Міністерство освіти і науки

Національний університет «Львівська політехніка»

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

**Звіт**

до лабораторної роботи №3

*з дисципліни*

«Технології розподілених систем та паралельних обчислень»

*на тему*

**«RESTful веб сервіси (частина 2)»**

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

Прийняв:

Скорохода О.В.

**Львів 2021**

**Мета роботи:** Оволодіти практичними прийомами розробки RESTful веб сервісів.

**Послідовність роботи:**

1. Напишіть REST-сервіс для обраної задачі, який буде опрацьовувати запити різних

видів (Get, Post, Put, Delete).

2. Можна використовувати будь-яку мову програмування / фреймворк.

3. Дані повинні зберігатися хоча б на час роботи вашого сервісу. Відповідно можна

використати як базу даних, так і просто зберігати інформацію в оперативній

пам’яті.

4. 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**

21.Система Команда розробників. **Замовник** надає **Технічне** **завдання** **(ТЗ)**, у якому перераховано перелік **Робіт** із вказанням кваліфікації та потрібної кількості спеціалістів. **Менеджер** розглядає **ТЗ** и оформляє Проект, призначаючи на нього незайнятих **Розробників** потрібної кваліфікації, після чого розраховується вартість **Проекту** і Замовнику виставляється **Рахунок**. **Розробник** має можливість відзначити кількість годин, які він витратив на роботу над проектом.

**Текст програми**

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

***BillController:***

package com.example.lab3.controllers;

import com.example.lab3.models.Bill;

import com.example.lab3.services.BillService;

import lombok.AllArgsConstructor;

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")

@AllArgsConstructor

public class BillController {

@Autowired

private BillService billService;

@GetMapping("/{projectId}")

public Bill showBill(@PathVariable long projectId) {

return billService.showBill(projectId);

}

}

***DeveloperController:***

package com.example.lab3.controllers;

import com.example.lab3.models.Developer;

import com.example.lab3.services.DeveloperService;

import lombok.AllArgsConstructor;

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

import java.util.List;

@RestController

@RequestMapping("/developers")

@AllArgsConstructor

public class DeveloperController {

private DeveloperService developerService;

@GetMapping("/all")

public List<Developer> getAllDevelopers() {

return developerService.getAllDevelopers();

}

@PostMapping

public void saveDeveloper(@RequestBody Developer developer) {

developerService.saveDeveloper(developer);

}

@GetMapping("/{id}")

public Developer findById(@PathVariable long id) {

return developerService.findById(id);

}

@DeleteMapping("/{id}")

public void deleteDeveloper(@PathVariable long id) {

developerService.deleteDeveloper(id);

}

@PutMapping("/{id}")

public void updateDeveloper(@PathVariable long id, @RequestBody Developer newDeveloper) {

developerService.updateDeveloper(id, newDeveloper);

}

@GetMapping("/{id}/logWork")

public void logWork(@PathVariable long id, @RequestParam int hours) {

developerService.logWork(id, hours);

}

}

***PtojectController:***

package com.example.lab3.controllers;

import com.example.lab3.models.Project;

import com.example.lab3.services.ProjectService;

import lombok.AllArgsConstructor;

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

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

import java.util.List;

@RestController

@RequestMapping("/projects")

@AllArgsConstructor

public class ProjectController {

@Autowired

private ProjectService projectService;

@GetMapping("/all")

public List<Project> getAllProjects() {

return projectService.getAllProjects();

}

@PostMapping

public void createProject(@RequestBody Project project) {

projectService.createProject(project);

}

@GetMapping("/{id}")

public Project findById(@PathVariable long id) {

return projectService.findById(id);

}

@GetMapping("/{projectId}/assign/{developerId}")

public void assignDeveloperToProject(@PathVariable long projectId, @PathVariable long developerId) {

projectService.assignDeveloperToProject(projectId, developerId);

}

@GetMapping("/{id}/cost")

public long calculateCostOfProject(@PathVariable long id) {

return projectService.calculateCostOfProject(id);

}

}

***TechnicalTaskController:***

package com.example.lab3.controllers;

import com.example.lab3.models.TechnicalTask;

import com.example.lab3.services.TechnicalTaskService;

import lombok.AllArgsConstructor;

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

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

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

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

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

@RestController

@RequestMapping("/technicalTasks")

@AllArgsConstructor

public class TechnicalTaskController {

@Autowired

private TechnicalTaskService technicalTaskService;

@PostMapping

public void createTechnicalTask(@RequestBody TechnicalTask technicalTask) {

technicalTaskService.createTechnicalTask(technicalTask);

}

}

***BillService:***

package com.example.lab3.services;

import com.example.lab3.models.Bill;

import com.example.lab3.models.Project;

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

import org.springframework.stereotype.Service;

@Service

public class BillService {

@Autowired

private ProjectService projectService;

public Bill showBill(long projectId) {

Project project = projectService.findById(projectId);

long costOfProject = projectService.calculateCostOfProject(projectId);

return new Bill(costOfProject, project);

}

}

***DeveloperService:***

package com.example.lab3.services;

import com.example.lab3.models.Developer;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class DeveloperService {

private static List<Developer> developerList = new ArrayList<>();

static {

developerList.add(new Developer(1L, "Karl", "Brown", 10, 0));

developerList.add(new Developer(2L, "Tom", "Black", 12, 0));

developerList.add(new Developer(3L, "Ihor", "White", 8, 0));

}

public List<Developer> getAllDevelopers() {

return developerList;

}

public void saveDeveloper(Developer developer) {

developerList.add(developer);

}

public Developer findById(long id) {

return developerList.stream()

.filter(dev -> dev.getId() == id)

.findFirst()

.orElseThrow(() -> new RuntimeException("Developer not found!"));

}

public void deleteDeveloper(long id) {

Developer developer = findById(id);

developerList.remove(developer);

}

public void updateDeveloper(long id, Developer newDeveloper) {

Developer oldDeveloper = findById(id);

oldDeveloper.setFirstName(newDeveloper.getFirstName());

oldDeveloper.setLastName(newDeveloper.getLastName());

oldDeveloper.setCostPerHour(newDeveloper.getCostPerHour());

oldDeveloper.setCountOfHours(newDeveloper.getCountOfHours());

}

public void logWork(long developerId, int hours) {

Developer developer = findById(developerId);

developer.setCountOfHours(developer.getCountOfHours() + hours);

}

}

***ProjectService:***

package com.example.lab3.services;

import com.example.lab3.models.Developer;

import com.example.lab3.models.Project;

import lombok.AllArgsConstructor;

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

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

@AllArgsConstructor

public class ProjectService {

private static List<Project> projectList = new ArrayList<>();

static {

projectList.add(new Project(1L, 365, new ArrayList<>(), "Fun project!"));

projectList.add(new Project(2L, 300, new ArrayList<>(), "Test project!"));

projectList.add(new Project(3L, 365, new ArrayList<>(), "Fake project!"));

}

@Autowired

private DeveloperService developerService;

public List<Project> getAllProjects() {

return projectList;

}

public void createProject(Project project) {

this.projectList.add(project);

}

public void assignDeveloperToProject(long projectId, long developerId) {

Project project = findById(projectId);

Developer developer = developerService.findById(developerId);

project.getDeveloperList().add(developer);

}

public Project findById(long id) {

return projectList.stream()

.filter(p -> p.getId() == id)

.findFirst()

.orElseThrow(() -> new RuntimeException("Project not found"));

}

public long calculateCostOfProject(long projectId) {

Project project = findById(projectId);

List<Developer> developerList = project.getDeveloperList();

int daysDuration = project.getDaysDuration();

long cost = 0;

for (Developer developer : developerList) {

cost += developer.getCostPerHour() \* 8 \* daysDuration;

}

return cost;

}

}

***TechnicalTaskService:***

package com.example.lab3.services;

import com.example.lab3.models.TechnicalTask;

import org.springframework.stereotype.Service;

import java.util.ArrayList;

import java.util.List;

@Service

public class TechnicalTaskService {

private List<TechnicalTask> technicalTaskList = new ArrayList<>();

public void createTechnicalTask(TechnicalTask technicalTask){

this.technicalTaskList.add(technicalTask);

}

}

***Bill:***

package com.example.lab3.models;

import lombok.AllArgsConstructor;

import lombok.Data;

@Data

@AllArgsConstructor

public class Bill {

private long cost;

private Project project;

}

***Developer:***

package com.example.lab3.models;

import lombok.AllArgsConstructor;

import lombok.Data;

@Data

@AllArgsConstructor

public class Developer {

private long id;

private String firstName;

private String lastName;

private int costPerHour;

private int countOfHours;

}

***Project:***

package com.example.lab3.models;

import lombok.AllArgsConstructor;

import lombok.Data;

import java.util.ArrayList;

import java.util.List;

@Data

@AllArgsConstructor

public class Project {

private long id;

private int daysDuration;

private List<Developer> developerList;

private String description;

}

***TechnicalTask:***

public void logWork(long developerId, int hours) {

package com.example.lab3.models;

import lombok.Data;

import java.util.ArrayList;

import java.util.List;

@Data

public class TechnicalTask{

private long id;

private List<Work> workList = new ArrayList<>();

}

***Work:***

package com.example.lab3.models;

import lombok.Data;

@Data

public class Work {

private long id;

private int countOfWorkers;

private String description;

}

***Lab3Application:***

package com.example.lab3;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Lab3Application {

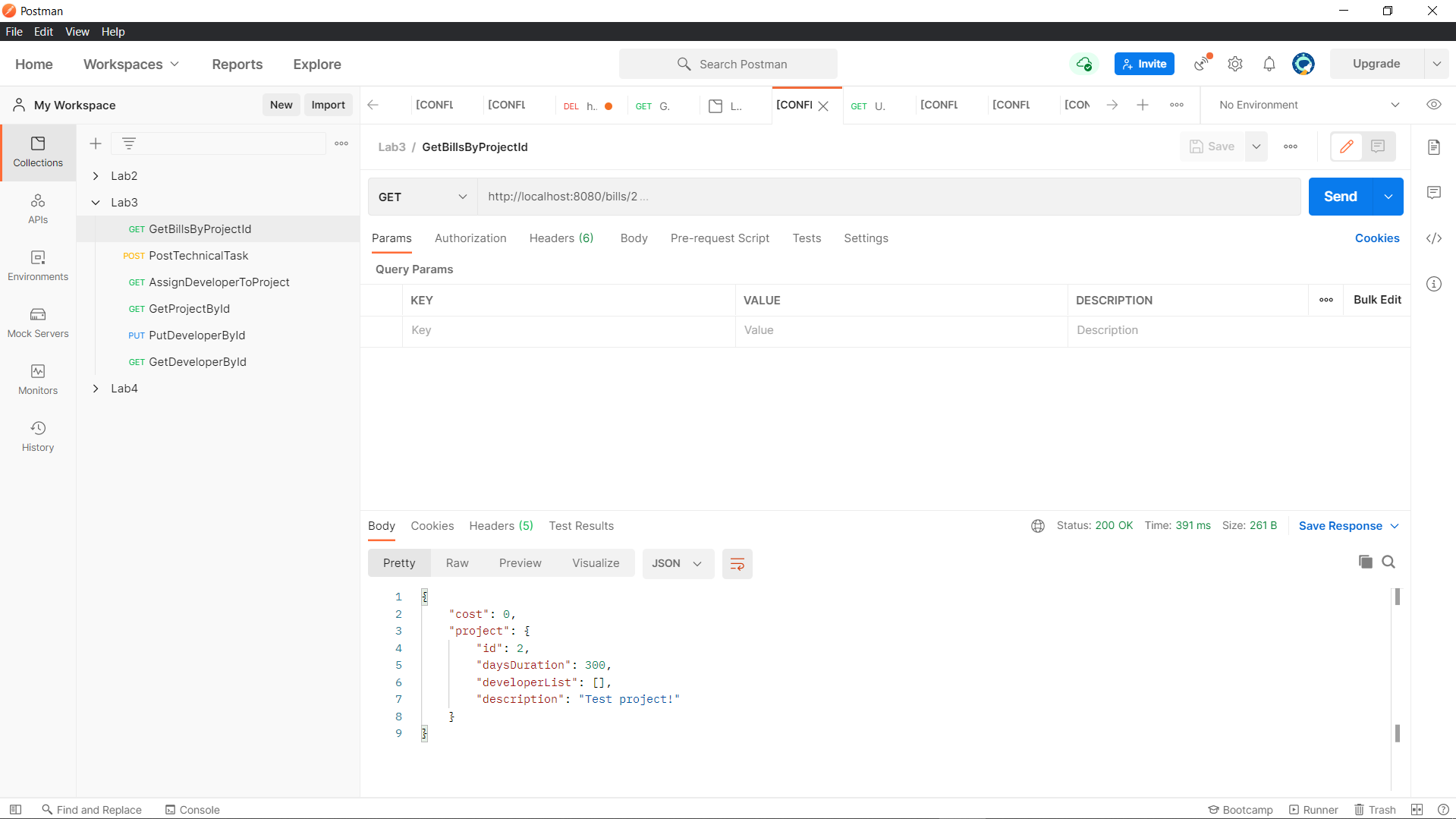
public static void main(String[] args) {

SpringApplication.run(Lab3Application.class, args);

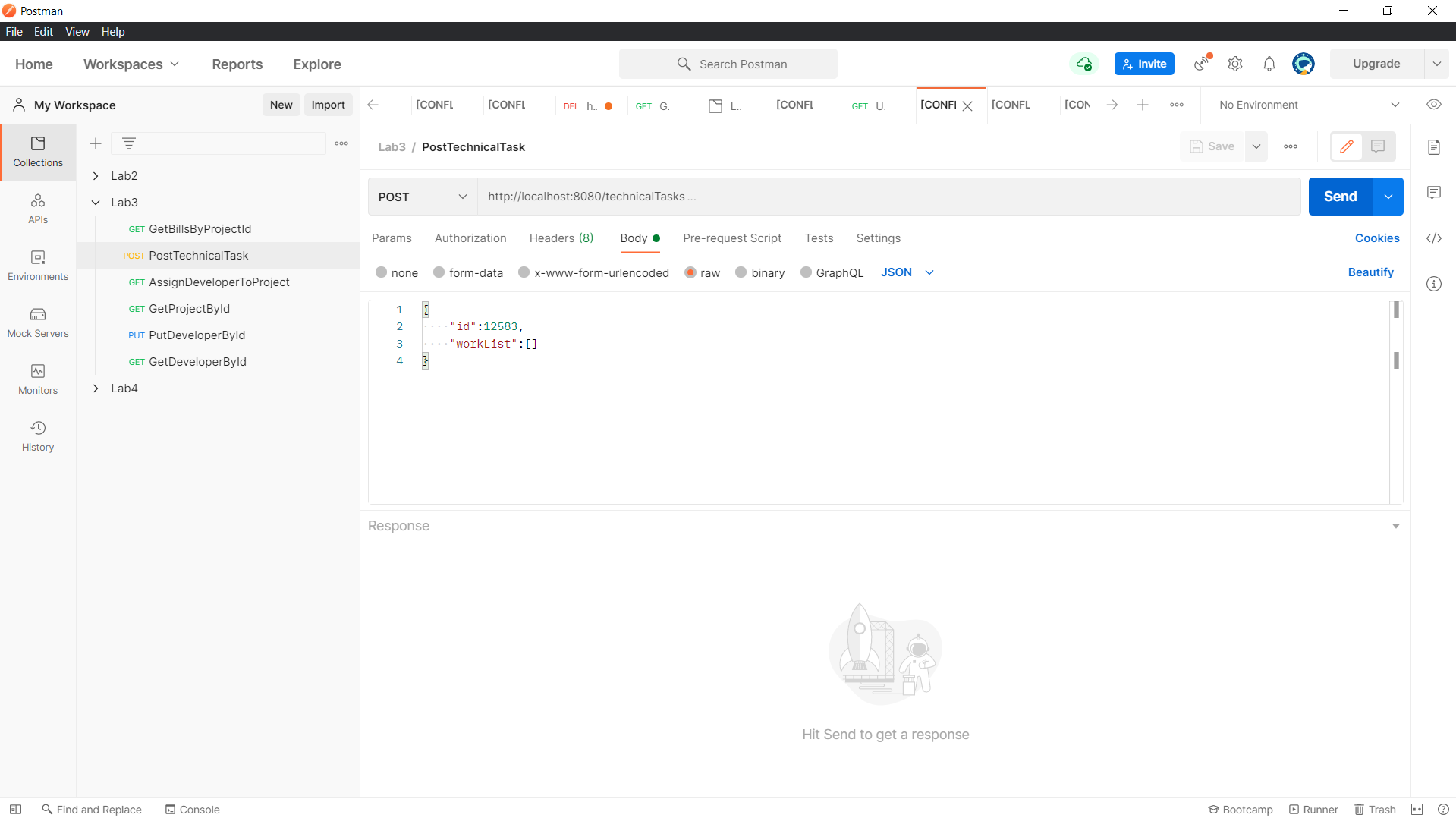
}

}

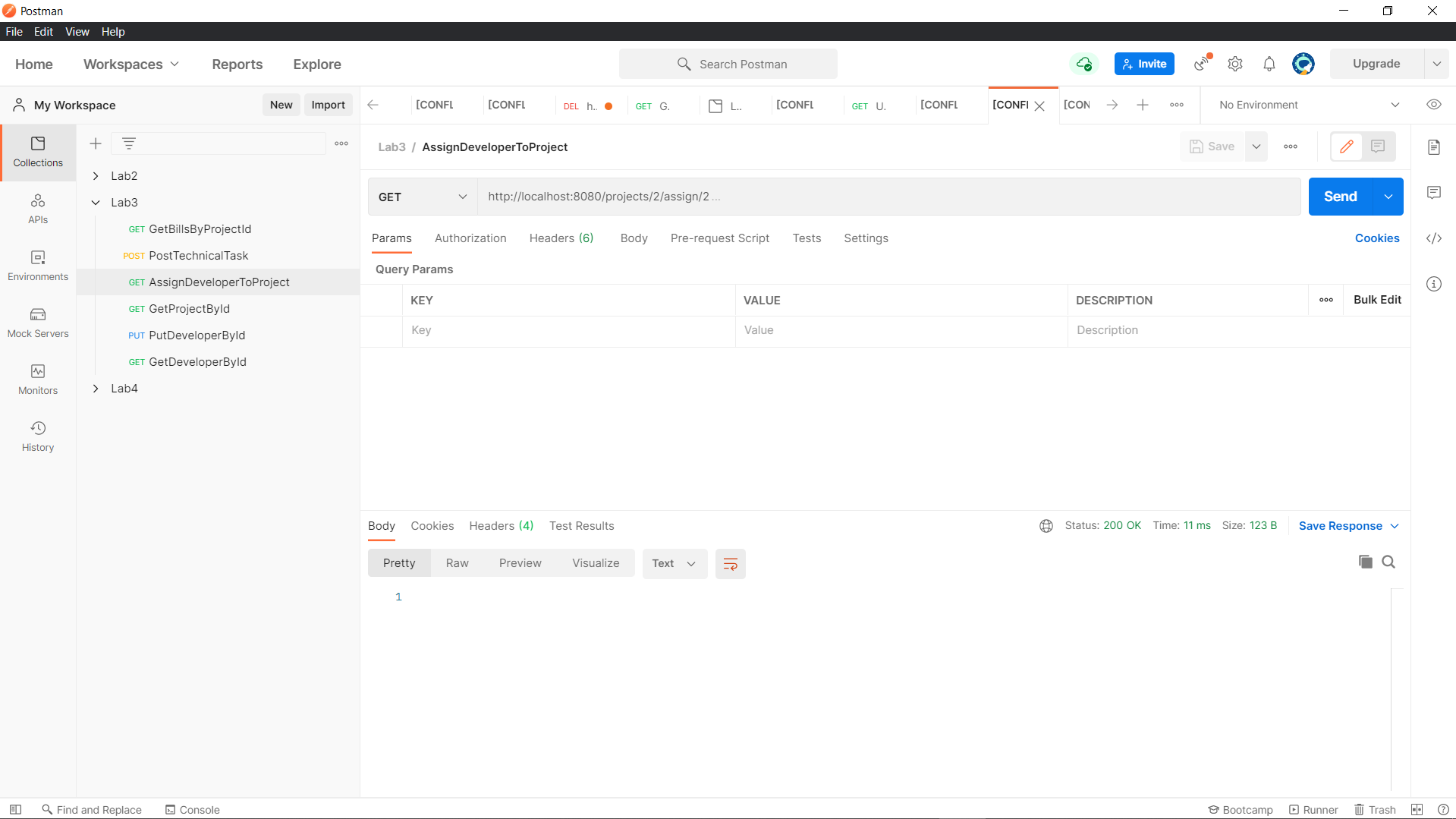
**Результат виконання програми**

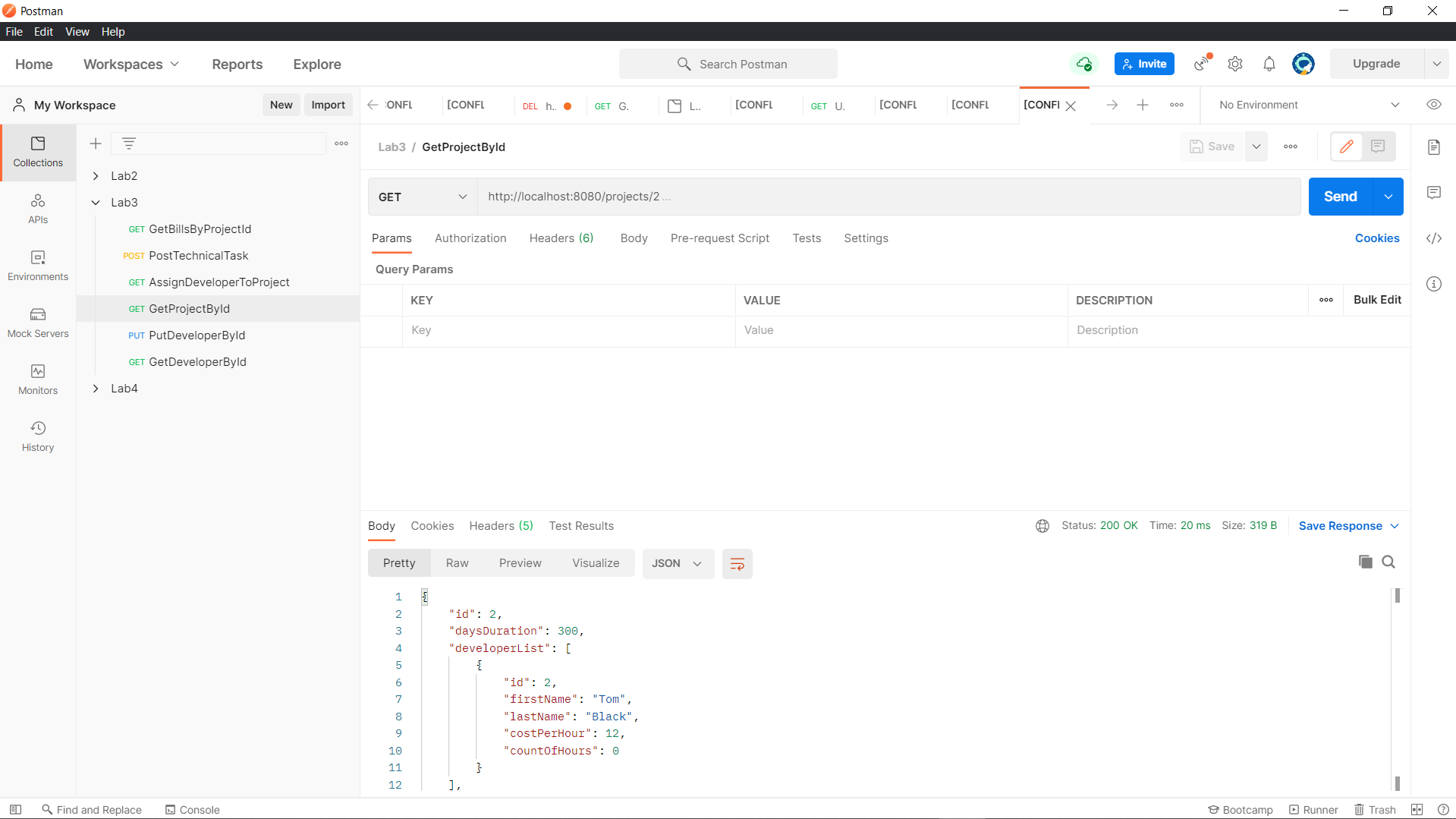


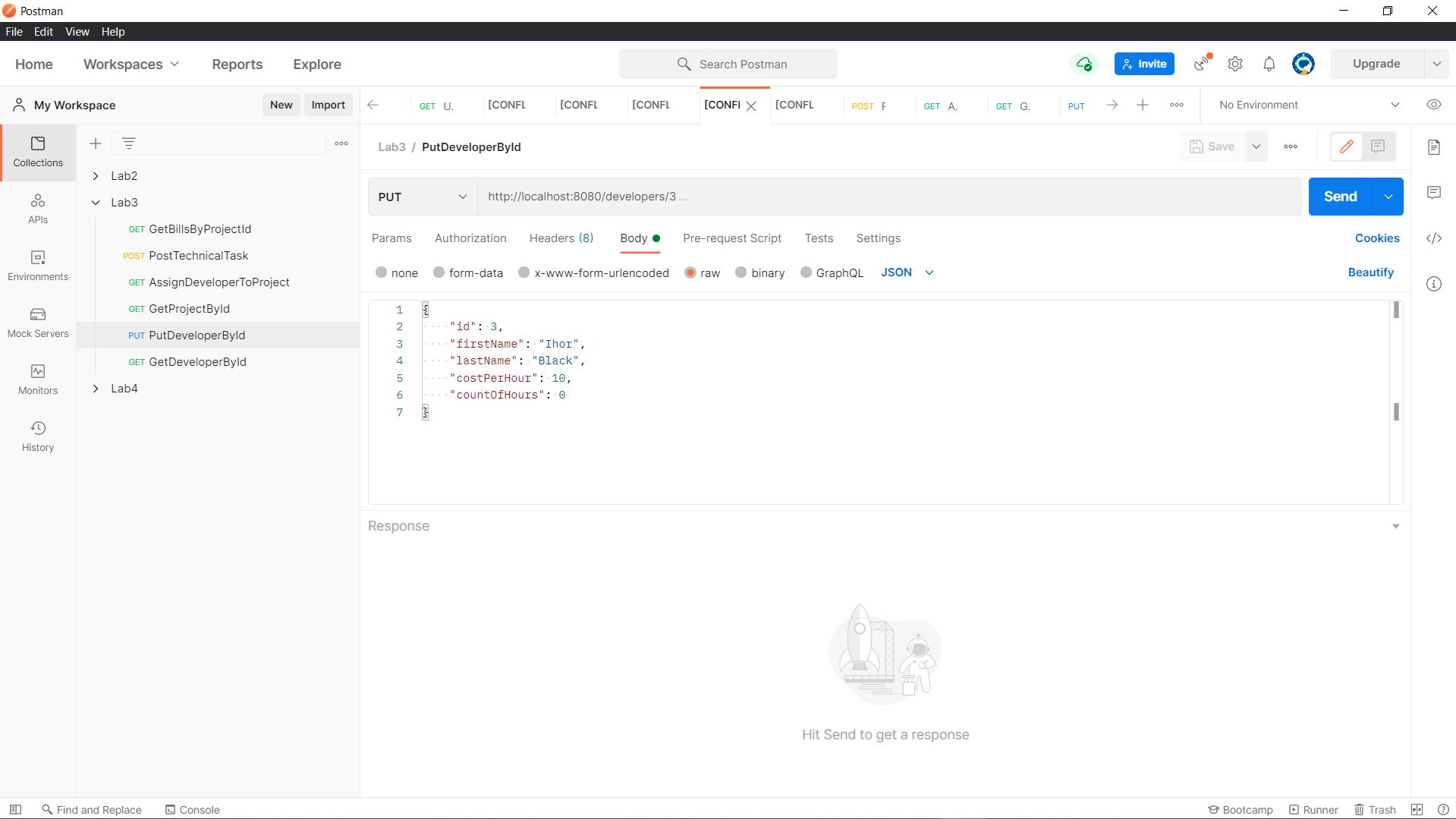
*Рис. 1. Запит на отримання конкретного рахунку*



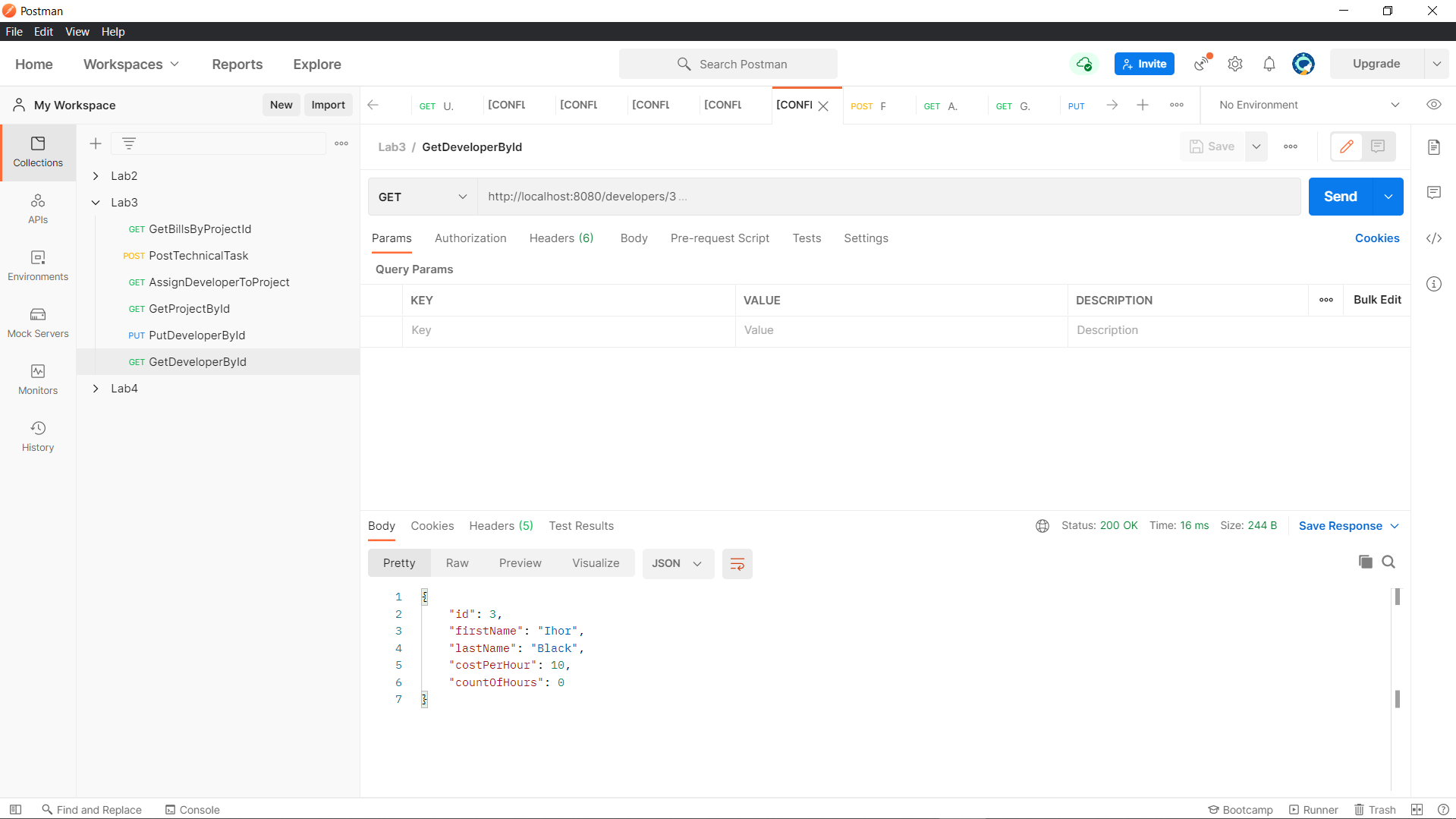
*Рис.2 Створення нового технічного завдання*

 *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 RestFul сервіс для системи, заданої у індивідуальному варіанті. Було реалізовано наздсилання запитів, для перевірки роботи сервісів. Результати роботи програми було наведено у звіті.