

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 6232-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Александров А.А.

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

Самара 2023

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc153562390)

[2 Результаты работы 4](#_Toc153562391)

[2.1 Результаты задания 1 4](#_Toc153562392)

[2.2 Результаты задания 2 5](#_Toc153562393)

[2.3 Результаты задания 3 7](#_Toc153562394)

[2.4 Результаты задания 4 8](#_Toc153562395)

[3 Вывод 11](#_Toc153562396)

[ПРИЛОЖЕНИЕ А Листинг программы 12](#_Toc153562397)

1. Постановка задачи

**Цель и постановка задания**

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

**Задание 1**

Создать несколько environments (например local и dev). Прописать для них переменные, необходимые для формирования url (хост, порт, протокол).

**Задание 2**

Создать коллекцию, в которую добавить реквесты для:

* енд-поинта получения результата сложения;
* енд-поинта получения результата умножения;
* енд-поинта получения результата вычитания;
* енд-поинта получения результата деления;
* енд-поинта получения данных обо всех вычислениях за определенный момент времени.

**Задание 3**

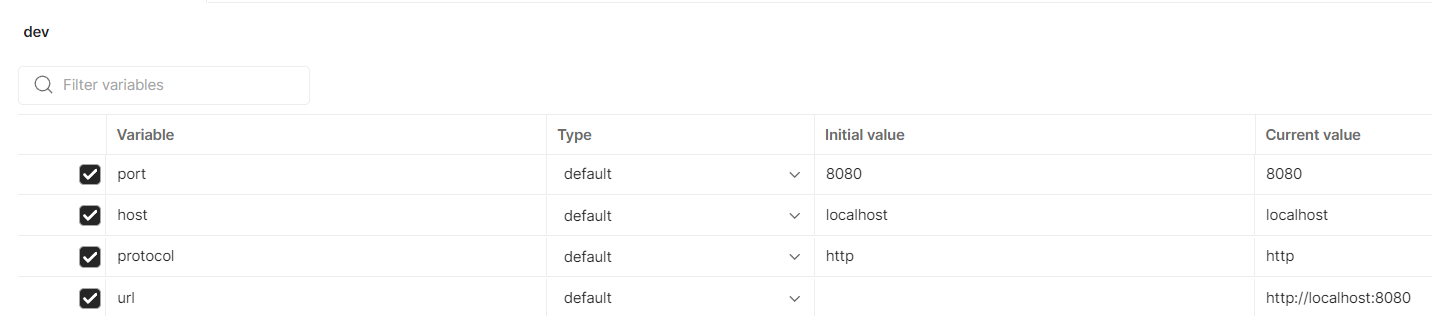
Создать pre-request script, для коллекции, который на основе выбранного environment будет формировать url. Во всех реквестах заменить захардкоженную строку подключения, на переменную, которая будет формироваться в pre-request скрипте.

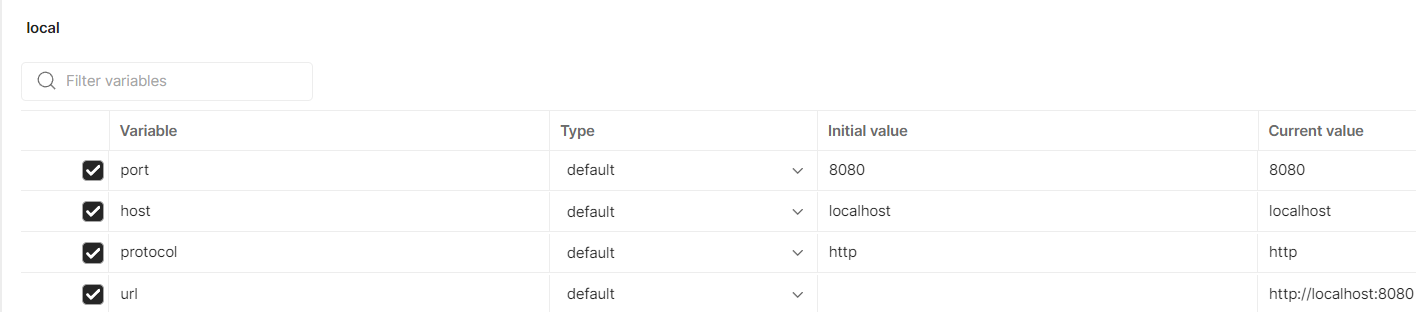
**Задание 4**

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.

1. Результаты работы
   1. Результаты задания 1

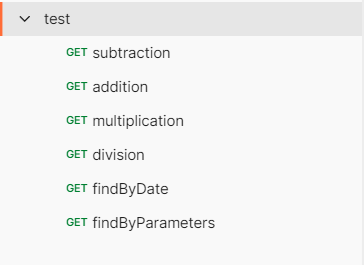
Создать несколько environments (например local и dev). Прописать для них переменные, необходимые для формирования url (хост, порт, протокол).

  
Рисунок 1 – environment dev

  
Рисунок 2 – environment local

* 1. Результаты задания 2

Создадим коллекцию, в которую добавим необходимые реквесты. Сама коллекция показана на рисунке 3.

  
Рисунок 3 – Коллекция test

На рисунках 4-9 представлены реквесты для:

* енд-поинта получения результата сложения;
* енд-поинта получения результата умножения;
* енд-поинта получения результата вычитания;
* енд-поинта получения результата деления;
* енд-поинта получения данных обо всех вычислениях за определенный момент времени;
* енд-поинта получения данных по параметрам.

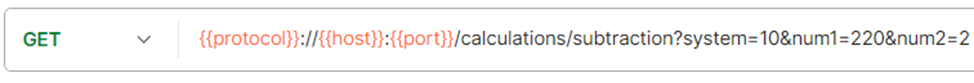
  
Рисунок 4 – Енд-поинт получения результата сложения

  
Рисунок 5 – Енд-поинт получения результата умножения

  
Рисунок 6 – Енд-поинт получения результата вычитания

  
Рисунок 7 – Енд-поинт получения результата деления

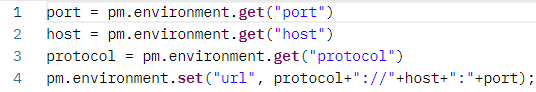
  
Рисунок 8 – Енд-поинт получения данных обо всех вычислениях за определенный момент времени

  
Рисунок 9 – Енд-поинт получения данных по параметрам

* 1. Результаты задания 3

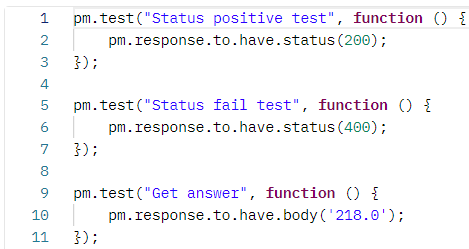
Создим pre-request script, для коллекции, который на основе выбранного environment будет формировать url.

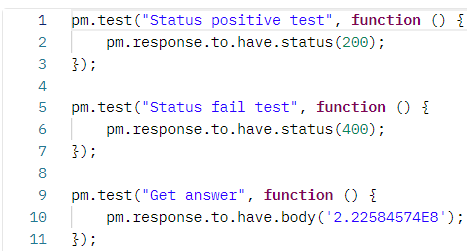
Во всех реквестах заменим захардкоженную строку подключения, на переменную, которая будет формироваться в pre-request скрипте. На рисунке 10 показан pre-request script для коллекции test.

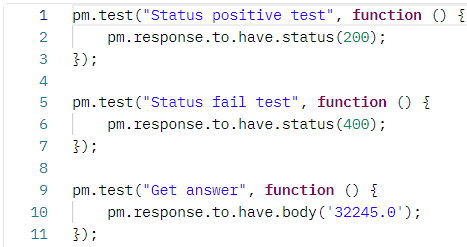
  
Рисунок 10 – pre-request script, для коллекции test

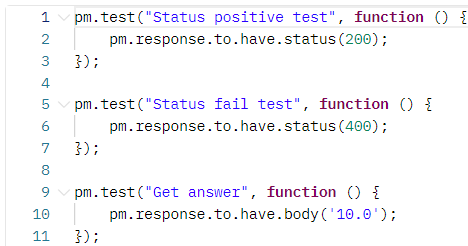
* 1. Результаты задания 4

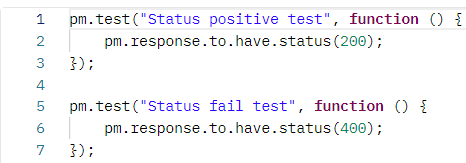
Для каждого реквеста добавим несколько тестов (как положительных, так и негативных). На рисунках 11-16 показаны тесты для необходимых конечных точек.

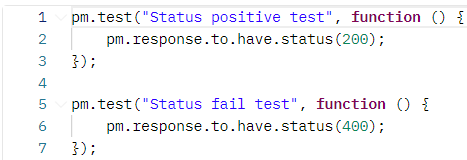
  
Рисунок 11 – Тесты для получения результата сложения

  
Рисунок 12 – Тесты для получения результата умножения

  
Рисунок 13 – Тесты для получения результата вычитания

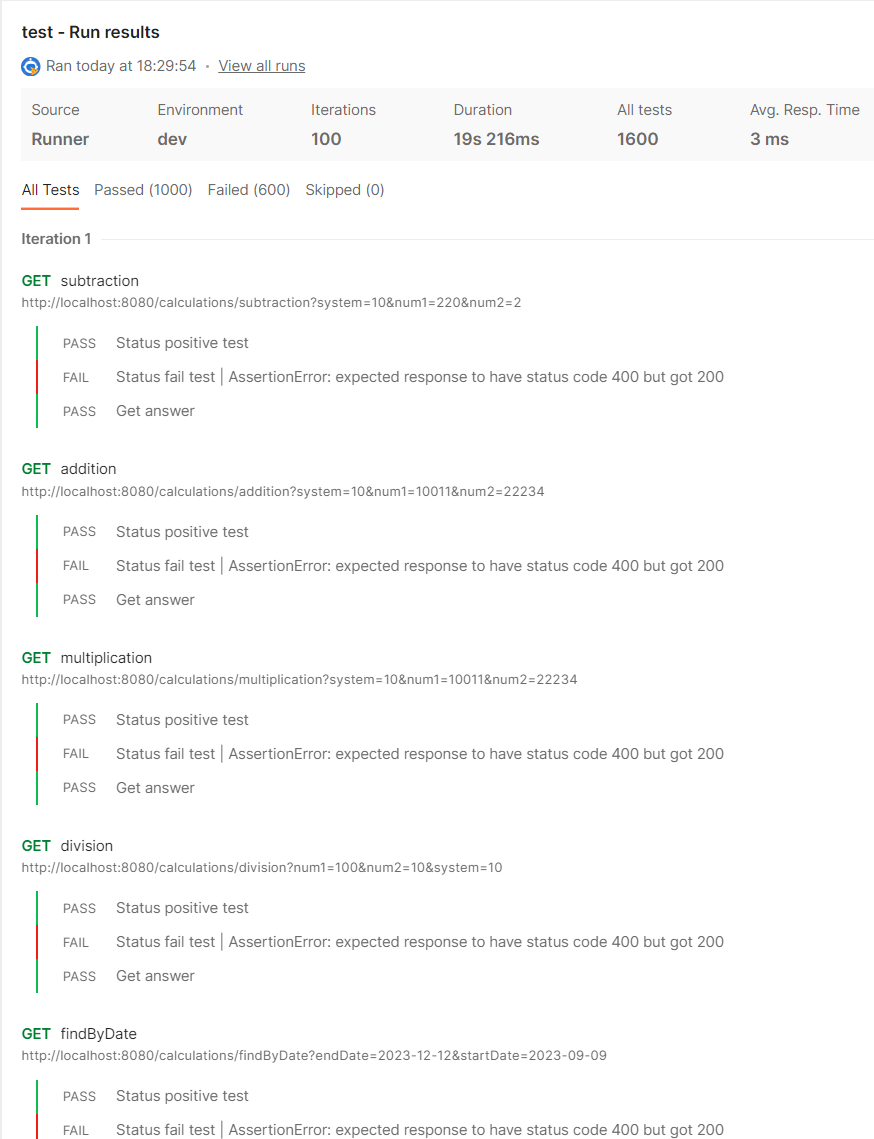
  
Рисунок 14 – Тесты для получения результата деления

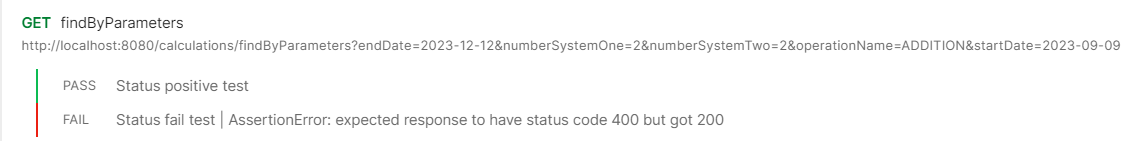
  
Рисунок 15 – Тесты для получения получения данных обо всех вычислениях за определенный момент времени

  
Рисунок 16 – Тесты для получения получения данных по параметрам

Проверим как сервер справляется с нагрузкой при последовательной отправке 100 итераций вызовов конечных точек.

На рисунке 17 показана последовательная отправка 100 итераций вызовов конечных точек (отображена 1 итерация).



  
Рисунок 17 – Последовательная отправка100 итераций вызовов конечных точек

1. Вывод

В результате выполнения лабораторной работы познакомился с возможностью автоматизированного тестирования API в Postman, а также выполнены необходимые задания, которые описаны в пункте 1.

Написан отчёт.

[Исходный код доступен по ссылке](https://github.com/Allexeey/autotests-lab1/tree/lab-45)

ПРИЛОЖЕНИЕ А  
Листинг программы

{  
 "info": {  
 "\_postman\_id": "c1e53ffb-c3b8-40a0-9770-1569bd1dffc1",  
 "name": "test",  
 "schema": "https://schema.getpostman.com/json/collection/v2.0.0/collection.json",  
 "\_exporter\_id": "24602051"  
 },  
 "item": [  
 {  
 "name": "subtraction",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Get answer\", function () {\r",  
 " pm.response.to.have.body('218.0');\r",  
 "});"  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "request": {  
 "method": "GET",  
 "header": [],  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/subtraction?system=10&num1=220&num2=2",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "subtraction"  
 ],  
 "query": [  
 {  
 "key": "system",  
 "value": "10"  
 },  
 {  
 "key": "num1",  
 "value": "220"  
 },  
 {  
 "key": "num2",  
 "value": "2"  
 }  
 ]  
 }  
 },  
 "response": []  
 },  
 {  
 "name": "multiplication",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Get answer\", function () {\r",  
 " pm.response.to.have.body('2.22584574E8');\r",  
 "});"  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "request": {  
 "method": "GET",  
 "header": [],  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/multiplication?system=10&num1=10011&num2=22234",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "multiplication"  
 ],  
 "query": [  
 {  
 "key": "system",  
 "value": "10"  
 },  
 {  
 "key": "num1",  
 "value": "10011"  
 },  
 {  
 "key": "num2",  
 "value": "22234"  
 }  
 ]  
 }  
 },  
 "response": []  
 },  
 {  
 "name": "addition",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Get answer\", function () {\r",  
 " pm.response.to.have.body('32245.0');\r",  
 "});\r",  
 ""  
 ],  
 "type": "text/javascript"  
 }  
 },  
 {  
 "listen": "prerequest",  
 "script": {  
 "exec": [  
 ""  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "protocolProfileBehavior": {  
 "disableBodyPruning": true  
 },  
 "request": {  
 "method": "GET",  
 "header": [],  
 "body": {  
 "mode": "raw",  
 "raw": "",  
 "options": {  
 "raw": {  
 "language": "json"  
 }  
 }  
 },  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/addition?system=10&num1=10011&num2=22234",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "addition"  
 ],  
 "query": [  
 {  
 "key": "system",  
 "value": "10"  
 },  
 {  
 "key": "num1",  
 "value": "10011"  
 },  
 {  
 "key": "num2",  
 "value": "22234"  
 }  
 ]  
 }  
 },  
 "response": []  
 },  
 {  
 "name": "division",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Get answer\", function () {\r",  
 " pm.response.to.have.body('10.0');\r",  
 "});"  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "request": {  
 "method": "GET",  
 "header": [],  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/division?num1=100&num2=10&system=10",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "division"  
 ],  
 "query": [  
 {  
 "key": "num1",  
 "value": "100"  
 },  
 {  
 "key": "num2",  
 "value": "10"  
 },  
 {  
 "key": "system",  
 "value": "10"  
 }  
 ]  
 }  
 },  
 "response": []  
 },  
 {  
 "name": "findByDate",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 ""  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "request": {  
 "method": "GET",  
 "header": [],  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/findByDate?endDate=2023-12-12&startDate=2023-09-09",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "findByDate"  
 ],  
 "query": [  
 {  
 "key": "endDate",  
 "value": "2023-12-12"  
 },  
 {  
 "key": "startDate",  
 "value": "2023-09-09"  
 }  
 ]  
 }  
 },  
 "response": []  
 },  
 {  
 "name": "findByParameters",  
 "event": [  
 {  
 "listen": "test",  
 "script": {  
 "exec": [  
 "pm.test(\"Status positive test\", function () {\r",  
 " pm.response.to.have.status(200);\r",  
 "});\r",  
 "\r",  
 "pm.test(\"Status fail test\", function () {\r",  
 " pm.response.to.have.status(400);\r",  
 "});\r",  
 "\r",  
 ""  
 ],  
 "type": "text/javascript"  
 }  
 }  
 ],  
 "request": {  
 "method": "GET",  
 "header": [],  
 "url": {  
 "raw": "{{protocol}}://{{host}}:{{port}}/calculations/findByParameters?endDate=2023-12-12&numberSystemOne=2&numberSystemTwo=2&operationName=ADDITION&startDate=2023-09-09",  
 "protocol": "{{protocol}}",  
 "host": [  
 "{{host}}"  
 ],  
 "port": "{{port}}",  
 "path": [  
 "calculations",  
 "findByParameters"  
 ],  
 "query": [  
 {  
 "key": "endDate",  
 "value": "2023-12-12"  
 },  
 {  
 "key": "numberSystemOne",  
 "value": "2"  
 },  
 {  
 "key": "numberSystemTwo",  
 "value": "2"  
 },  
 {  
 "key": "operationName",  
 "value": "ADDITION"  
 },  
 {  
 "key": "startDate",  
 "value": "2023-09-09"  
 }  
 ]  
 }  
 },  
 "response": []  
 }  
 ],  
 "event": [  
 {  
 "listen": "prerequest",  
 "script": {  
 "type": "text/javascript",  
 "exec": [  
 "port = pm.environment.get(\"port\")",  
 "host = pm.environment.get(\"host\")",  
 "protocol = pm.environment.get(\"protocol\")",  
 "pm.environment.set(\"url\", protocol+\"://\"+host+\":\"+port);"  
 ]  
 }  
 },  
 {  
 "listen": "test",  
 "script": {  
 "type": "text/javascript",  
 "exec": [  
 ""  
 ]  
 }  
 }  
 ]  
}