

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Институт Информатики и кибернетики   
Кафедра Программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
  
к лабораторной работе №6 по дисциплине «Автоматизация тестирования»

Обучающийся группы 6231-020302D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гижевская В.Д.

Обучающийся группы 6231-020302D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алкеев М.Г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лобанков А.А.

Самара 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc146727791)

[2 Результаты работы 4](#_Toc146727792)

[ПРИЛОЖЕНИЕ А Листинг программы](#_Toc146727796) 7

1. Постановка задачи

В процессе выполнения заданий познакомиться с возможностью автоматизированного тестирования API в Postman.

1. Создать несколько environments (например local и dev).Прописать для них переменные, необходимые для формирования url (хост, порт, протокол).
2. Создать коллекцию, в которую добавить реквесты для:
   1. енд-поинта получения результата сложения
   2. енд-поинта получения результата умножения
   3. енд-поинта получения результата вычитания
   4. енд-поинта получения результата деления
   5. енд-поинта получения данных о всех вычислениях за определенный момент времени
3. Создать pre-request script, для коллекции, который на основе выбранного environment будет формировать url. Во всех реквестах заменить захардкоженную строку подключения, на переменную, которая будет формироваться в pre-request скрипте
4. Для каждого реквеста необходимо добавить несколько тестов (как положительных так и негативных). Проверить как сервер справляется с нагрузкой при последовательной отправке 100 итераций вызовов конечных точек.
5. Результаты работы

На рисунках 1-8 представлены результаты работы.

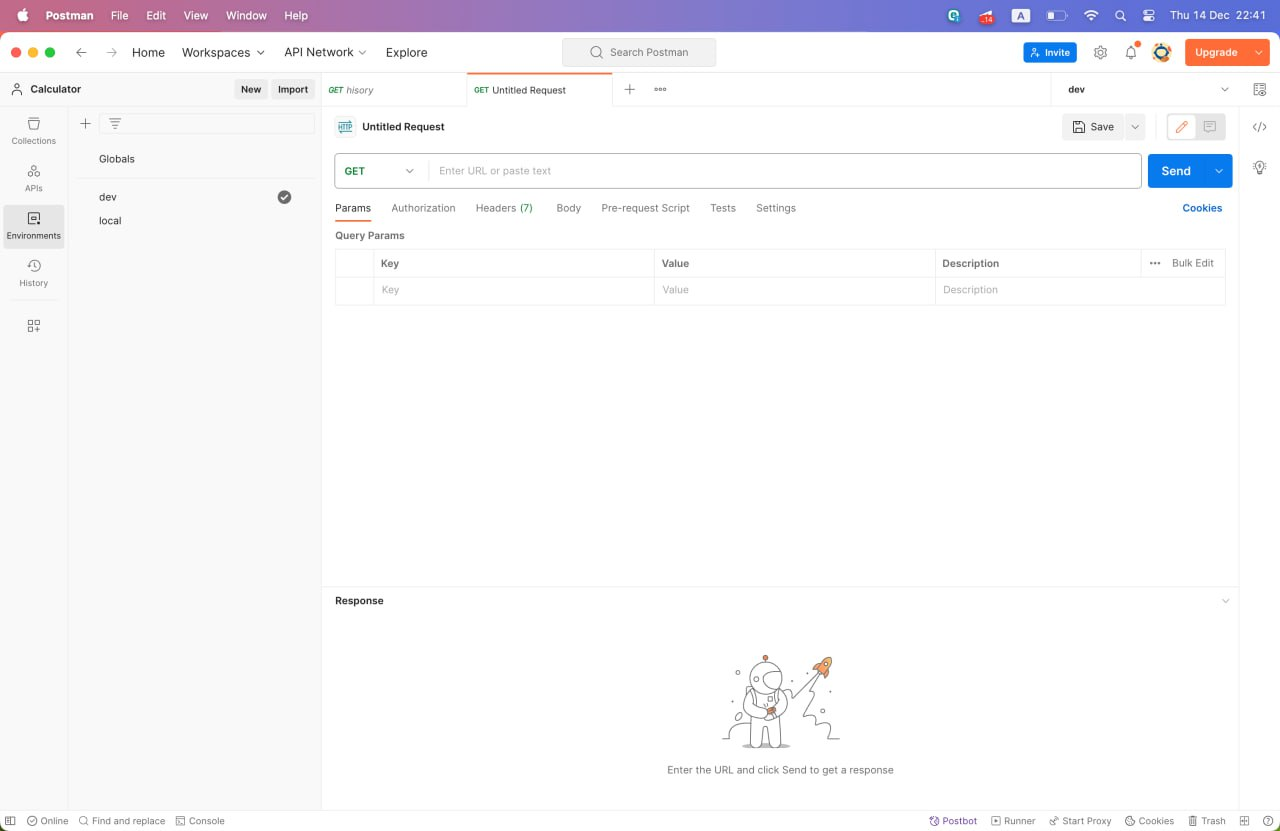


Рисунок 1 – environments

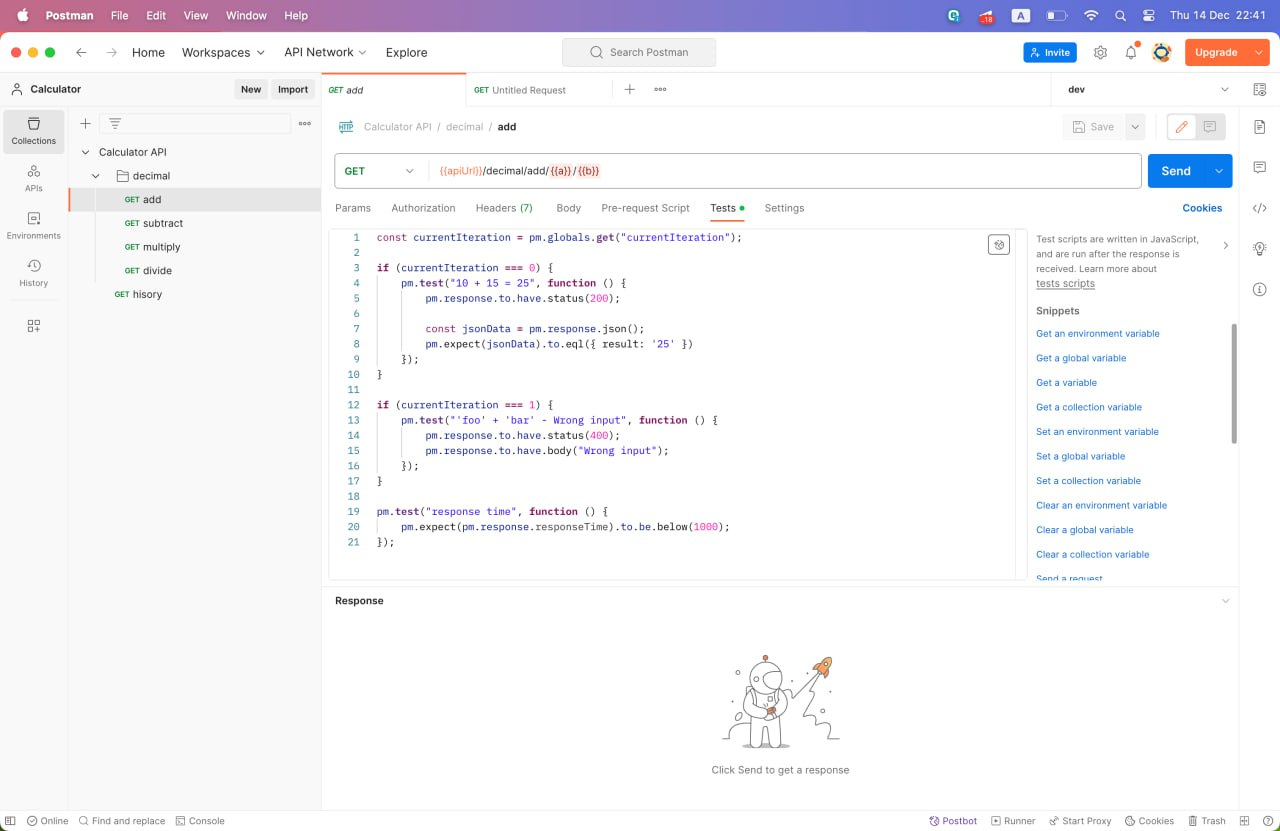


Рисунок 2 – decimal add реквест

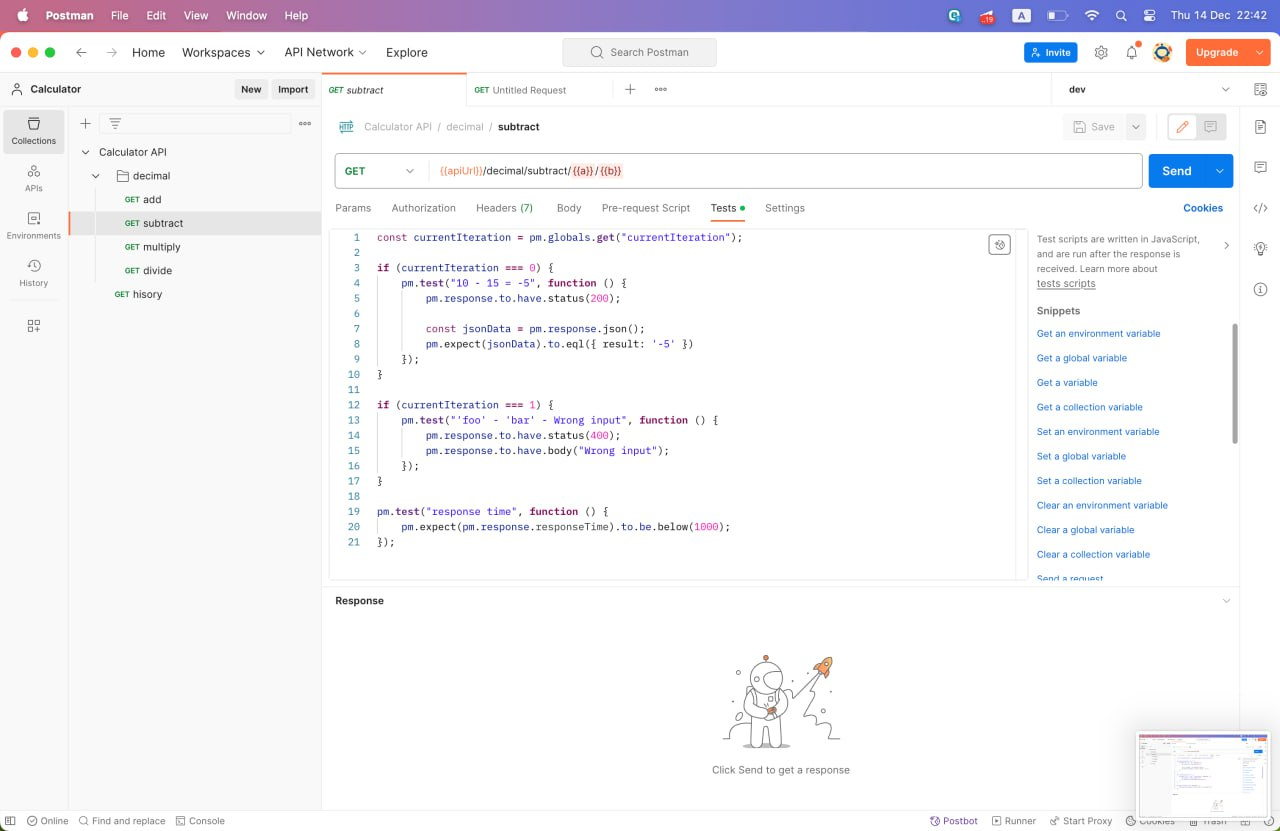


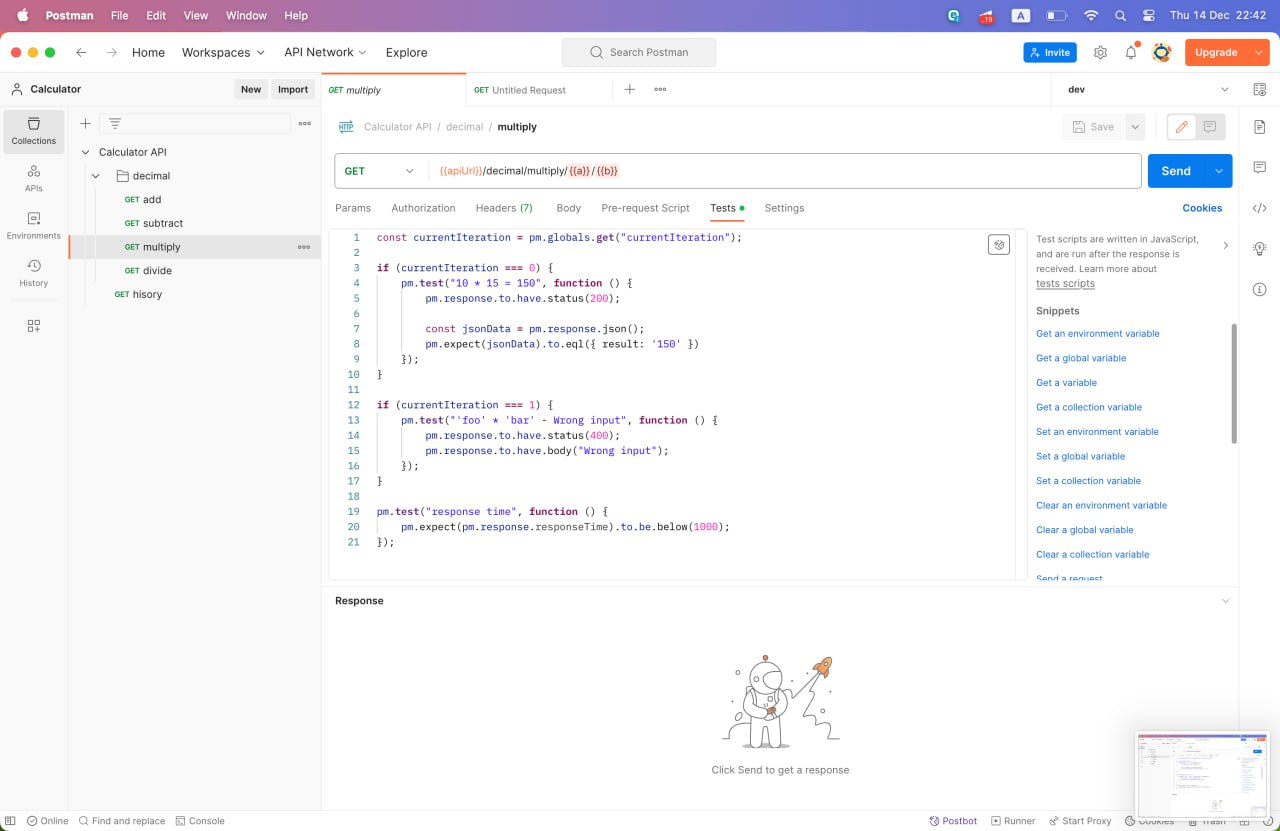
Рисунок 3 – decimal subtract реквест

Рисунок 4 – decimal multiply реквест

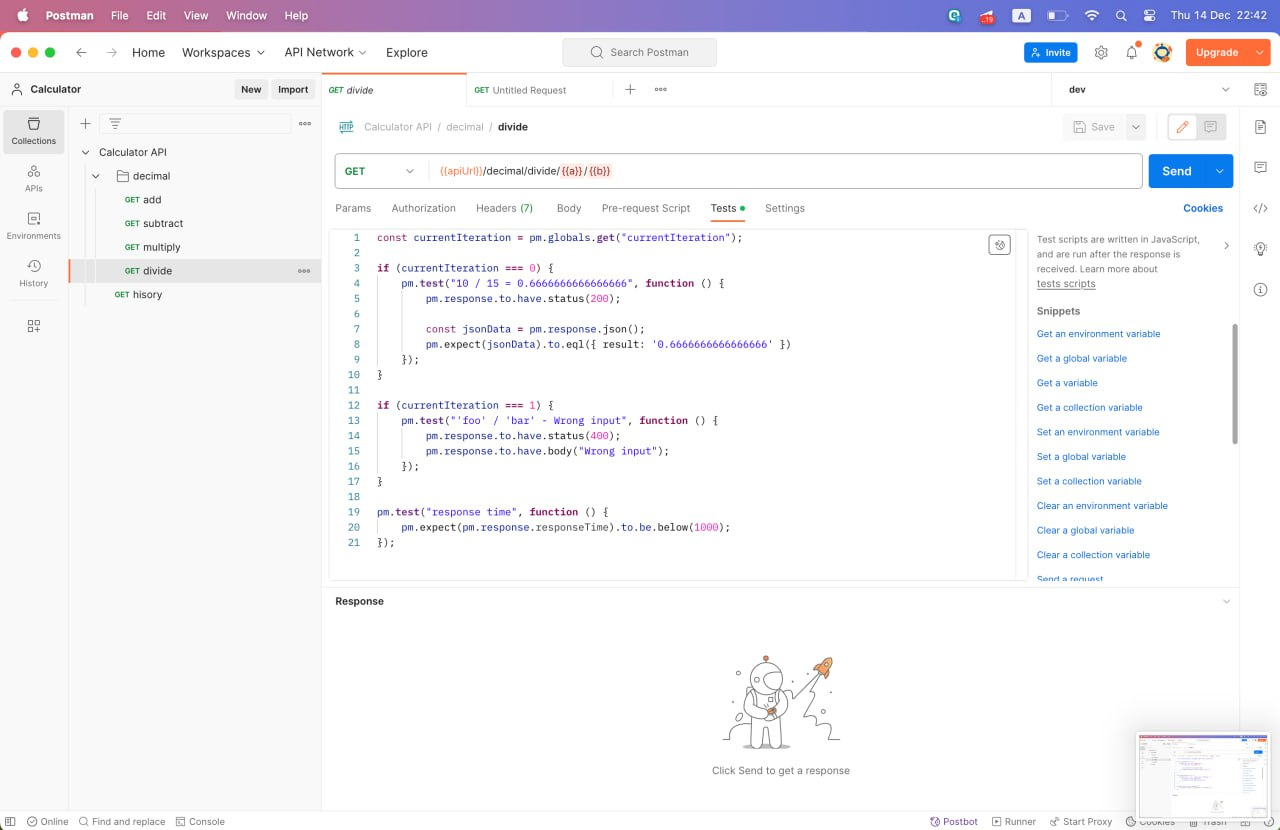


Рисунок 5 – decimal divide реквест

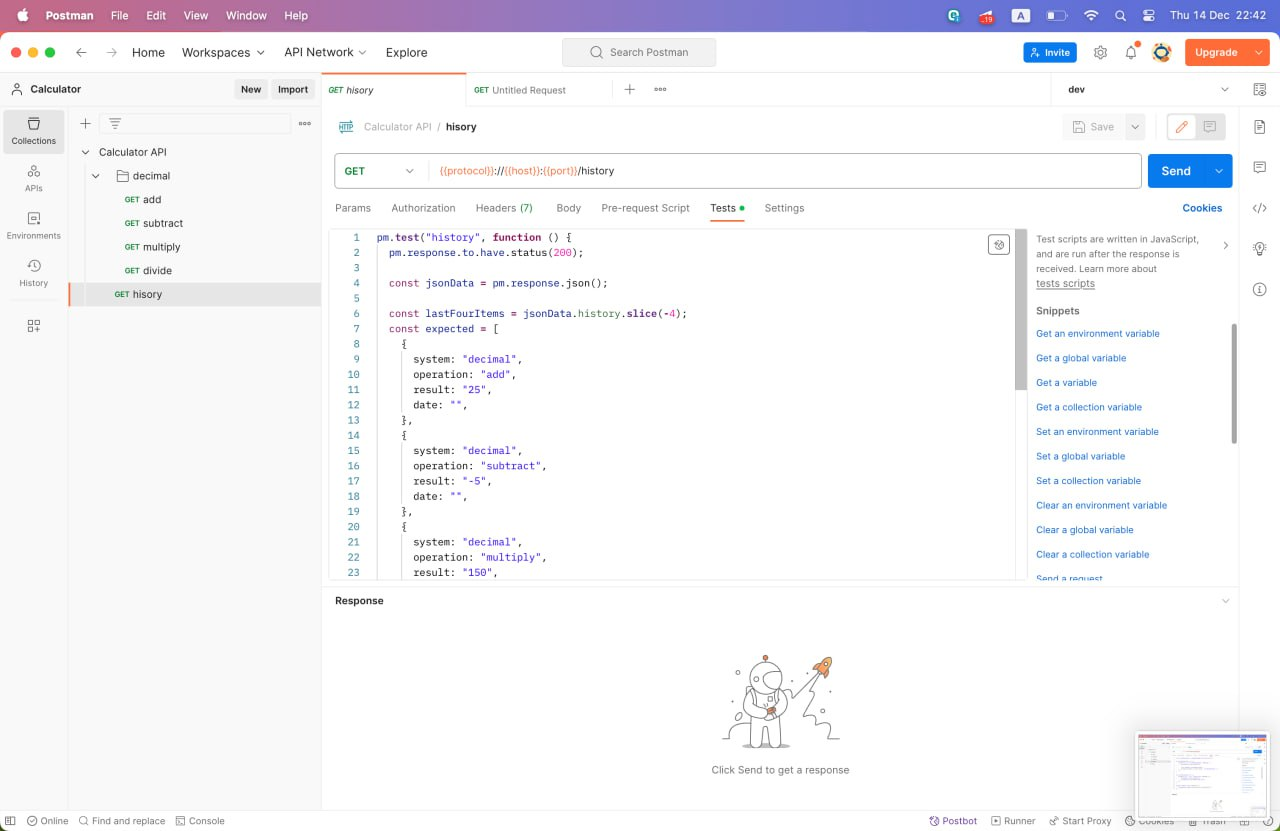


Рисунок 6 –history реквест

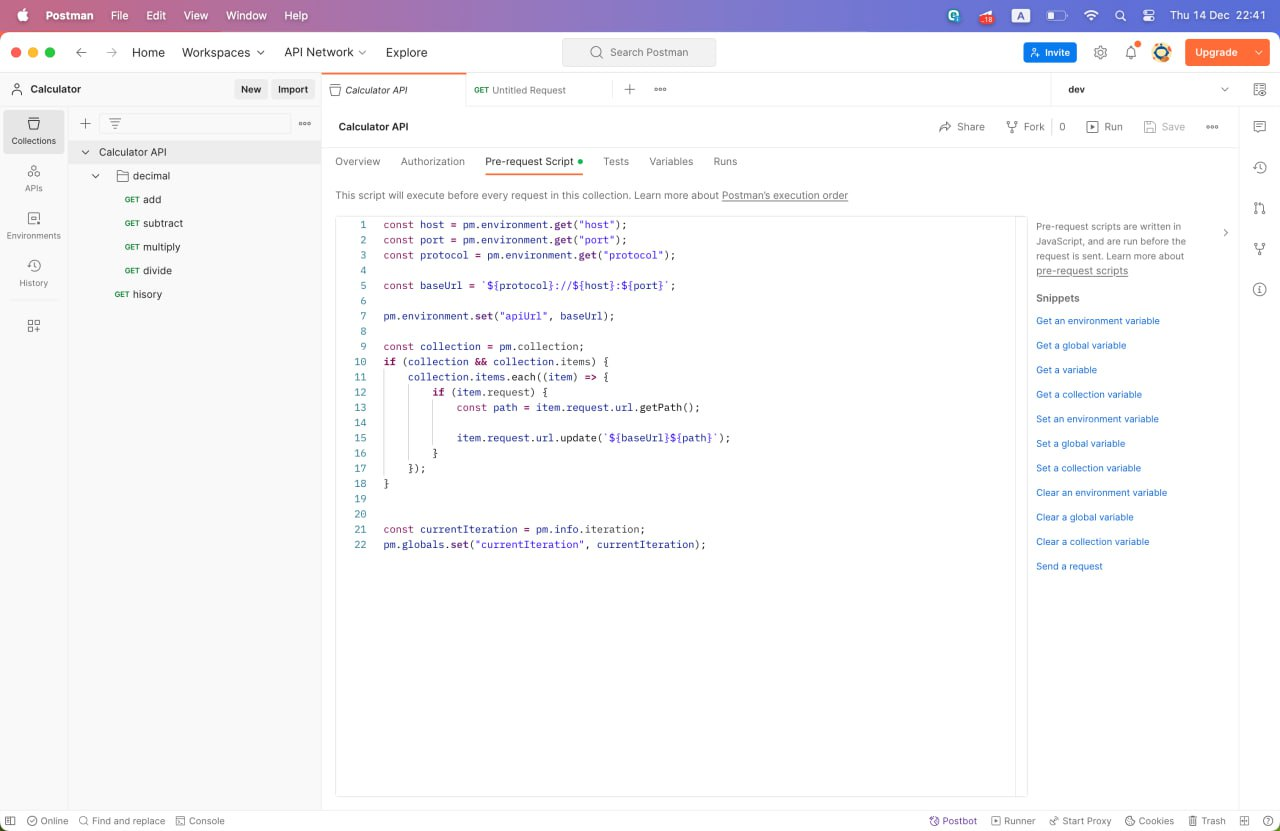


Рисунок 7 – pre-request script для коллекции

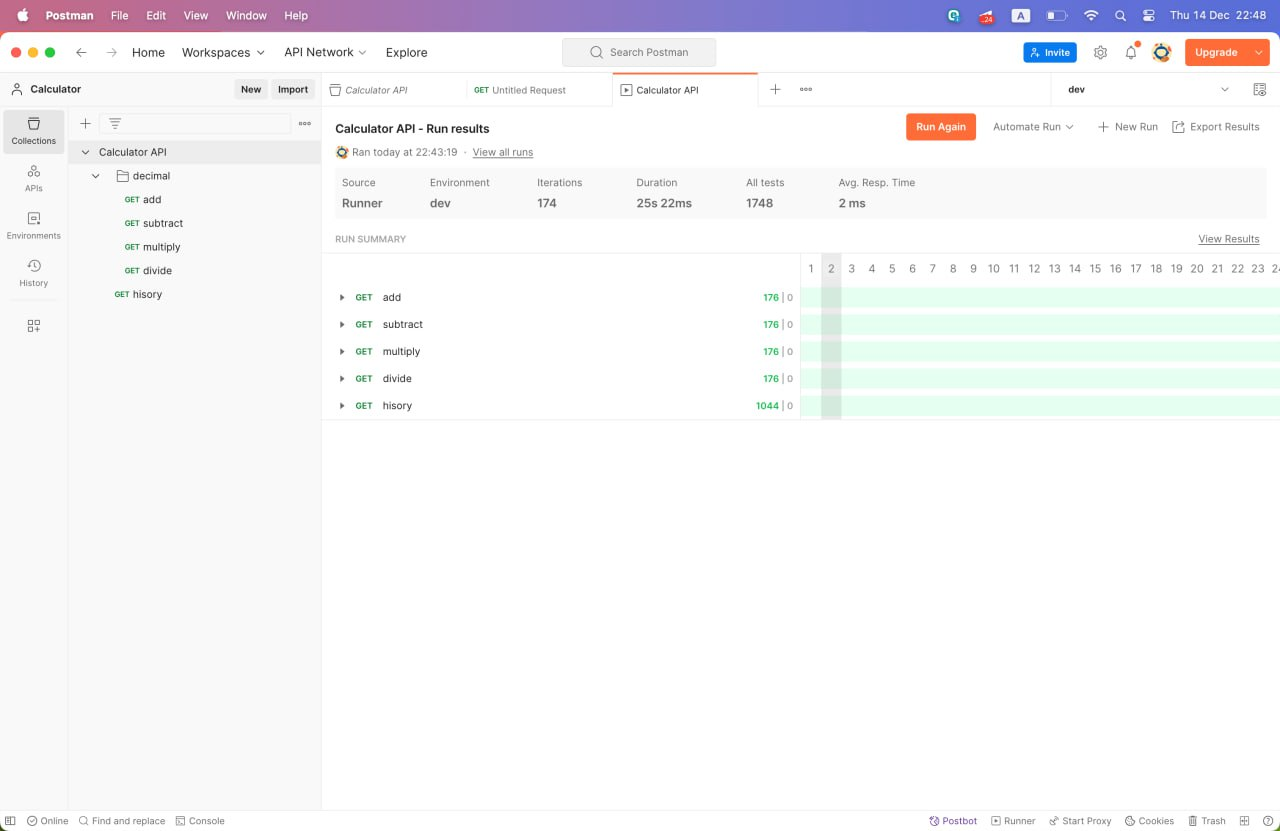


Рисунок 8 – результат последовательной отправки вызовов

ПРИЛОЖЕНИЕ А  
Листинг программы

const host = pm.environment.get("host");

const port = pm.environment.get("port");

const protocol = pm.environment.get("protocol");

const baseUrl = ${protocol}://${host}:${port};

pm.environment.set("apiUrl", baseUrl);

const collection = pm.collection;

if (collection && collection.items) {

collection.items.each((item) => {

if (item.request) {

const path = item.request.url.getPath();

item.request.url.update(${baseUrl}${path});

}

});

}

const currentIteration = pm.info.iteration;

pm.globals.set("currentIteration", currentIteration);

const currentIteration = pm.globals.get("currentIteration");

if (currentIteration === 0) {

pm.test("10 + 15 = 25", function () {

pm.response.to.have.status(200);

const jsonData = pm.response.json();

pm.expect(jsonData).to.eql({ result: '25' })

});

}

if (currentIteration === 1) {

pm.test("'foo' + 'bar' - Wrong input", function () {

pm.response.to.have.status(400);

pm.response.to.have.body("Wrong input");

});

}

pm.test("response time", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});

const currentIteration = pm.globals.get("currentIteration");

if (currentIteration === 0) {

pm.test("10 - 15 = -5", function () {

pm.response.to.have.status(200);

const jsonData = pm.response.json();

pm.expect(jsonData).to.eql({ result: '-5' })

});

}

if (currentIteration === 1) {

pm.test("'foo' - 'bar' - Wrong input", function () {

pm.response.to.have.status(400);

pm.response.to.have.body("Wrong input");

});

}

pm.test("response time", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});

const currentIteration = pm.globals.get("currentIteration");

if (currentIteration === 0) {

pm.test("10 \* 15 = 150", function () {

pm.response.to.have.status(200);

const jsonData = pm.response.json();

pm.expect(jsonData).to.eql({ result: '150' })

});

}

if (currentIteration === 1) {

pm.test("'foo' \* 'bar' - Wrong input", function () {

pm.response.to.have.status(400);

pm.response.to.have.body("Wrong input");

});

}

pm.test("response time", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});

const currentIteration = pm.globals.get("currentIteration");

if (currentIteration === 0) {

pm.test("10 / 15 = 0.6666666666666666", function () {

pm.response.to.have.status(200);

const jsonData = pm.response.json();

pm.expect(jsonData).to.eql({ result: '0.6666666666666666' })

});

}

if (currentIteration === 1) {

pm.test("'foo' / 'bar' - Wrong input", function () {

pm.response.to.have.status(400);

pm.response.to.have.body("Wrong input");

});

}

pm.test("response time", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});

pm.test("history", function () {

pm.response.to.have.status(200);

const jsonData = pm.response.json();

const lastFourItems = jsonData.history.slice(-4);

const expected = [

{

system: "decimal",

operation: "add",

result: "25",

date: "",

},

{

system: "decimal",

operation: "subtract",

result: "-5",

date: "",

},

{

system: "decimal",

operation: "multiply",

result: "150",

date: "",

},

{

system: "decimal",

operation: "divide",

result: "0.6666666666666666",

date: "",

},

];

lastFourItems.forEach((item, index) => {

pm.test(verify history item with date ${item.date}, function () {

pm.expect(lastFourItems[index].system).to.equal(expected[index].system);

pm.expect(lastFourItems[index].operation).to.equal(

expected[index].operation

);

pm.expect(lastFourItems[index].result).to.be.equal(

expected[index].result

);

pm.expect(lastFourItems[index].date).to.be.a("string");

});

});

});

pm.test("response time", function () {

pm.expect(pm.response.responseTime).to.be.below(1000);

});