Міністерство освіти і науки

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій  
Кафедра автоматизованих систем управління

**Звіт**

до лабораторної роботи №4

*з дисципліни*

«Технології розподілених систем та паралельних обчислень»

*на тему*

**«Мікросервіси»**

Виконав:  
 ст.гр. КН-303  
Сидоркін В.С.

Прийняв:

Скорохода О.В.

**Львів 2021**

**Мета роботи:** Оволодіти практичними прийомами розробки мікросервісів.

**Послідовність роботи:**

1. Розбийте систему, розроблену вами у лабораторній роботі No3 на окремі

мікросервіси.

2. Мікросервіси можна запускати локально на окремих портах, або використати

Docker контейнери.

3. Для демонстрації роботи програми використайте будь-який http клієнт,

наприклад, Postman

7. Оформити звіт про роботу за такою структурою:

• назва роботи;

• мета роботи;

• послідовність роботи;

• індивідуальне завдання;

• текст програми;

• скріншоти з http запитами та відповідями від сервісу;

• висновки.

**Індивідувальне завдання**

Розділити систему на мікросервіси, розроблену у 3 лабораторній роботі.

**Текст програми**

Приклад коду буде наведено для одного мікросервіса. Решта мікросервісів були реалізовані за аналогією.

***BillController:***

package com.example.bill\_service.controllers;

import com.example.bill\_service.models.Bill;

import com.example.bill\_service.services.BillService;

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

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

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

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

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

@RestController

@RequestMapping("/bills")

public class BillController {

@Autowired

private BillService billService;

@GetMapping("/{projectId}")

public Bill showBill(@PathVariable long projectId) {

return billService.showBill(projectId);

}

}

***Bill:***

package com.example.bill\_service.models;

import lombok.AllArgsConstructor;

import lombok.Data;

@Data

@AllArgsConstructor

public class Bill {

private long cost;

private Project project;

}

***Developer:***

package com.example.bill\_service.models;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Developer {

private long id;

private String firstName;

private String lastName;

private int costPerHour;

private int countOfHours;

}

***Project:***

package com.example.bill\_service.models;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

import java.util.List;

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Project {

private long id;

private int daysDuration;

private List<Developer> developerList;

private String description;

}

***BillService:***

package com.example.bill\_service.services;

import com.example.bill\_service.models.Bill;

import com.example.bill\_service.models.Project;

import lombok.RequiredArgsConstructor;

import org.springframework.stereotype.Service;

import org.springframework.web.client.RestTemplate;

@Service

@RequiredArgsConstructor

public class BillService {

private final RestTemplate restTemplate;

public Bill showBill(long projectId) {

Project project = restTemplate.getForObject("http://PROJECT-SERVICE/projects/" + projectId, Project.class);

long costOfProject = restTemplate.getForObject("http://PROJECT-SERVICE/projects/" + projectId + "/cost", long.class);

return new Bill(costOfProject, project);

}

}

***BillServiceApplication:***

package com.example.bill\_service;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.cloud.client.loadbalancer.LoadBalanced;

import org.springframework.cloud.netflix.eureka.EnableEurekaClient;

import org.springframework.context.annotation.Bean;

import org.springframework.web.client.RestTemplate;

@SpringBootApplication

@EnableEurekaClient

public class BillServiceApplication {

public static void main(String[] args) {

SpringApplication.run(BillServiceApplication.class, args);

}

@Bean

@LoadBalanced

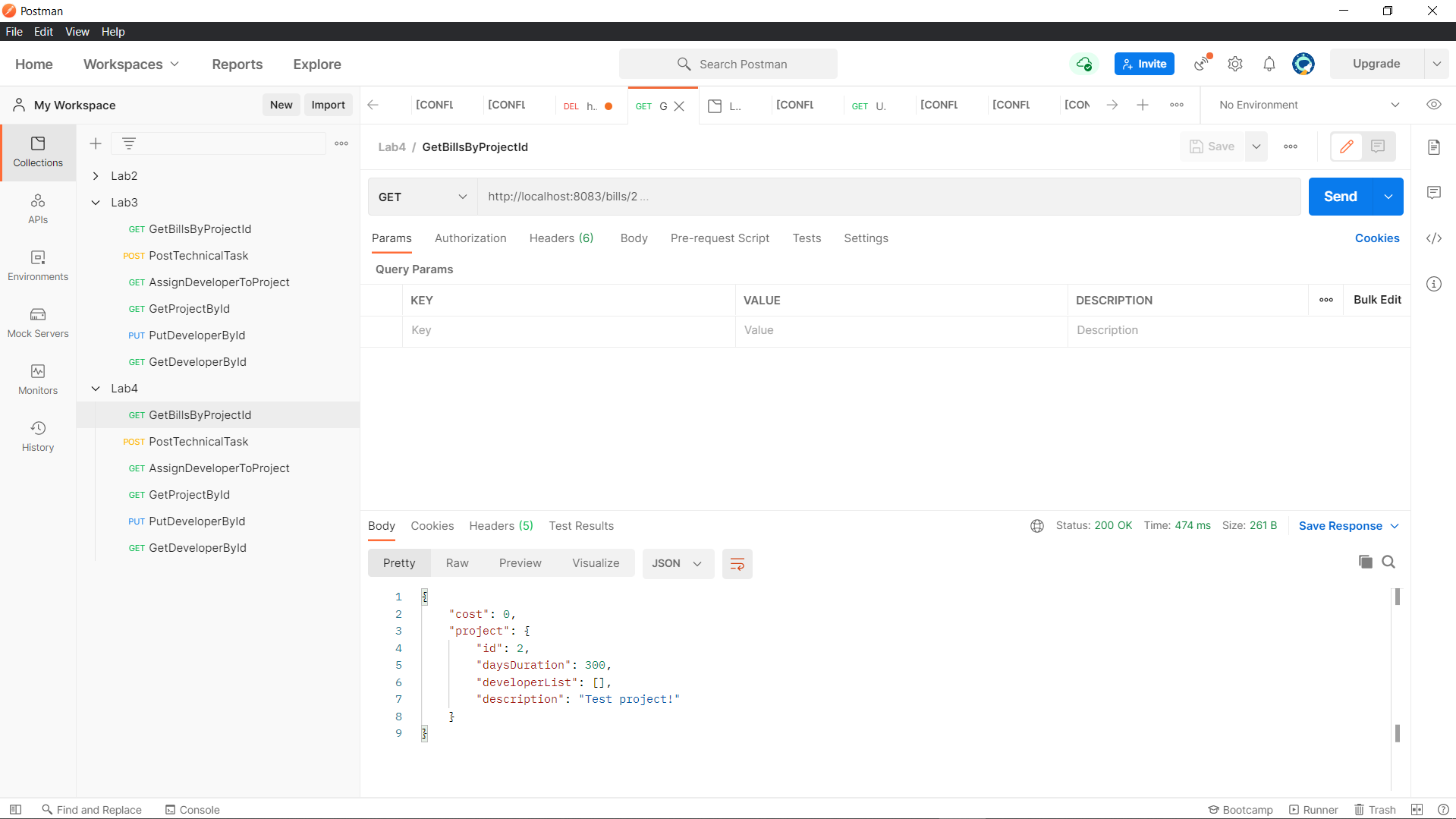
public RestTemplate restTemplate() {

return new RestTemplate();

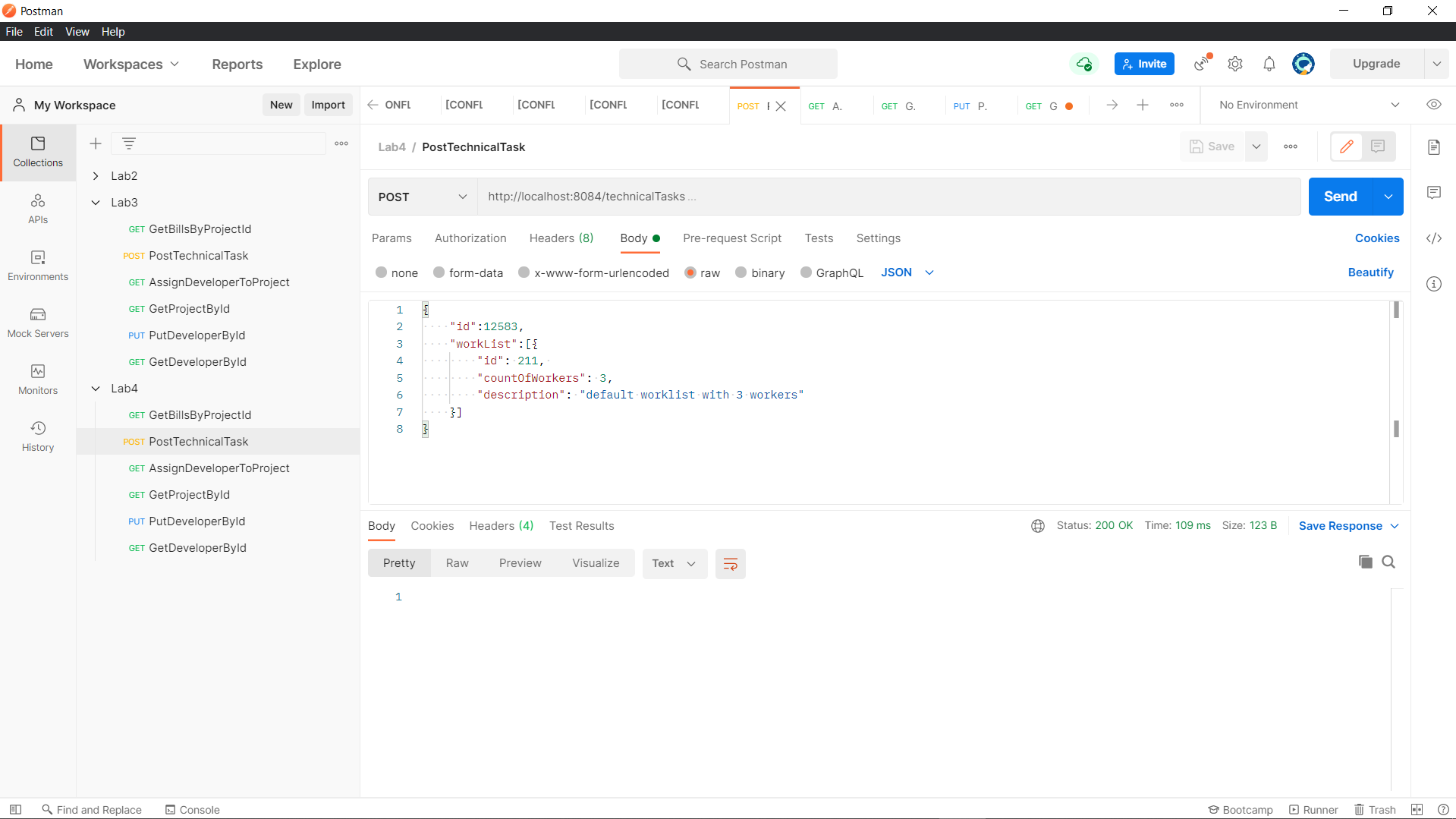
}

}

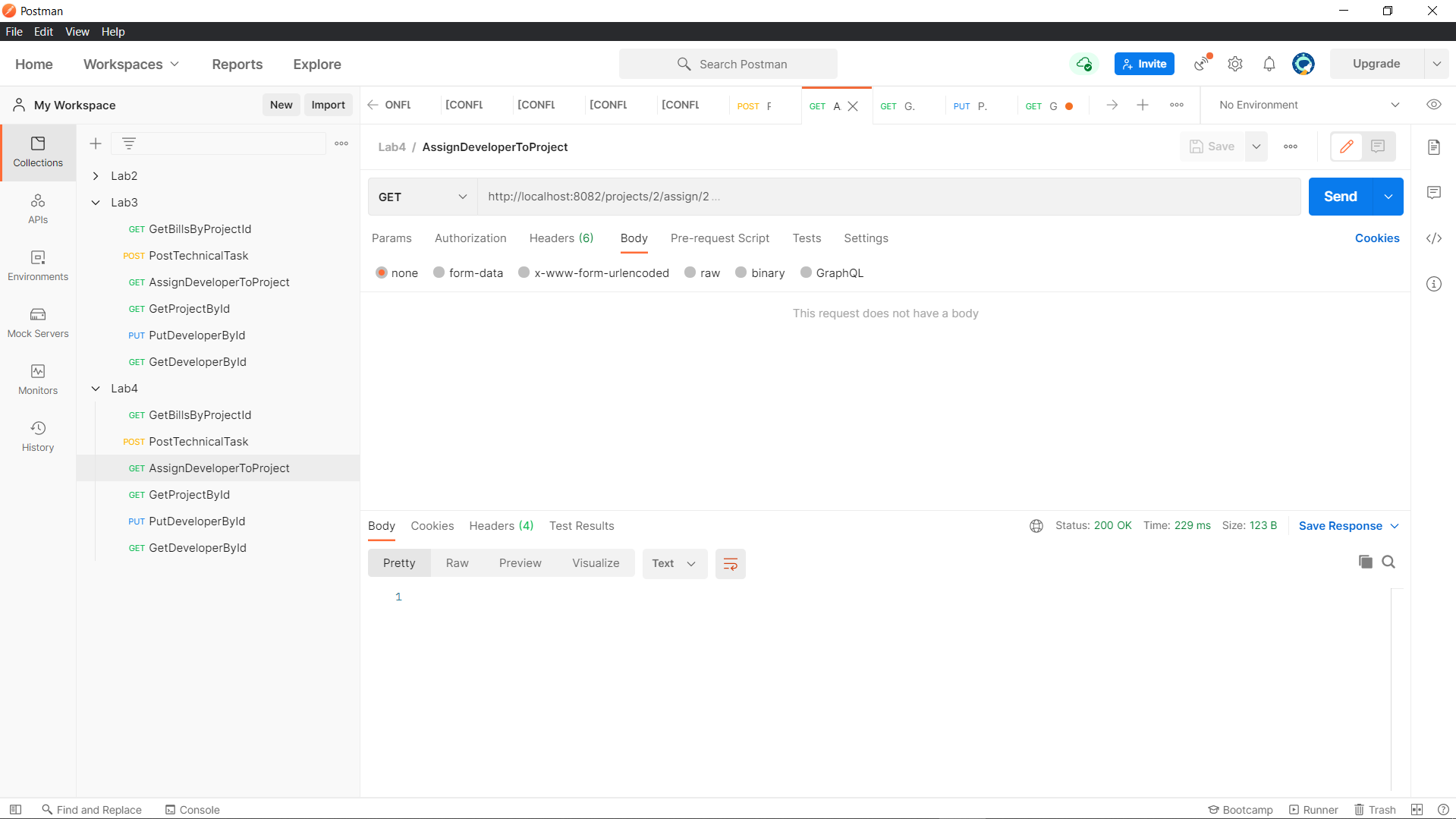
**Результат виконання програми**

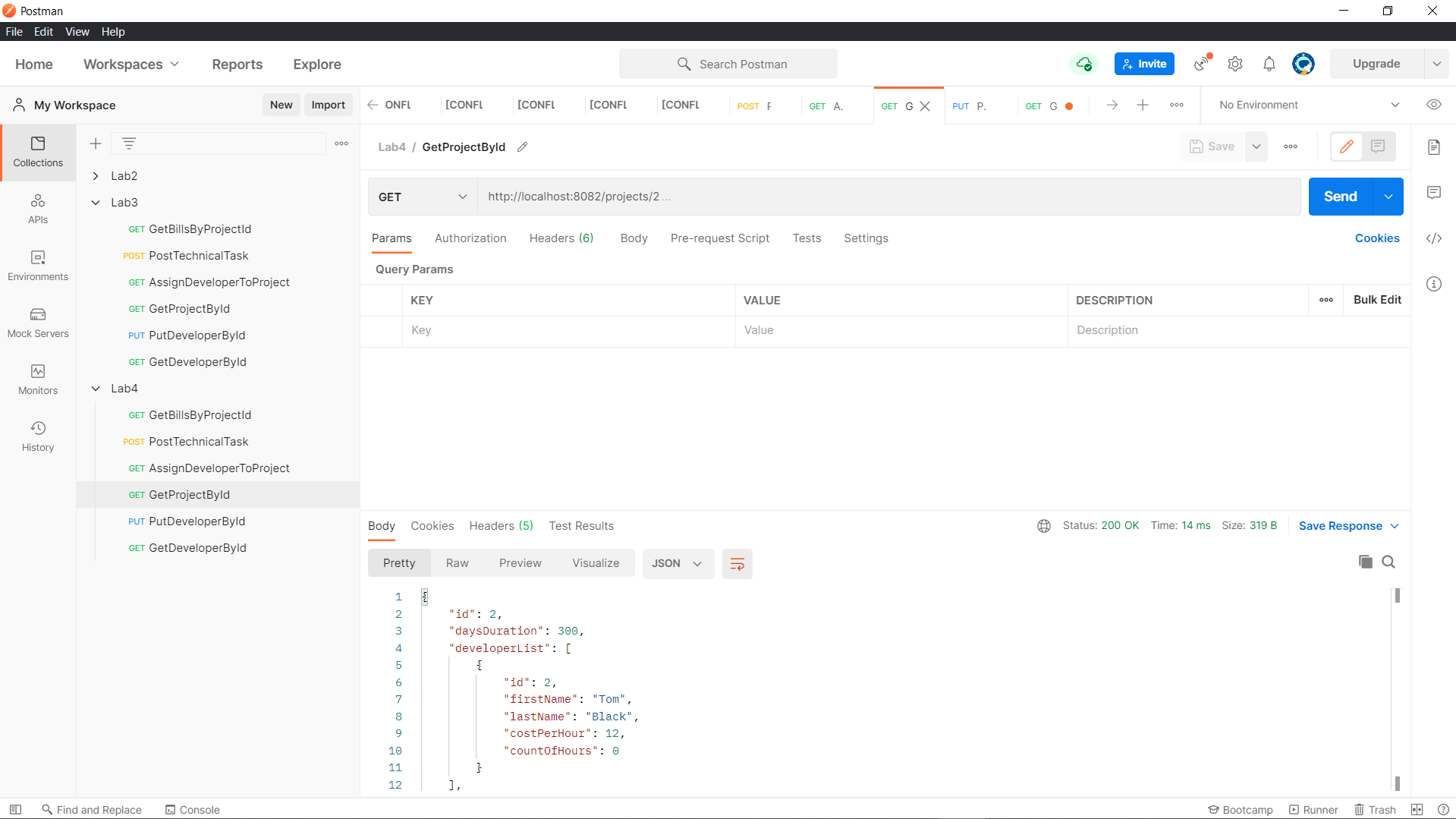


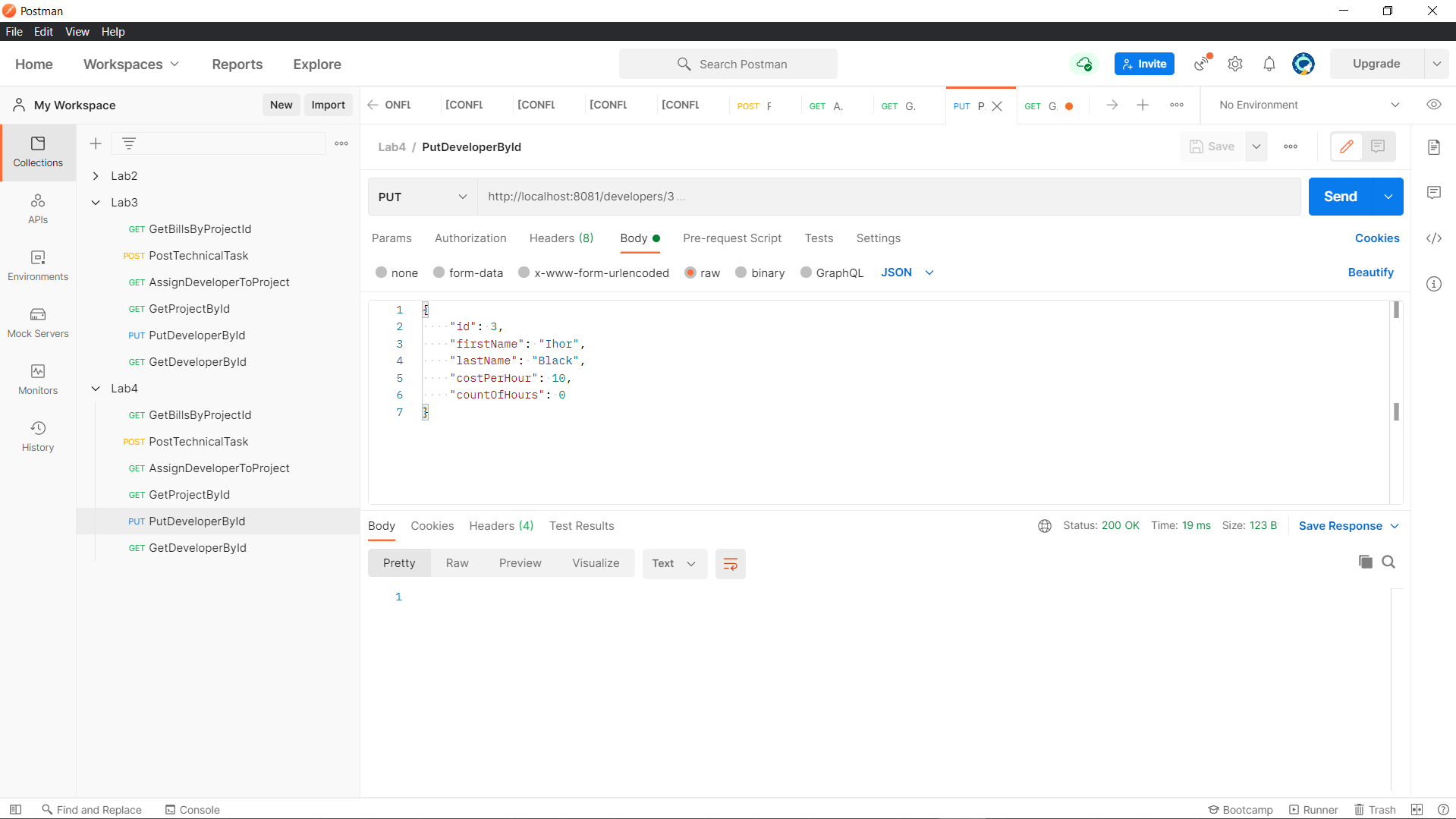
*Рис. 1. Запит на отримання конкретного рахунку*



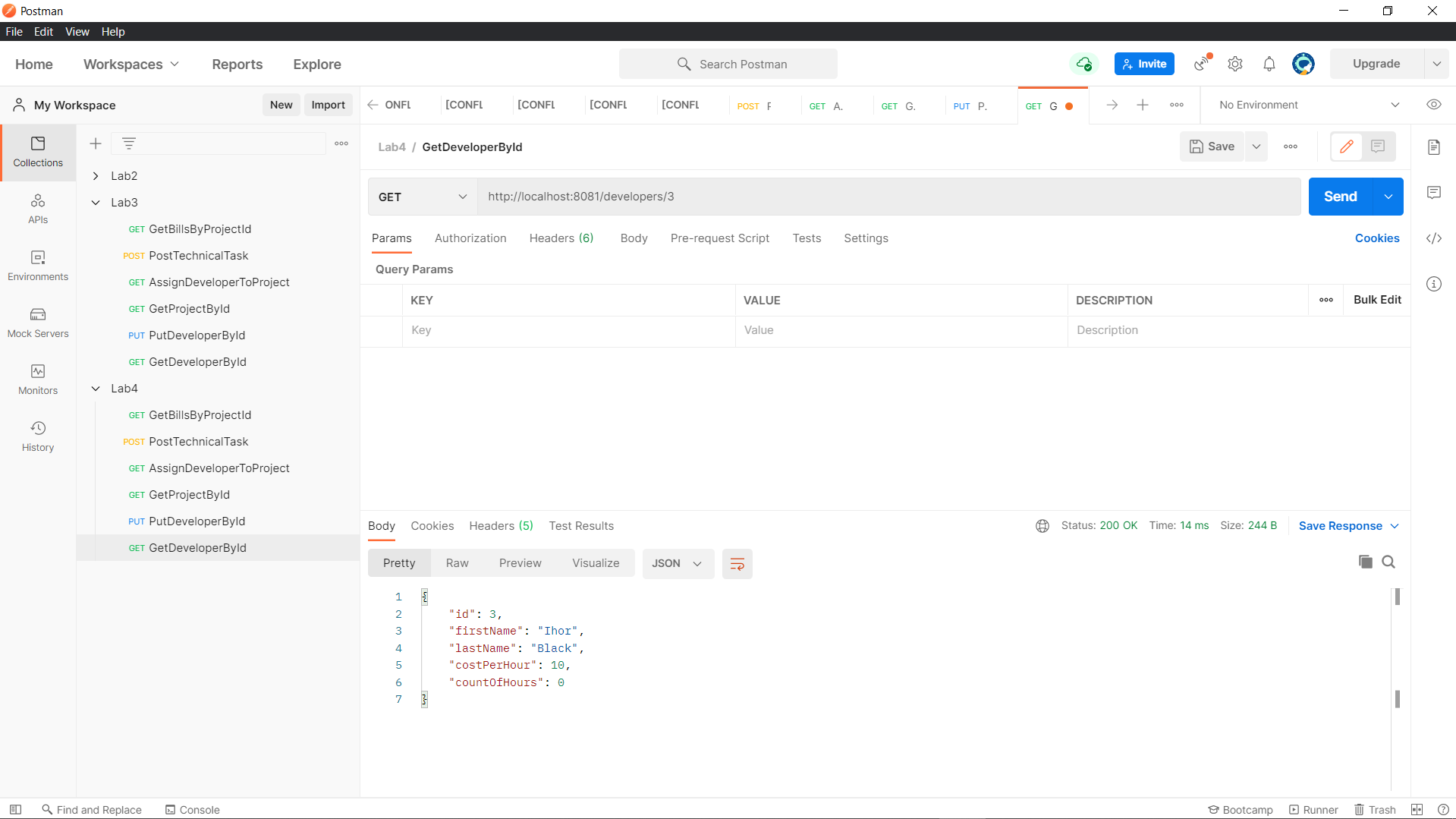
*Рис.2 Створення нового технічного завдання*

 *Рис.3 Результат додавання розробника до проекту*

*Рис.4 Запит на отримання конкретного проекту*



*Рис.5 Запит на оновлення даних розробника*



*Рис.6 Запит на отримання конкретного ррозробника*

**Висновки**

На даній лаблраторній роботі було розроблено програму за завданням відповідного ідивідуального варіанту. У програмі було реалізовано мікросервіси для системи, реалізованої у 3 лабораторній роботі. Було реалізовано наздсилання запитів, для перевірки роботи сервісів. Результати роботи програми було наведено у звіті.