**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

****

**Інститут: ІКНІ**

**Кафедра: ІСМ**

**ЗВІТ**

До лабораторної роботи **№ 4**

**На тему:** «Maven.Spring»

**З дисципліни**:«Прикладне програмування»

**Виконав:**

студент групи IT-21

Голинський Р.

**Прийняв:**

Крупа Д.В.

« **» Січня 2021**

**Львів - 2021**

[**Завдання**](https://github.com/strelovidniy/OOP_lab_8_7_1)

Використовуючи технологію Spring створити веб-сервер, який повинен рахувати вартість туру.

Завдання:

1. Створити структуру класів, які відповідатимуть завданню.

2. Дані можна зберігати in-memory або у файлі.

3. Реалізувати мінімум 5 кінцевих точок (endpoints), які будуть працювати відповідно до завдання. Наприклад, “/get-all-cities”, “/calculate-tour”, “/get-all-users”.

4. Створити глобальний ExceptionHandler та використати unchecked exception із лабораторної роботи №3.

5. Реалізувати перевірку коректності введених даних користувачем (валідацію) там де це необхідно.

6. Допускається вибір власної теми, крім “інтернет-магазину”

CustomGlobalExceptionHandler.java:

package shop;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class ShopApplication  
{  
 public static void main(String[] args)  
 {  
 SpringApplication.*run*(ShopApplication.class, args);  
 }  
  
}

HttpException.java:

package shop;  
  
import org.springframework.http.HttpStatus;  
  
public class HttpException extends RuntimeException  
{  
 private final HttpStatus status;  
  
 public HttpException(HttpStatus status, String errorMessage)  
 {  
 super(errorMessage);  
 this.status = status;  
 }  
  
 public HttpStatus getStatus()  
 {  
 return this.status;  
 }  
}

Item.java:

package shop;  
  
import org.springframework.http.HttpStatus;  
  
import java.util.UUID;  
  
public class Item  
{  
 private final String id;  
 private String name;  
 private Integer price;  
  
 Item(String name, int price)  
 {  
 this.name = name;  
 this.price = price;  
 this.id = UUID.*randomUUID*().toString();  
 }  
  
 public String getName()  
 {  
 return name;  
 }  
  
 public Integer getPrice()  
 {  
 return price;  
 }  
  
 public void setName(String name)  
 {  
 this.name = name;  
 }  
  
 public void setPrice(Integer price)  
 {  
 this.price = price;  
 }  
  
 public String getId()  
 {  
 return id;  
 }  
  
 public void validate()  
 {  
 boolean isNameValid = this.name instanceof String && this.name.length() != 0;  
 boolean isPriceValid = this.price instanceof Integer && this.price >= 0;  
  
 if (!isNameValid)  
 {  
 throw new HttpException(HttpStatus.*BAD\_REQUEST*, "Invalid name");  
 }  
  
 if (!isPriceValid)  
 {  
 throw new HttpException(HttpStatus.*BAD\_REQUEST*, "Invalid price");  
 }  
 }  
}

ItemController.java:

package shop;  
  
import org.springframework.http.HttpStatus;  
import org.springframework.web.bind.annotation.\*;  
import org.springframework.beans.factory.annotation.Autowired;  
  
@RestController  
@RequestMapping("/items")  
public class ItemController  
{  
 @Autowired  
 private ItemService itemService;  
  
 @PostMapping  
 @ResponseStatus(HttpStatus.*CREATED*)  
 void createItem(@RequestBody Item item)  
 {  
 item.validate();  
  
 itemService.save(item);  
 }  
  
 @GetMapping  
 Item[] getAllItems()  
 {  
 return itemService.getAll();  
 }  
  
 @GetMapping("/{itemId}")  
 Item getItemById(@PathVariable("itemId") String itemId)  
 {  
 return itemService.getById(itemId);  
 }  
  
 @PutMapping("/{itemId}")  
 void updateItemById(@PathVariable("itemId") String itemId, @RequestBody Item item)  
 {  
 item.validate();  
  
 itemService.updateById(itemId, item);  
 }  
  
 @DeleteMapping("/{itemId}")  
 void deleteItemById(@PathVariable("itemId") String itemId)  
 {  
 itemService.deleteById(itemId);  
 }  
}

ShopApplication.java:

package shop;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class ShopApplication  
{  
 public static void main(String[] args)  
 {  
 SpringApplication.*run*(ShopApplication.class, args);  
 }  
  
}

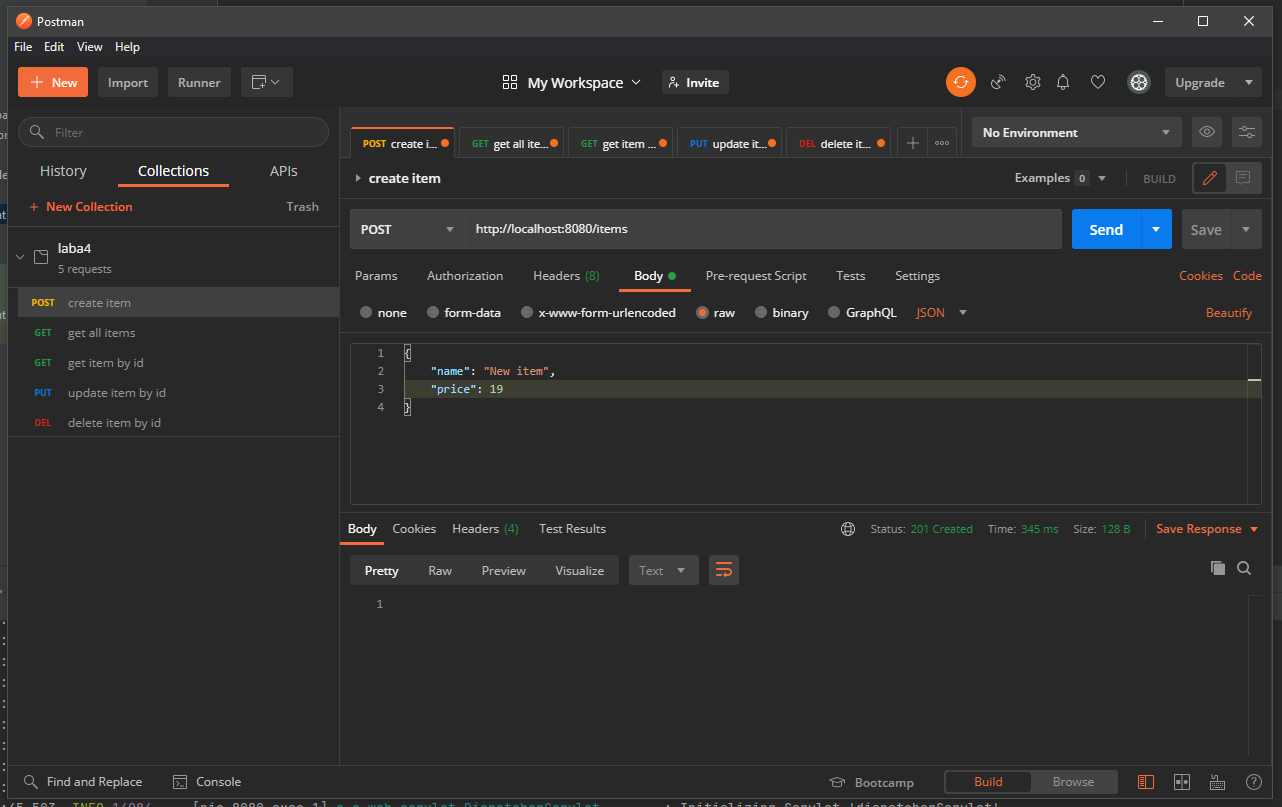


Рисунок 1

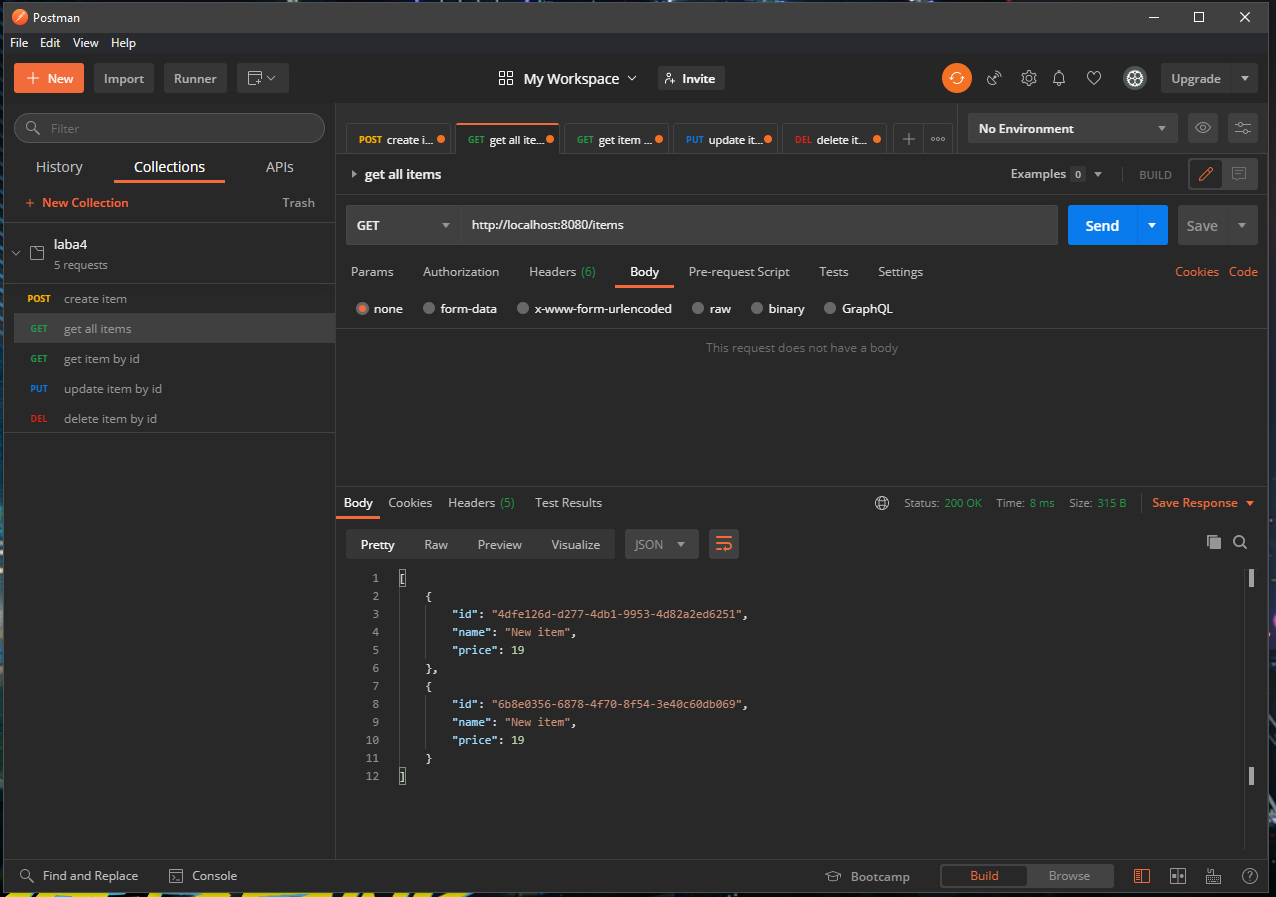


Рисунок 2

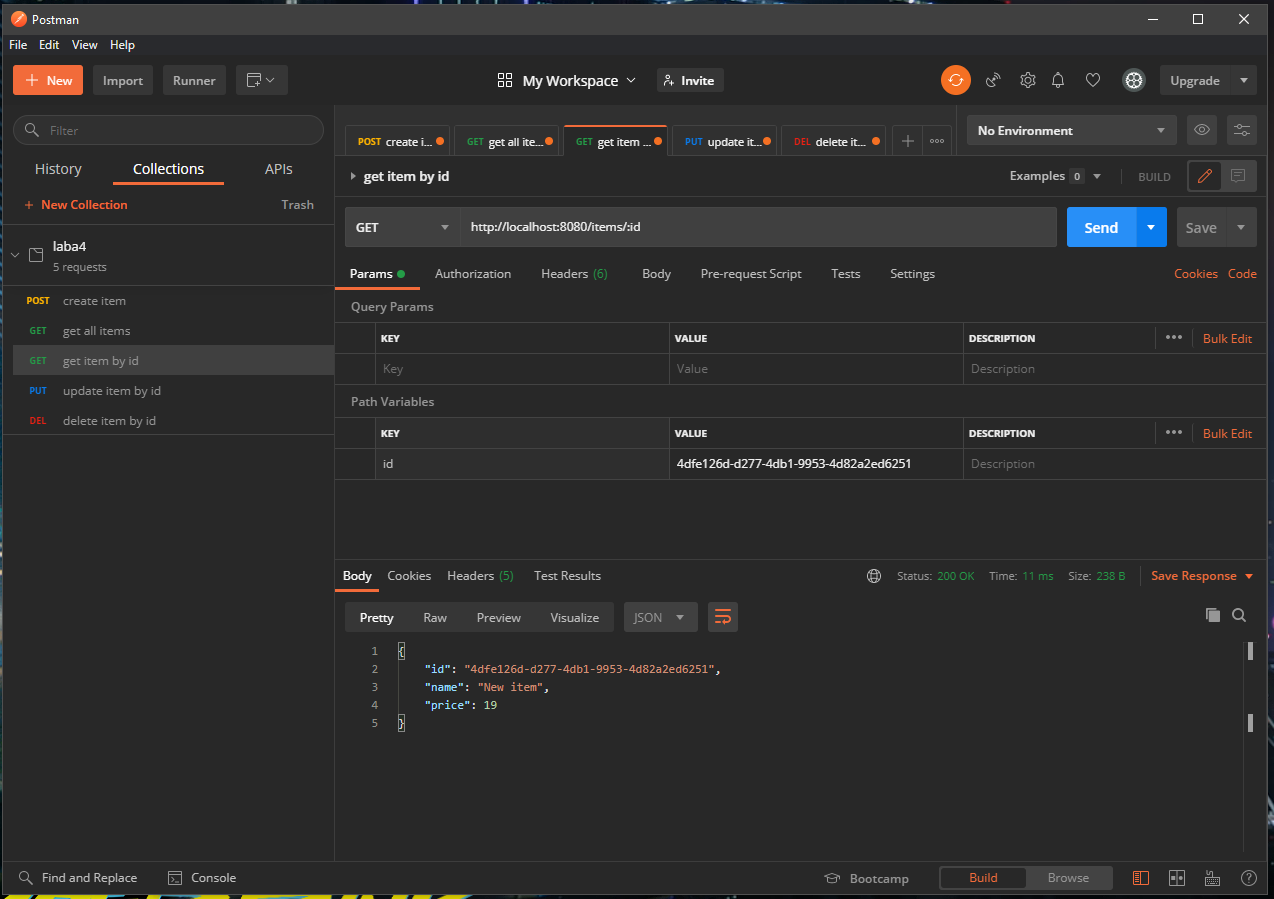


Рисунок 3

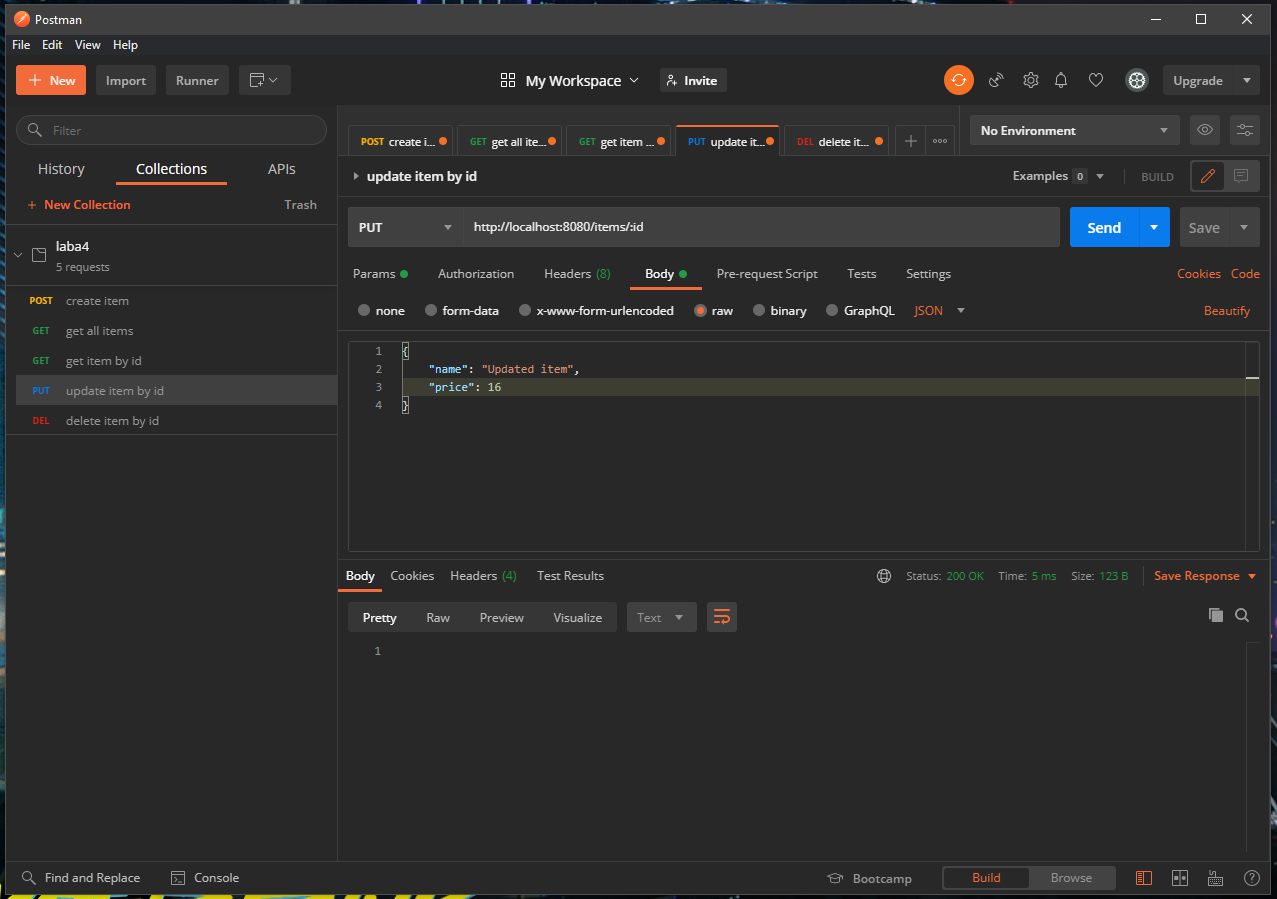


Рисунок 4

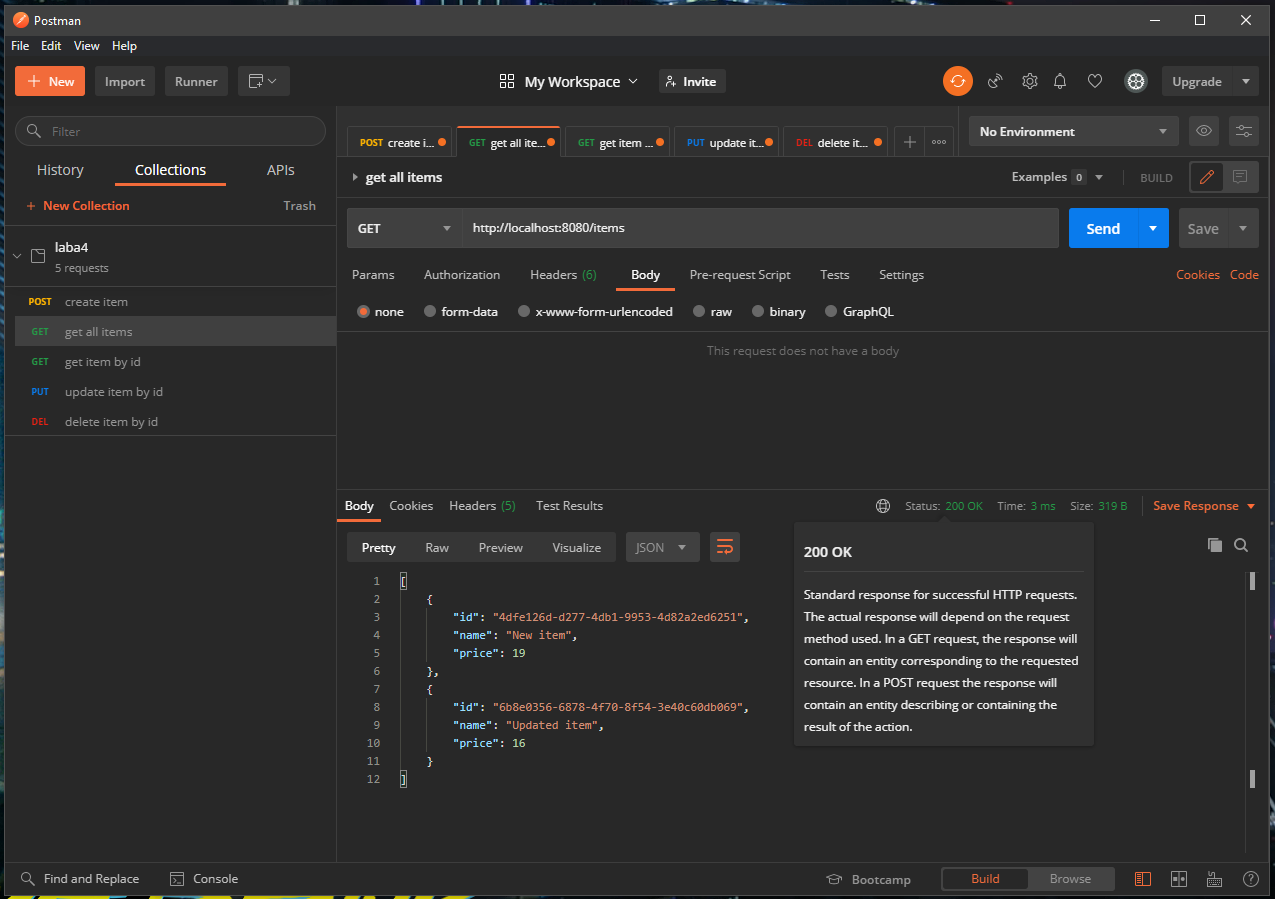


Рисунок 5

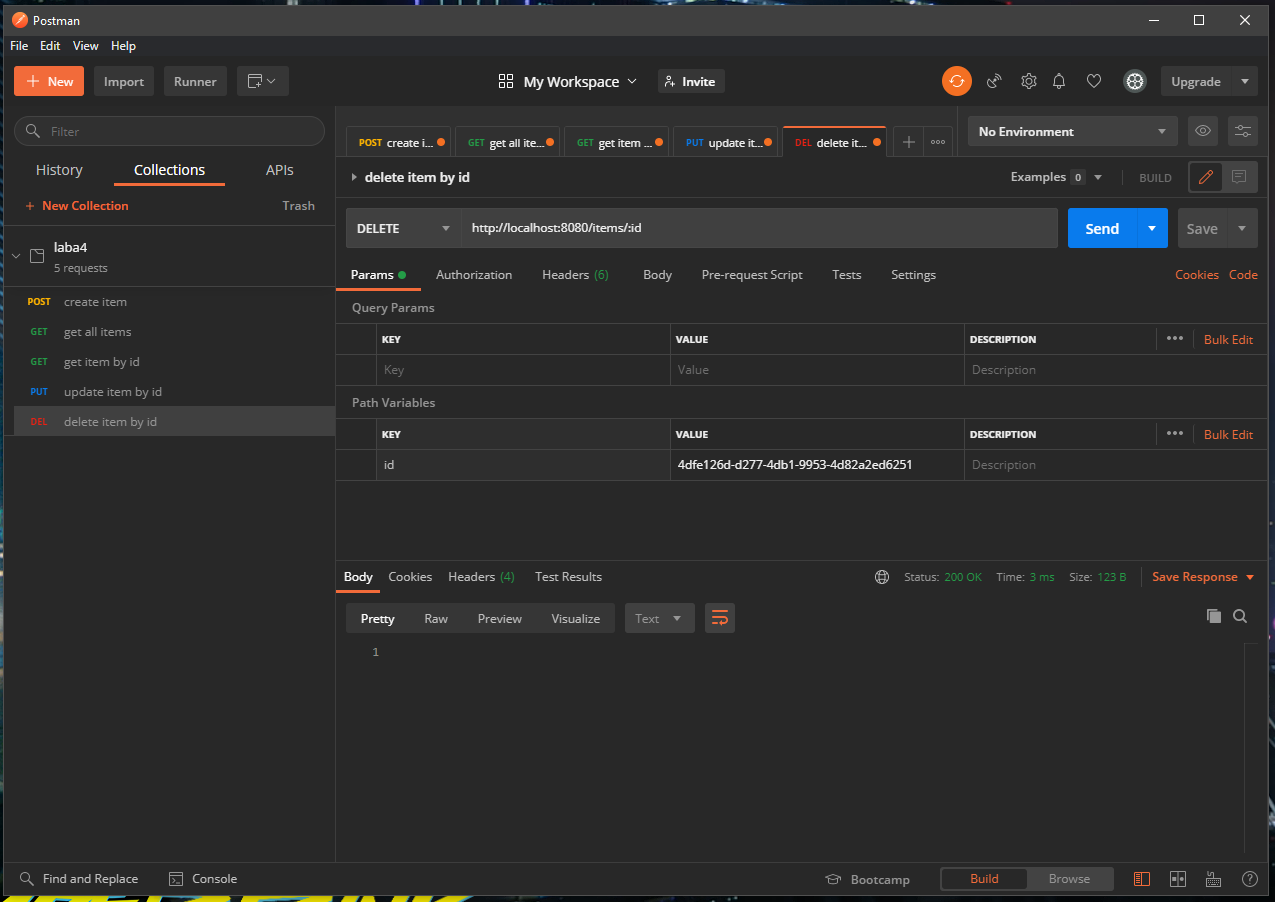


Рисунок 6

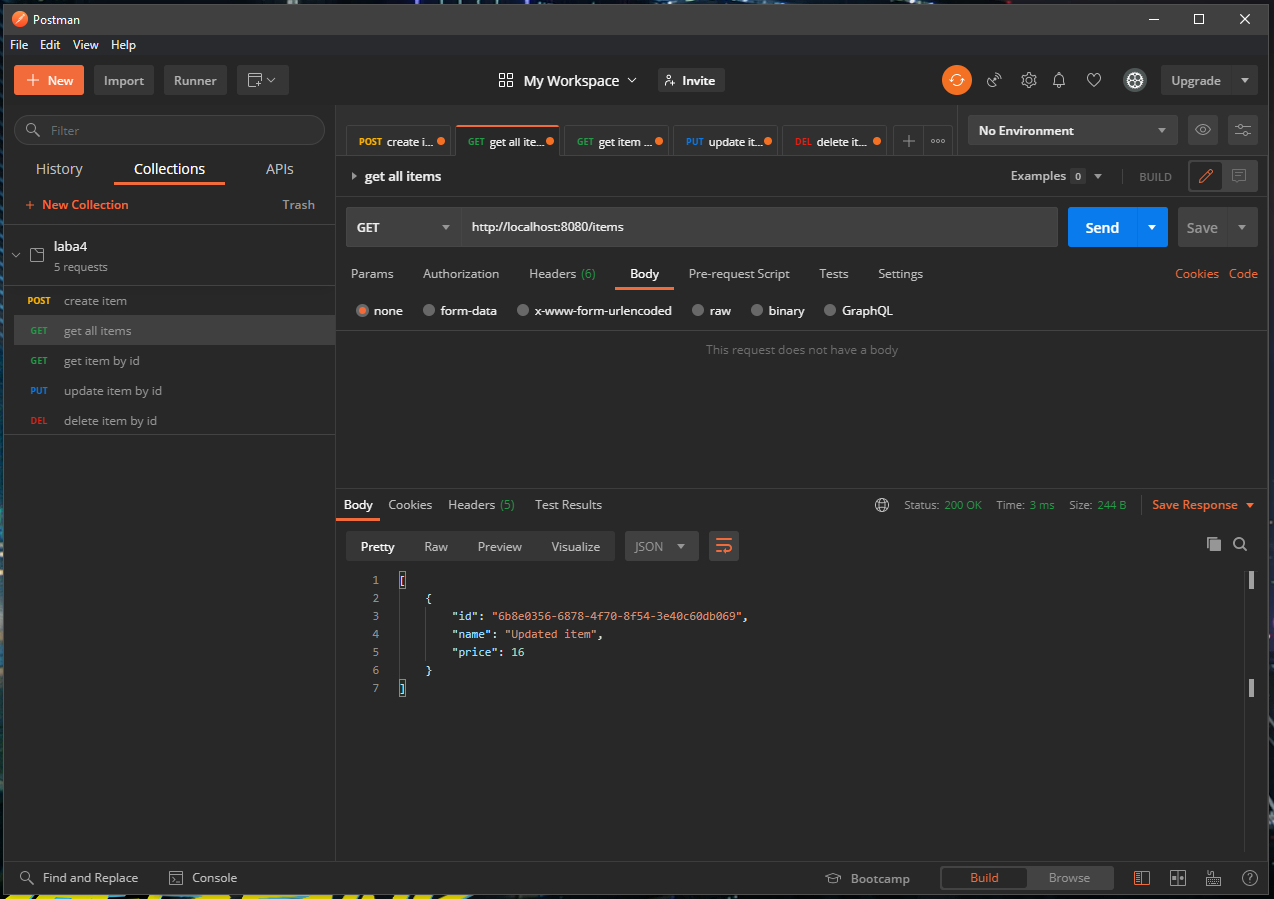


Рисунок 7

**Висновок:** використовуючи технологію Spring створив веб-сервер, який рахує вартість туру. Створив структуру класів, які відповідають завданню. Створив глобальний ExceptionHandler та використав unchecked exception.