Integer>();

params.put("personId", personId);

String result = restTemplate.getForObject(uri, String.class, params);

PersonEntity pe=personRepository.findById(personId).get();

PersonResponse pr=new PersonResponse();

pr.setPersonId(pe.getPersonId());

pr.setName(pe.getName());

pr.setAge(pe.getAge());

pr.setHobby(result);

return pr;

}

public void addPerson(PersonEntity pe){

personRepository.save(pe);

}

}

**6. Creating a Controller Class: -**

package com.example.test;

import org.springframework.beans.factory.annotation.Autowired;

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

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

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

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

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

@RestController

@RequestMapping(path = "/webapione")

public class PersonController {

@Autowired

PersonService personService;

@RequestMapping("/person/{personId}")

public PersonResponse getPerson(@PathVariable int personId){

return personService.getPerson(personId);

}

@RequestMapping(method=RequestMethod.POST, value="/person")

public void addPerson(@RequestBody PersonEntity pe ) {

personService.addPerson(pe);

}

}

**7. Creating a Response Class**

**package** com.example.test;

**public** **class** PersonResponse {

**private** Integer personId;

**private** String name;

**private** Integer age;

**private** String hobby;

**public** Integer getPersonId() {

**return** personId;

}

**public** **void** setPersonId(Integer personId) {

**this**.personId = personId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Integer getAge() {

**return** age;

}

**public** **void** setAge(Integer age) {

**this**.age = age;

}

**public** String getHobby() {

**return** hobby;

}

**public** **void** setHobby(String result) {

**this**.hobby = result;

}

}

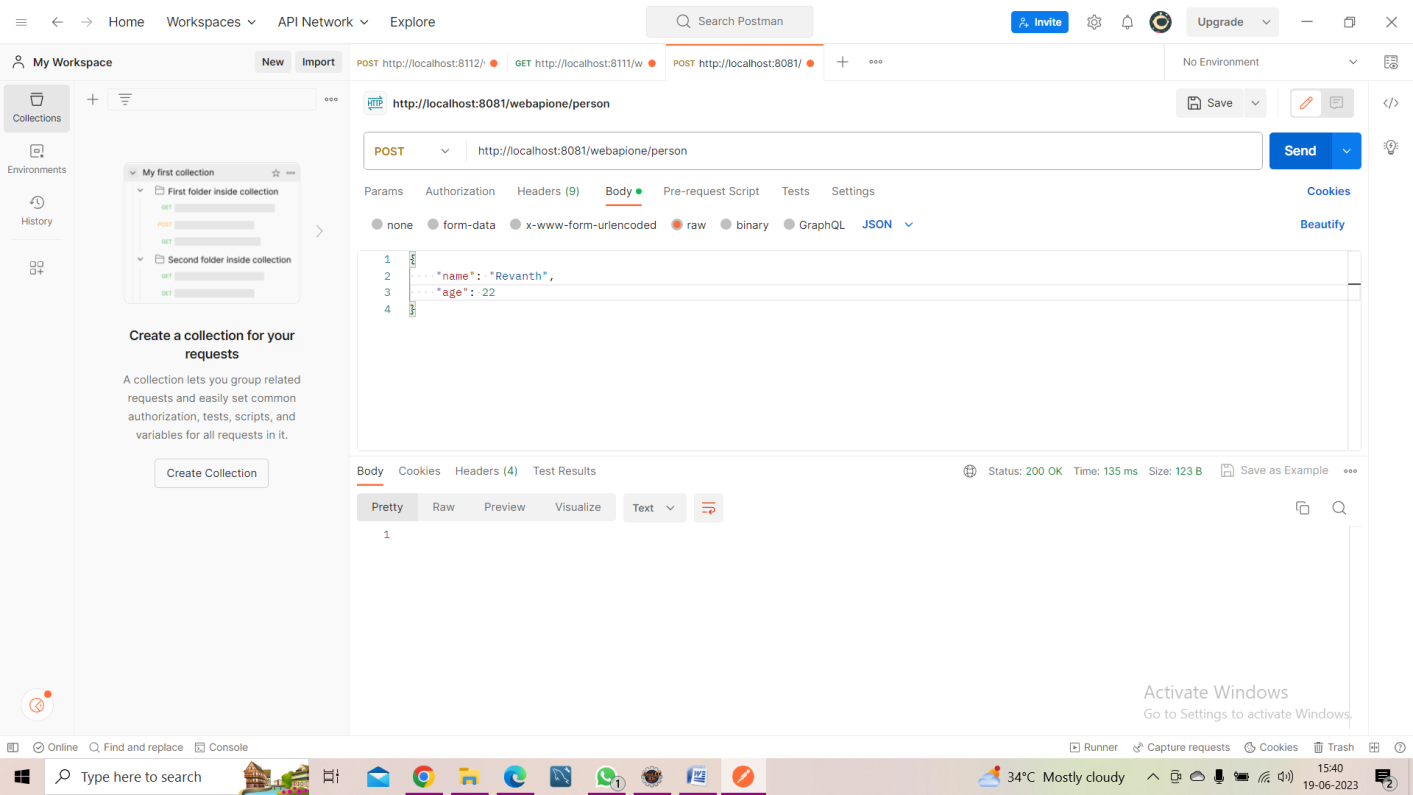
**8.** **Setting the port number for the project in the application properties: -**

server.port=8081

spring.application.name=RestApiOne

**9. Executing the project as ‘Spring Boot App’: -**

* It will run on port:8081 and make a POST request using POSTMAN



**10. Creating another project with the name ‘RestApiTwo’ as above**

* + **It will automatically create the main method: -**

package com.example.test;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class RestApiTwoApplication {

public static void main(String[] args) {

SpringApplication.run(RestApiTwoApplication.class, args);

}

}

**11. Creating an Entity Class: -**

**package** com.example.test;

**import** jakarta.persistence.Column;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.GeneratedValue;

**import** jakarta.persistence.GenerationType;

**import** jakarta.persistence.Id;

@Entity

**public** **class** HobbyEntity {

@Id

@GeneratedValue(strategy = GenerationType.***AUTO***)

@Column(name = "id", updatable = **false**, nullable = **false**)

**private** Integer id;

@Column

**private** Integer personId;

@Column

**private** String name;

**public** HobbyEntity() {

**super**();

}

**public** HobbyEntity(Integer personId, String name) {

**super**();

**this**.personId = personId;

**this**.name = name;

}

**public** Integer getPersonId() {

**return** personId;

}

**public** **void** setPersonId(Integer personId) {

**this**.personId = personId;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

}

**12. Creating a Repository Class: -**

**package** com.example.test;

**import** org.springframework.data.jpa.repository.JpaRepository;

**import** org.springframework.data.jpa.repository.Query;

**import** org.springframework.stereotype.Repository;

@Repository

**public** **interface** HobbyRepository **extends** JpaRepository<HobbyEntity, Integer> {

@Query("SELECT h.name FROM HobbyEntity h WHERE h.personId=:personId")

**public** String findByPersonId(Integer personId);

}

**13. Creating a Service Class: -**

**package** com.example.test;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

@Service

**public** **class** HobbyService {

@Autowired

HobbyRepository hobbyRepository;

**public** String findByPersonId(**int** personid){

**return** hobbyRepository.findByPersonId(personid);

}

**public** **void** addHobby(HobbyEntity he){

hobbyRepository.save(he);

}

}

**14. Creating a Controller Class: -**

**package** com.example.test;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.web.bind.annotation.PathVariable;

**import** org.springframework.web.bind.annotation.RequestBody;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

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

@RestController

@RequestMapping(path = "/webapitwo")

**public** **class** HobbyController {

@Autowired

HobbyService hobbyService;

@RequestMapping("/hobby/{personid}")

**public** String findByPersonId(@PathVariable **int** personid){

**return** hobbyService.findByPersonId(personid);

}

@RequestMapping(method=RequestMethod.***POST***, value="/hobby")

**public** **void** addHobby(@RequestBody HobbyEntity he ) {

hobbyService.addHobby(he);

}

}

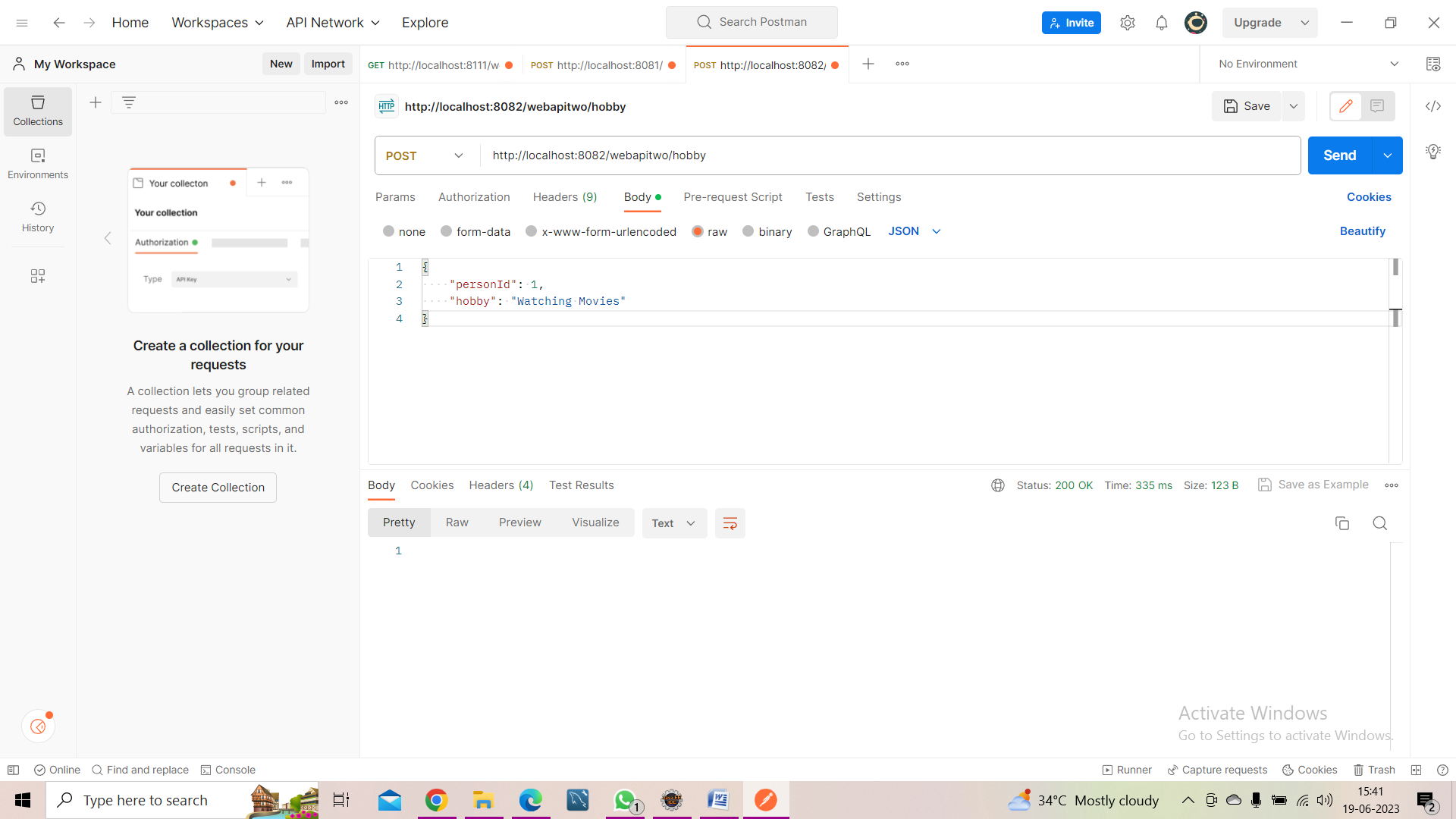
**15.** **Setting the port number for the project in application properties**

server.port=8082

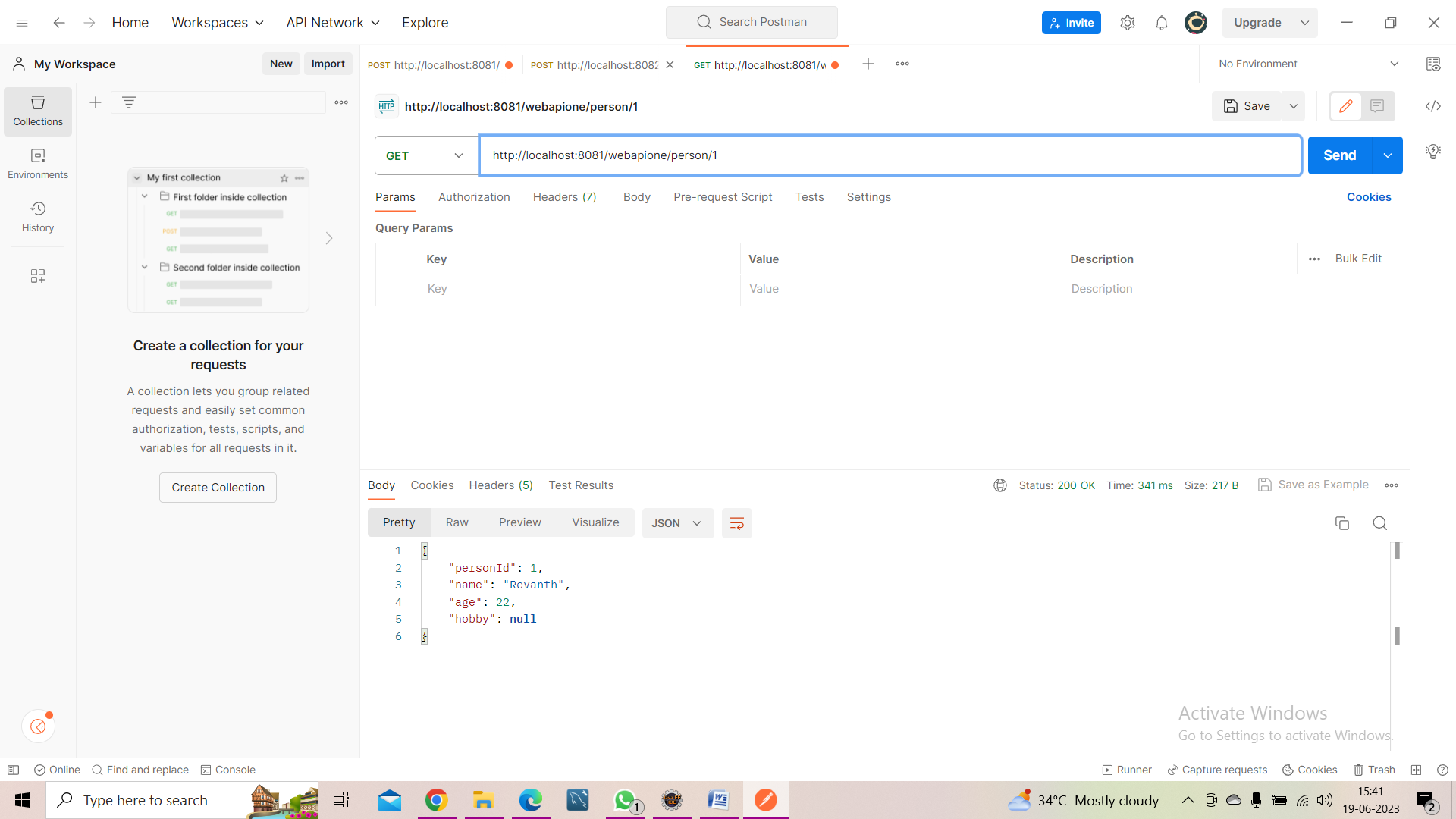
spring.application.name=RestApiTwo

**16**. **Executing the project ‘Spring Boot App’**

* It will run on port:8082 and making a POST request using Postman



**17. Making a GET request of ‘RestApiOne’ using Postman, it will be fetching the data from the existing running service ‘RestApiTwo’ and will be showing the communication of two microservices : -**

****

**18. Pushing the code to your GitHub repositories**

Step 1 : Open GitBash

Step 2 : Enter the below commands to push the files

* git init <foldername>
* cd <foldername>
* git add <filename>
* git commit –m “commit message”
* git push origin <branch>