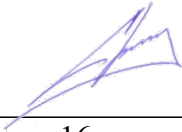


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

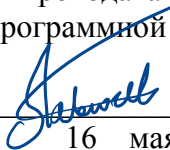
СОГЛАСОВАНО

Преподаватель департамента
программной инженерии ФКН,
кандидат компьютерных наук


____ С.А. Виденин
__16__ мая __2025__ г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»,
старший преподаватель департамента
программной инженерии


____ Н.А. Павлов
__16__ мая __2025__ г.

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ
ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ.
СЕРВЕРНАЯ ЧАСТЬ: УПРАВЛЕНИЕ УЧАСТНИКАМИ СИСТЕМЫ,
УВЕДОМЛЕНИЯМИ И ДЕЖУРСТВАМИ**

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.05-01 ТЗ 01-1-ЛУ

Исполнитель
студент группы БПИ214
____/ Е.К.Фортов/

__16__ мая __2025__ г.

Подп. и дата	
Инв. № дубл.	
Взам. Инв. №	
Подп. и дата	
Инв. № подл.	

УТВЕРЖДЕН
RU.17701729.05.05-01 ТЗ 01-1-ЛУ

**ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ
ИНЦИДЕНТАМИ В ОБЛАСТИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ.
СЕРВЕРНАЯ ЧАСТЬ: УПРАВЛЕНИЕ УЧАСТНИКАМИ СИСТЕМЫ,
УВЕДОМЛЕНИЯМИ И ДЕЖУРСТВАМИ**

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Программа и методика испытаний

RU.17701729.05.05-01 ТЗ 01-1-ЛУ

Листов 34

ГЛОССАРИЙ

1. Пользователь – человек, который использует разрабатываемую систему
2. API (Application Programming Interface) – описание способов (набор классов, методов и т. п.), которыми одна компьютерная программа (в данном случае, клиентская часть приложения) может взаимодействовать с другой (в данном случае, с сервером).
3. Юнит-тест — тест, написанный на определенный метод программы.
4. Контейнер – это модуль, в котором запускается одно приложение. Контейнеры занимают меньше памяти, используют небольшое количество ресурсов и почти не зависят от операционной системы кластера.
5. Docker — система для контейнеризации приложения.
6. Docker-compose — система для оркестрации (управления) Docker-контейнеров.
7. RESTful API — то же, что REST API.
8. Развертывание приложения – запуск приложения в какой-либо среде.
9. Эндпоинт API — это конкретный адрес (URL), по которому клиент (например, веб-приложение, мобильное приложение или другой сервис) может взаимодействовать с сервером, использующим API.
10. End-to-end тест — это тест, проверяющий функциональность всего сервиса, но не его частей.
11. Скрипт — файл с кодом, как правило, настраивающим что-то. Например, сборочный скрипт — это файл с инструкциями по разворачиванию сервиса.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Оглавление

ГЛОССАРИЙ.....	3
1. Объект испытаний.....	5
1.1. Наименование программы.....	5
1.2. Краткая характеристика области применения.....	5
2. Цель испытаний.....	6
3. Требования к программе.....	7
3.1. Требования к функциональным характеристикам.....	7
3.1.1. Требования к составу выполняемых функций.....	7
3.1.2. Требования к организации входных данных.....	8
3.1.3. Требования к организации выходных данных.....	8
3.2. Требования к интерфейсу.....	8
3.3. Требования к надежности.....	8
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	9
5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ.....	10
5.1. Технические средства, используемые во время испытаний.....	10
5.2. Программные средства, используемые во время испытаний.....	10
5.3. Порядок проведения испытаний.....	10
6. МЕТОДЫ ИСПЫТАНИЙ.....	11
6.1. Испытание выполнения требований к программной документации.....	11
6.2. Испытание выполнения требований к функциональным характеристикам.....	11
6.4. Испытание выполнения требований к надёжности.....	32
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	33
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	34

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. Объект испытаний

1.1. Наименование программы

Наименование программы – «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.».

Наименование программы на английском языке – «Information and Analytical Incident Management System in the Field of Security. Backend: Users, Notifications and Schedule Handling».

1.2. Краткая характеристика области применения

Программа будет представлять из себя три микросервиса и являться частью бекенда для системы по управлению проектами. Первый микросервис будет отвечать за профили пользователей системы вместе с группами, второй будет являться сервисом кастомизированных уведомлений, третий будет отвечать за расписание пользователей.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. Цель испытаний

Целью испытаний является проверка корректности выполнения программой функций, перечисленных в разделе «Требования к программе».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. Требования к программе

3.1. Требования к функциональным характеристикам

3.1.1. Требования к составу выполняемых функций

Программа представляет собой 3 сервиса, каждый из которых предоставляет пользователю RESTful API для взаимодействия.

Сервис профилей пользователей должен давать пользователю совершать следующие действия:

- создавать нового пользователя
- редактировать пользователя
- получать пользователя по идентификатору
- получать идентификаторы пользователей с определенным именем
- получать идентификаторы несуществующих пользователей из введенных идентификаторов
- получать идентификатор пользователя по почте и паролю
- получать пользователей по их идентификаторам
- получать документацию сервиса в виде спецификации RESTful API в формате .yaml

Сервис дежурств должен давать пользователю совершать следующие действия:

- создавать дежурство
- редактировать дежурство
- получать дежурство по идентификатору
- получать следующее дежурство пользователя
- удалять дежурство по идентификатору
- получать документацию сервиса в виде спецификации RESTful API в формате .yaml

Сервис уведомлений должен давать пользователю совершать следующие действия:

- создавать уведомление
- редактировать уведомление
- удалять уведомление по идентификатору
- удалять уведомления по идентификаторам

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- получать всю информацию об уведомлениях по их идентификаторам
- получать документацию сервиса в виде спецификации RESTful API в формате .yaml

3.1.2. Требования к организации входных данных

Входные данные — это HTTP-запросы к сервисам. Тела запросов сервисов должны иметь JSON-формат. Кодировка символов — UTF-8.

Более детальное описание формата входных данных представлено в спецификациях REST API (файлы .yaml, находятся в поддиректории API_DOCS корневой директории сервисов).

Также в системе предусмотрено автоматизированное юнит-тестирование, которое проверяет работоспособность отдельных методов классов.

3.1.3. Требования к организации выходных данных

Результат работы системы представляет собой корректный HTTP-ответ с соответствующим статусом.

Для юнит-тестов выходные данные — это значение, которое вернул тестируемый метод.

3.2. Требования к интерфейсу

Графический интерфейс у сервисов отсутствует, так как все сервисы являются исключительно серверной частью системы.

3.3. Требования к надежности

Дополнительные требования к надежности не предъявляются, кроме тех, которые предъявляются к используемым библиотекам и фреймворкам. При корректном использовании фреймворка (то есть при обеспечении необходимых условий для корректной отработки используемых функций библиотек) программа не должна завершаться аварийно.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации:

- «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Техническое задание (ГОСТ 19.201-78 [1])
- «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Программа и методика испытаний (ГОСТ 19.301-79 [2])
- «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Текст программы (ГОСТ 19.401-78 [3])
- «Информационно-аналитическая система управления инцидентами в области обеспечения безопасности. Серверная часть: управление участниками системы, уведомлениями и дежурствами.». Руководство оператора (ГОСТ 19.505-79 [3])

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1. Технические средства, используемые во время испытаний

Ноутбук Macbook Pro 14 дюймов, M1.

5.2. Программные средства, используемые во время испытаний

На ноутбуке имеется:

- ОС Mac OS Ventura 13.2.1
- ОЗУ 16 ГБ
- ЦП: Apple M1 Pro с 2 ядрами эффективности и 6 ядрами производительности, тактовая частота - 3.2 ГГц
- SSD 512Гб

5.3. Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

- проверка требований к программной документации,
- проверка требований к функциональным характеристикам,
- проверка требований к интерфейсу,
- проверка требований к надёжности.

5.4. Загрузка программы для испытания

Тестирование трех модулей — модуля профилей, модуля дежурств и модуля уведомлений — проводится локально, сервисы развернуты в контейнерах Docker, оркестратором контейнеров выступает docker-compose.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. МЕТОДЫ ИСПЫТАНИЙ

6.1. Испытание выполнения требований к программной документации

Состав программной документации проверяется с помощью визуально, проверяется наличие программной документации в системе LMS. Также визуально проверяется соответствие документации требованиям ГОСТ.

Все документы удовлетворяют представленным требованиям.

6.2. Испытание выполнения требований к функциональным характеристикам

Испытание выполнения функциональных требований проводилось в программе Postman с написанными на JavaScript end-to-end тестами в рамках postman-коллекции. Ниже приведены скриншоты с подробным описанием тестов, а также с их результатами.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

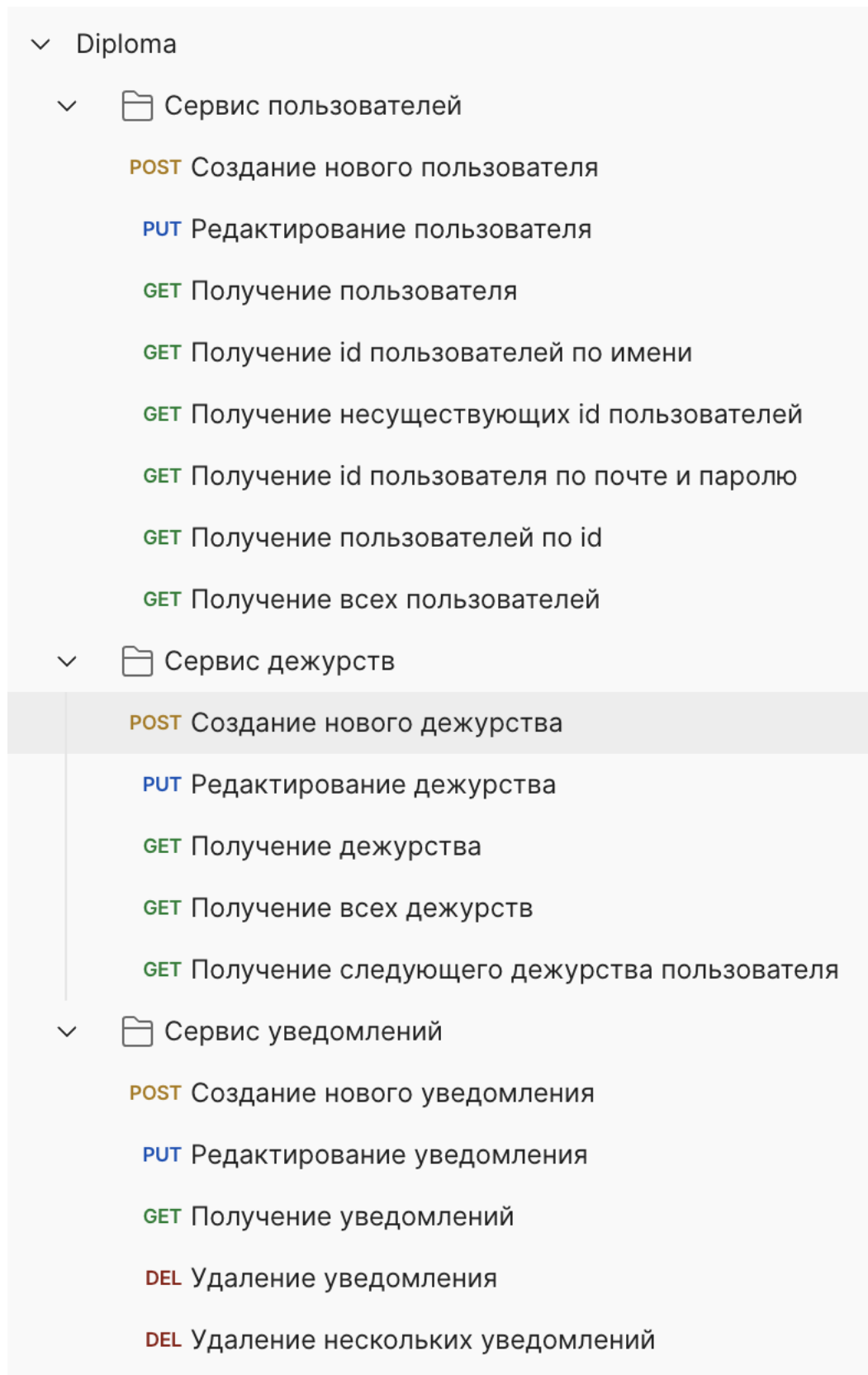


Рисунок 1. Список Postman-тестов для тестирования разработанных сервисов.

Сервис профилей пользователей, тесты:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

POST

▼

http://localhost:9090/users

ParamsAuthorizationHeaders (10)Body ●Scripts ●Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐

1

{

2

"name": "Иван Иванов",

3

"email": "ivan@example.com",

4

"phone": "+79123456789",

5

"password": "12345678"

6

}

Рисунок 2. Сервис пользователей. Тест на создание нового пользователя.

Ограничения на входные данные:

- name:
 - не пусто
 - длина: от 2 до 250 символов включительно
- email:
 - присутствует
 - соответствие формату почты
 - длина: от 6 до 100 символов включительно
 - уникальность
- phone:
 - присутствует
 - соответствует формату телефона
 - уникальность
- password:
 - не пусто
 - длина: от 8 до 40 символов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

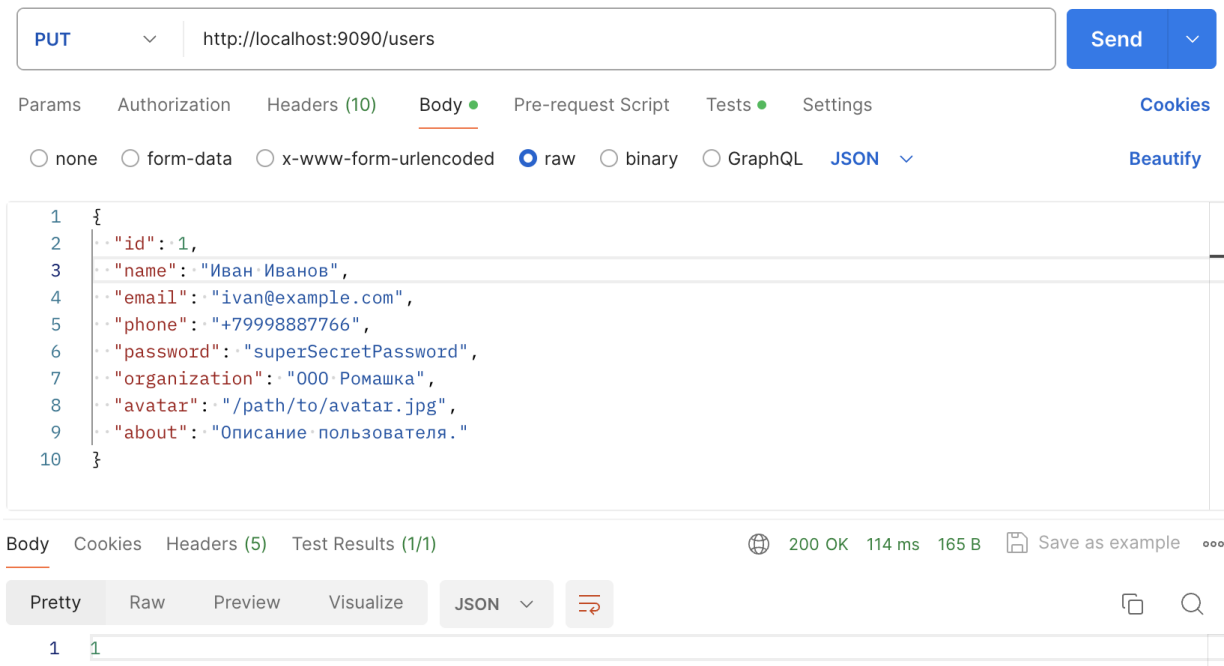


Рисунок 3. Сервис пользователей. Тест на редактирование пользователя.

Ограничения на входные данные:

- id:
 - присутствует
 - ≥ 0
- email:
 - уникальность
- телефон:
 - уникальность

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

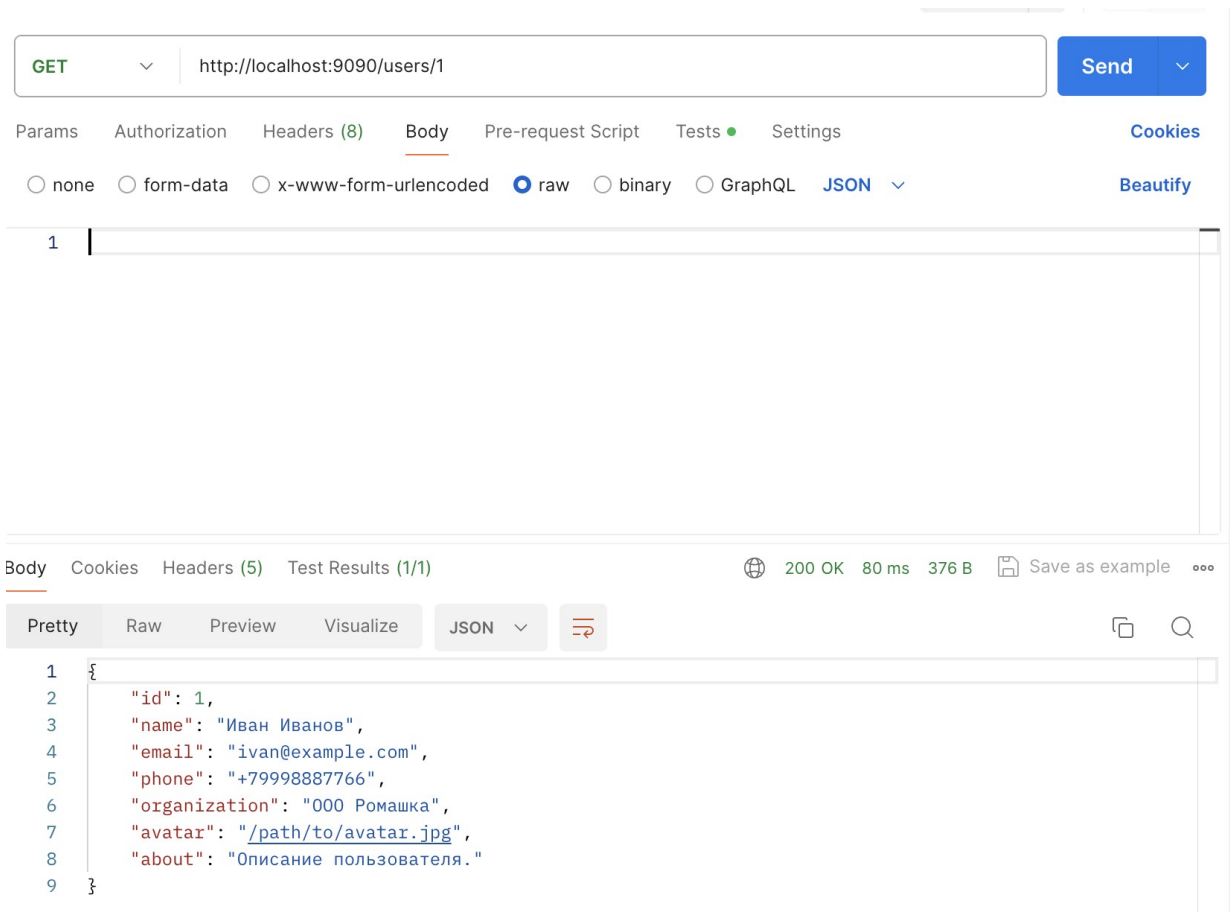


Рисунок 4. Сервис пользователей. Тест на получение пользователя по id.

Ограничения на входные данные:

- `id`:
 - `> 0`

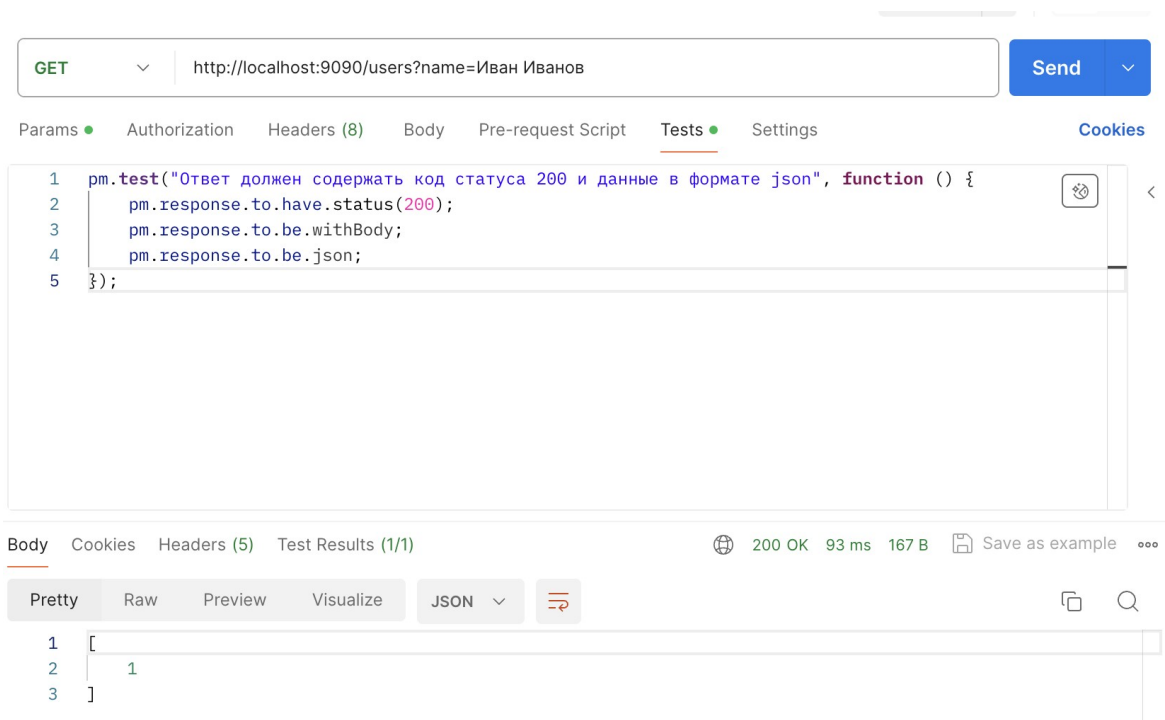


Рисунок 5. Сервис пользователей. Тест на получение пользователей по имени.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

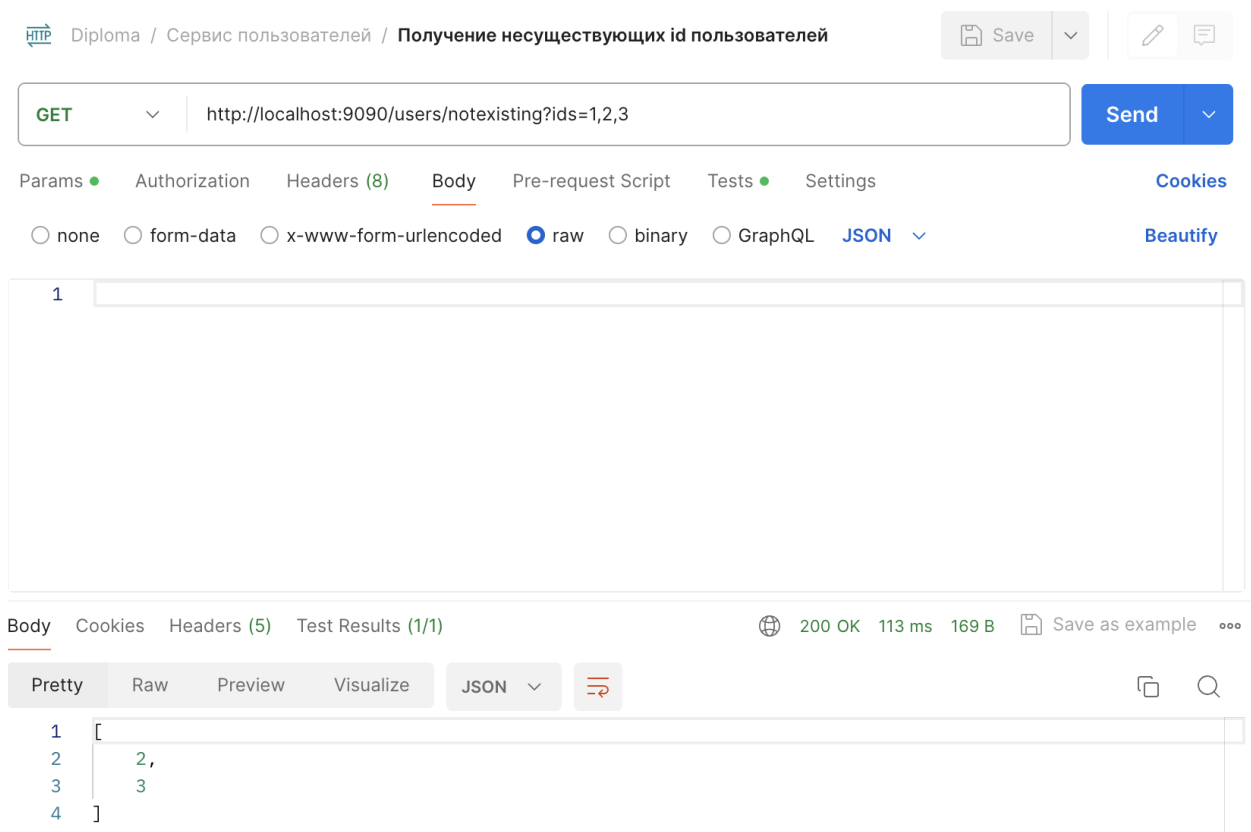


Рисунок 6. Сервис пользователей. Тест на получение несуществующих id.

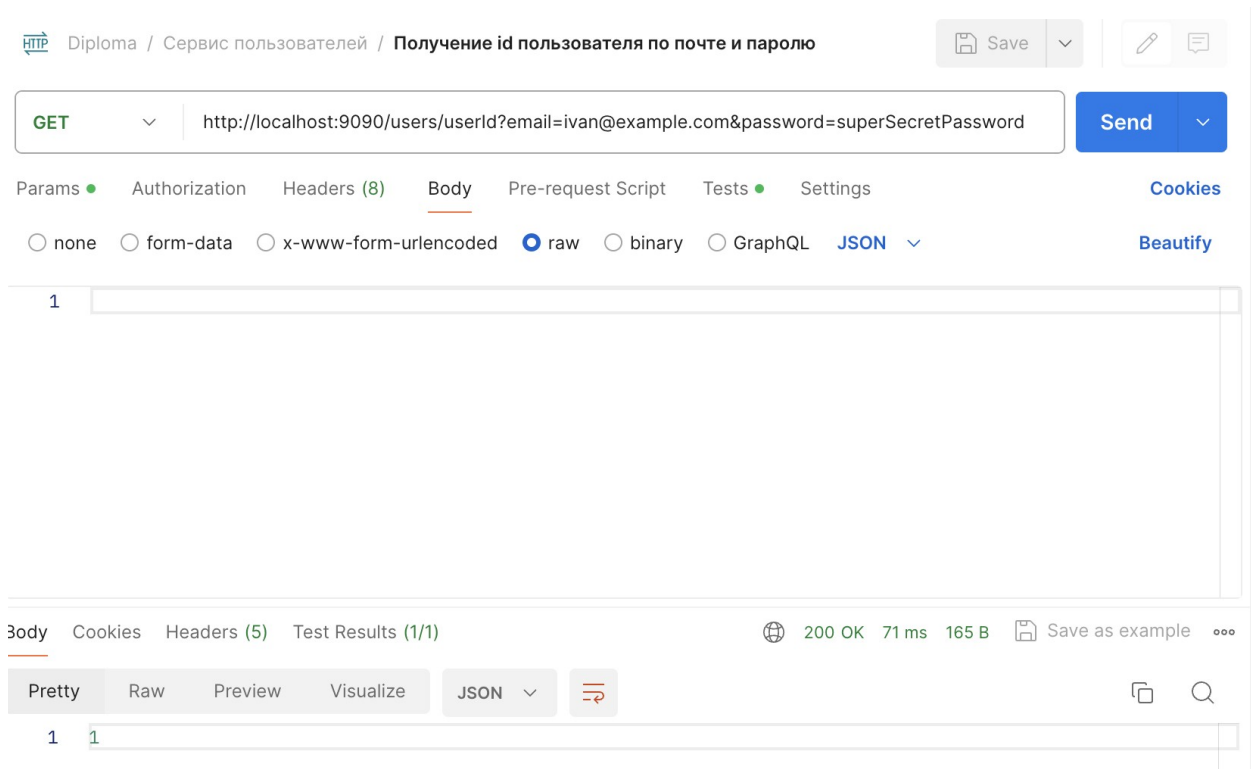


Рисунок 7. Тест на получение id пользователя по почте (уникальна) и паролю.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

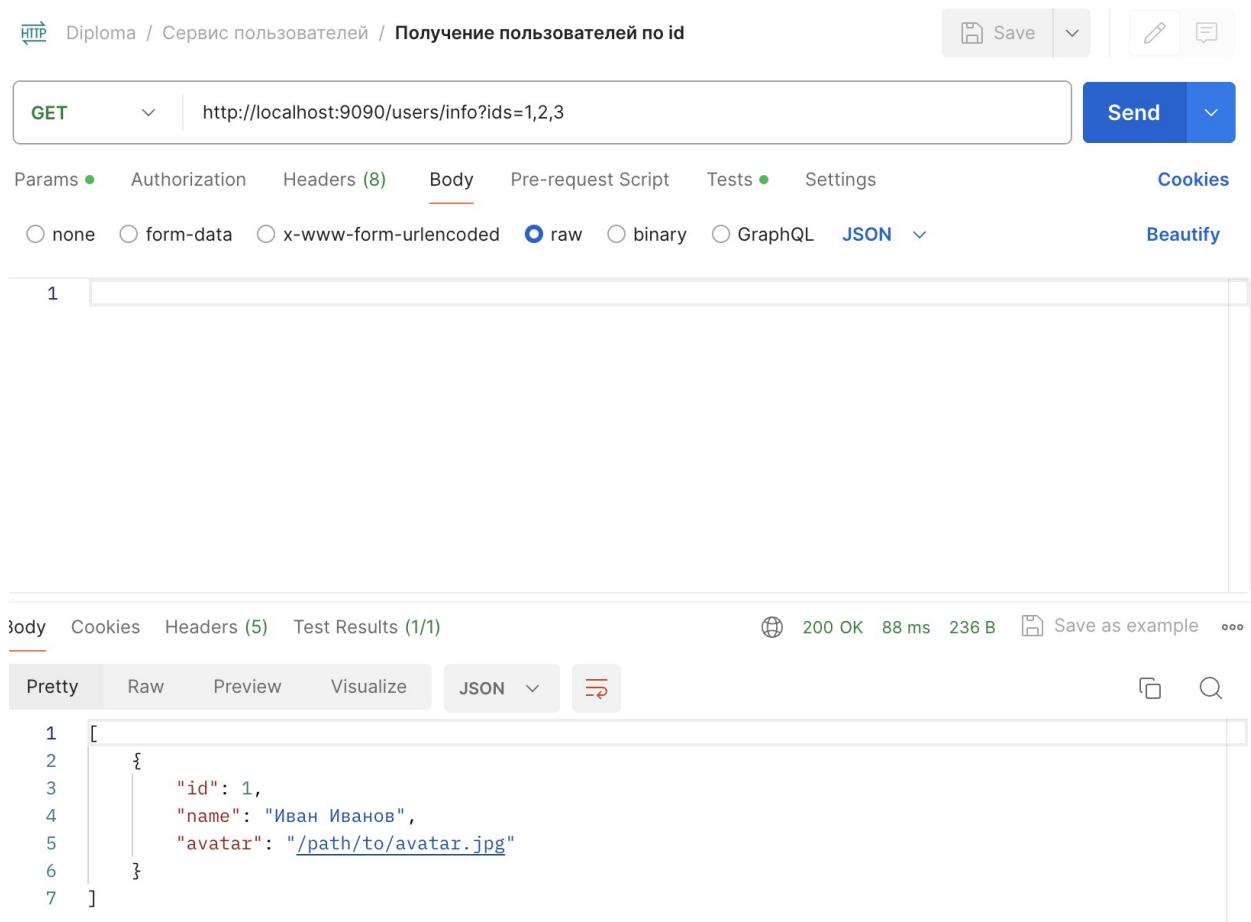


Рисунок 8. Тест по получение пользователей по id.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

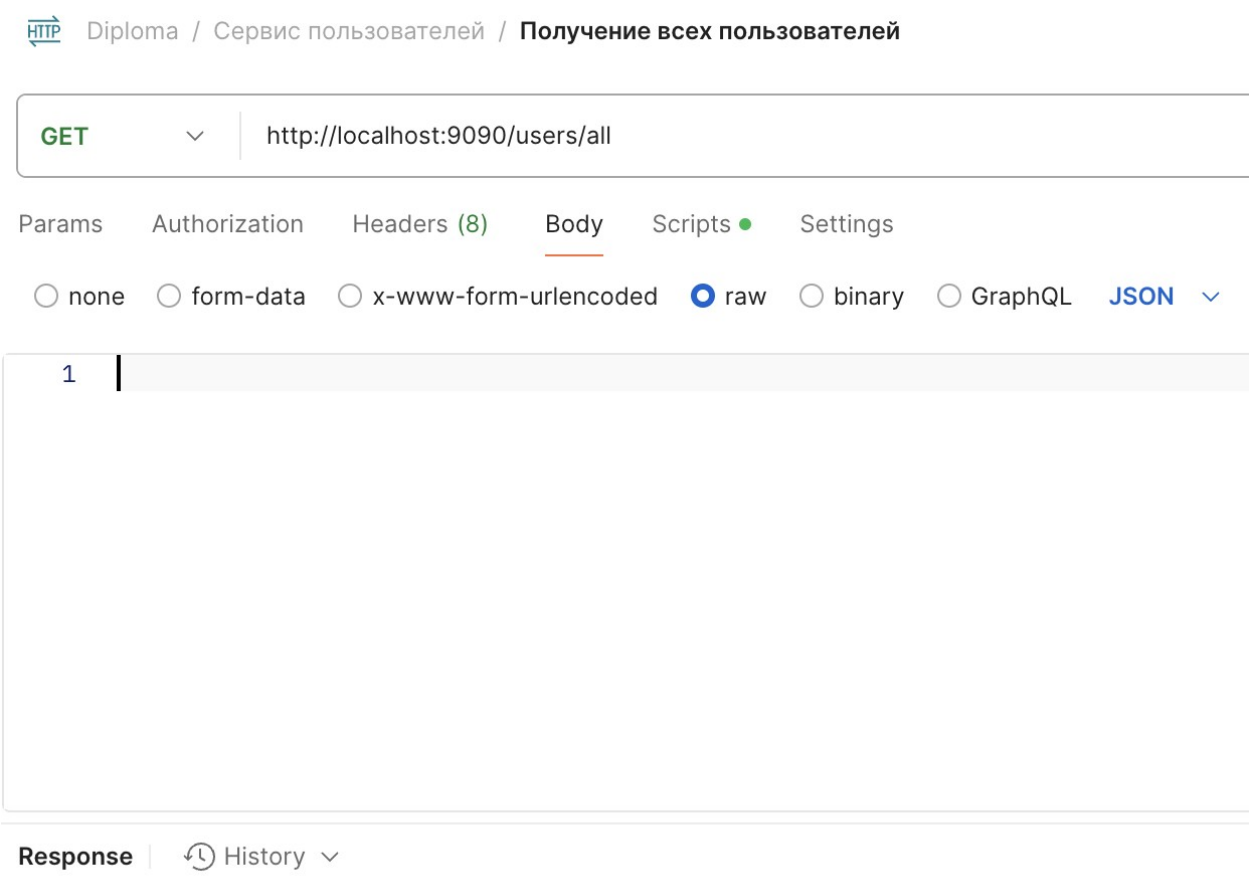


Рисунок 9. Тест на получение всех пользователей.

Результат выполнения тестов: все тесты успешно пройдены.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Diploma - Run results

Run Again

Automate Run

+ New Run

Export Results

Ran today at 19:42:51 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	1s 215ms	7	90 ms

All Tests

Passed (7)

Failed (0)

Skipped (0)

[View Summary](#)

Iteration 1

POST

Создание нового пользователя

http://localhost:9090/users

201 Created 387 ms 258 B

PASS

Ответ должен содержать код статуса 201 и данные в формате json

PUT

Редактирование пользователя

http://localhost:9090/users

200 OK 136 ms 165 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение пользователя

http://localhost:9090/users/1

200 OK 11 ms 376 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение id пользователей по имени

http://localhost:9090/users?name=Иван Иванов

200 OK 23 ms 167 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json


GET

Получение несуществующих id пользователей

Рисунок 10. Сервис пользователей. Результат запуска всей коллекции тестов: успешно (все).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Сервис дежурств, тесты:

 Diploma / Сервис дежурств / **Создание нового дежурства**

POST

▼

http://localhost:7070/duties

ParamsAuthorizationHeaders (10)Body●Scripts●Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

1

{

2

"start_time": "2024-01-01T12:00:00",

3

"name": "duty name",

4

"interval": "PT8H",

5

"ids": [

6

1

7

]

8


}

BodyCookiesHeaders (5)Test Results (1/1)

201 Created

{ } JSON ▼

▶ Preview

 Visualize ▼

1

{

2

"id": 1,

3

"name": "duty name",

4

"start_time": "2024-01-01T12:00:00",

5

"interval": "PT8H",

6

"ids": [

7

1

8

]

9

}

Рисунок 11. Сервис дежурств. Тест на создание дежурства.

Ограничения на входные данные:

- start_time:
 - должно присутствовать
 - формат: уууу-ММ-ддТ'НН:мм:сс

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- name:
 - не пусто
- interval:
 - должен присутствовать
 - формат: PnYnMnDTnHnMnS

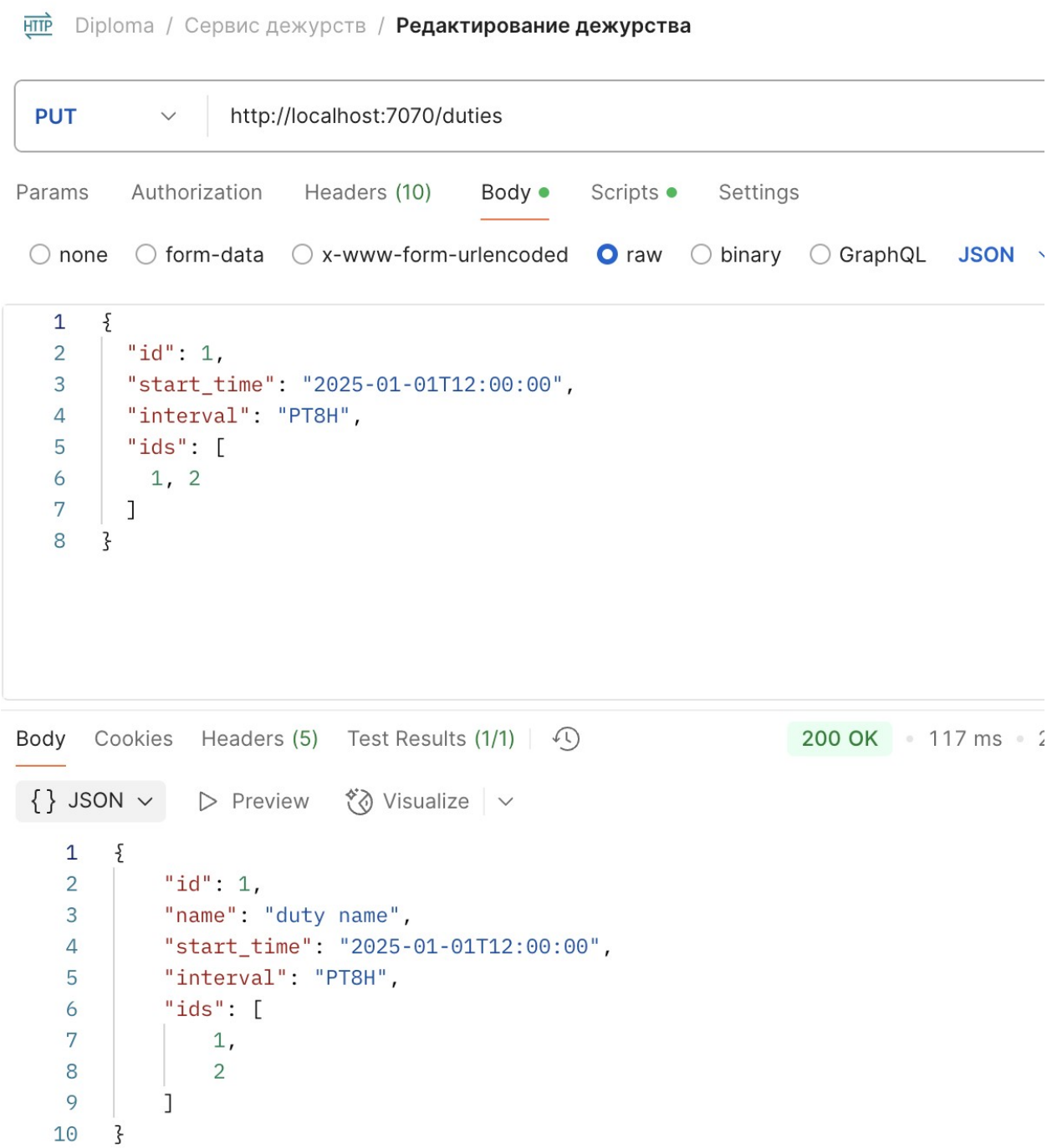


Рисунок 12. Сервис дежурств. Тест на редактирование дежурства.

Ограничения на входные данные:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- id:
 - должен присутствовать
- start_time (если присутствует):
 - формат: уууу-ММ-дд'T'HH:mm:ss
- interval (если присутствует):
 - формат: PnYnMnDTnHnMnS

HTTP

Diploma / Сервис дежурств / Получение дежурства

GET

▼

http://localhost:7070/duties/1

Params

Authorization

Headers (8)

Body

Scripts ●

Settings

Query Params

	Key	Value	Description
	Key	Value	Description

Body

Cookies

Headers (5)

Test Results (1/1)

🕒

200 OK

• 40 ms

• 278 B

•

{ } JSON ▼

▶ Preview

🔗 Visualize ▼

1

2

3

4

5

6

7

8

9

10

11

{

" id ": 1,

" name ": " duty name ",

" start_time ": " 2025-01-01T12:00:00 ",

" interval ": " PT8H ",

" ids ": [

1,

2

],

" currentDutyUserId ": 1

}

Рисунок 13. Сервис дежурств. Тест на получение дежурства по id.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[HTTP](#) Diploma / Сервис дежурств / **Получение всех дежурств**

GET

▼

http://localhost:7070/duties

ParamsAuthorizationHeaders (8)BodyScripts ●Settings

Query Params

	Key	Value	Description
	Key	Value	Description

BodyCookiesHeaders (5)Test Results (1/1)↺

200 OK • 32 ms • 280 B • 🌐

{ } JSON ▼▶ Preview🔗 Visualize ▼

1[

2 {

3 "id": 1,

4 "name": "duty name",

5 "start_time": "2025-01-01T12:00:00",

6 "interval": "PT8H",

7 "ids": [

8 1,

9 2

10],

11 "currentDutyUserId": 1

12 }

13]

Рисунок 14. Сервис дежурств. Тест на получение всех дежурств.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

HTTP Diploma / Сервис дежурств / Получение следующего дежурства пользователя

GET

▼

http://localhost:7070/duties/user/1

ParamsAuthorizationHeaders (8)BodyScripts ●Settings

Query Params

	Key	Value	Description
	Key	Value	Description

BodyCookiesHeaders (5)Test Results (1/1)🕒200 OK • 17 ms • 252 B •

{ } JSON ▼▶ Preview🔗 Visualize ▼

```
1 {
2   "nextDutyDate": "2025-01-01T12:00:00",
3   "duration": "PT8H",
4   "intervalBetweenDuties": "PT16H"
5 }
```

Рисунок 15. Сервис дежурств. Тест на получение следующего дежурства пользователя.

Результат выполнения тестов: все тесты успешно пройдены.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Iteration 1

POST

Создание нового дежурства

http://localhost:7070/duties

201

• 273 ms • 259 B

PASS

Ответ должен содержать код статуса 201 и данные в формате json

PUT

Редактирование дежурства

http://localhost:7070/duties

200

• 38 ms • 256 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение дежурства

http://localhost:7070/duties/1

200

• 19 ms • 278 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение всех дежурств

http://localhost:7070/duties

200

• 8 ms • 280 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение следующего дежурства пользователя

http://localhost:7070/duties/user/1

200

• 13 ms • 252 B

Рисунок 16. Сервис дежурств. Результат запуска всей коллекции тестов: успешно (все).

Сервис уведомлений, тесты:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

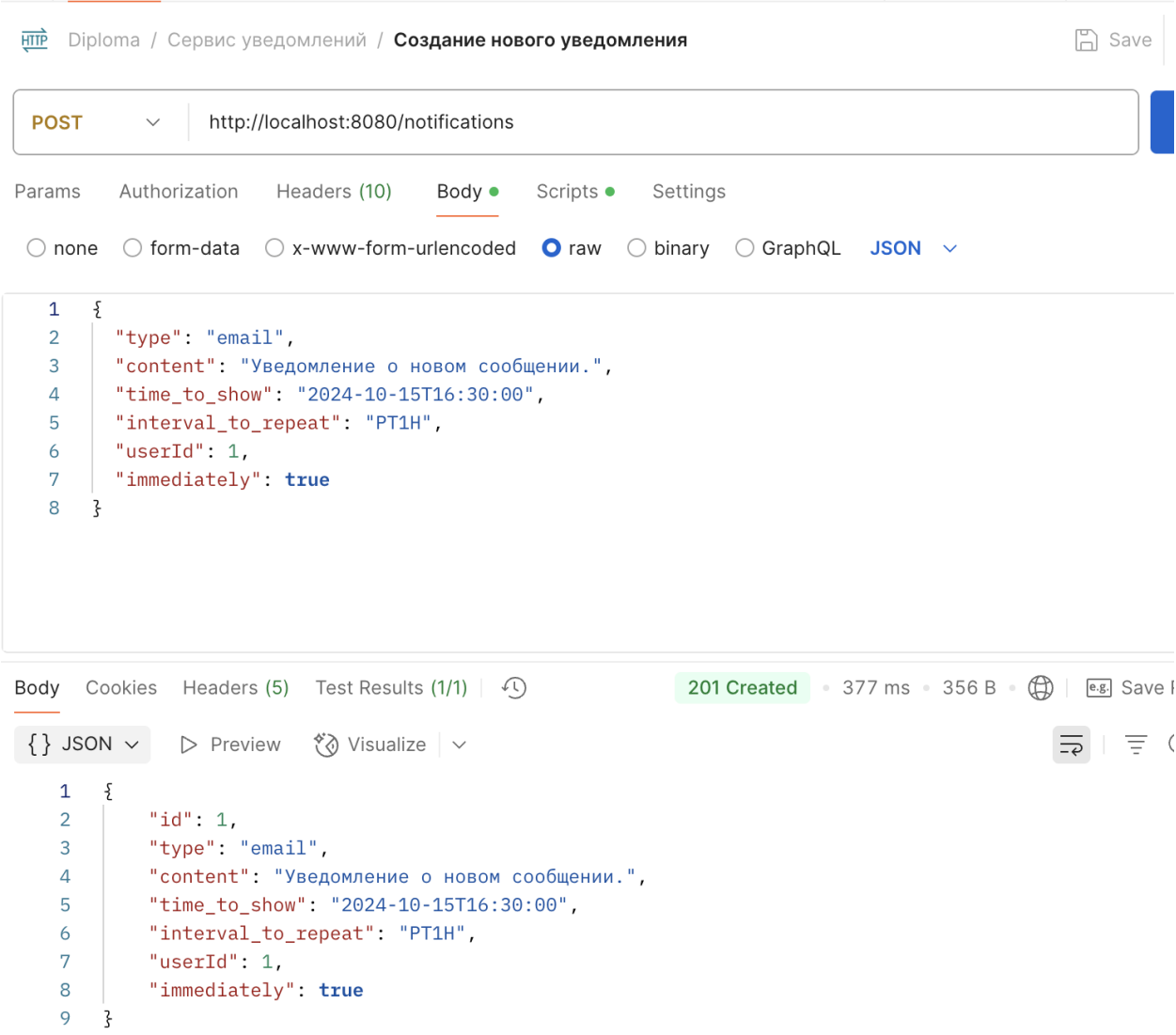


Рисунок 17. Сервис уведомлений. Тест на создание нового уведомления.

Ограничения на входные данные:

- type:
 - присутствует
 - допустимые значения: «sms» и «email»
- content:
 - присутствует
- userId:
 - присутствует
- time_to_show (если присутствует):
 - формат: "yyyy-MM-dd'T'HH:mm:ss"
- недопустимо, чтобы immediately = false И time_to_show = null
- interval_to_repeat (если присутствует):
 - формат: PnYnMnDTnHnMnS

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

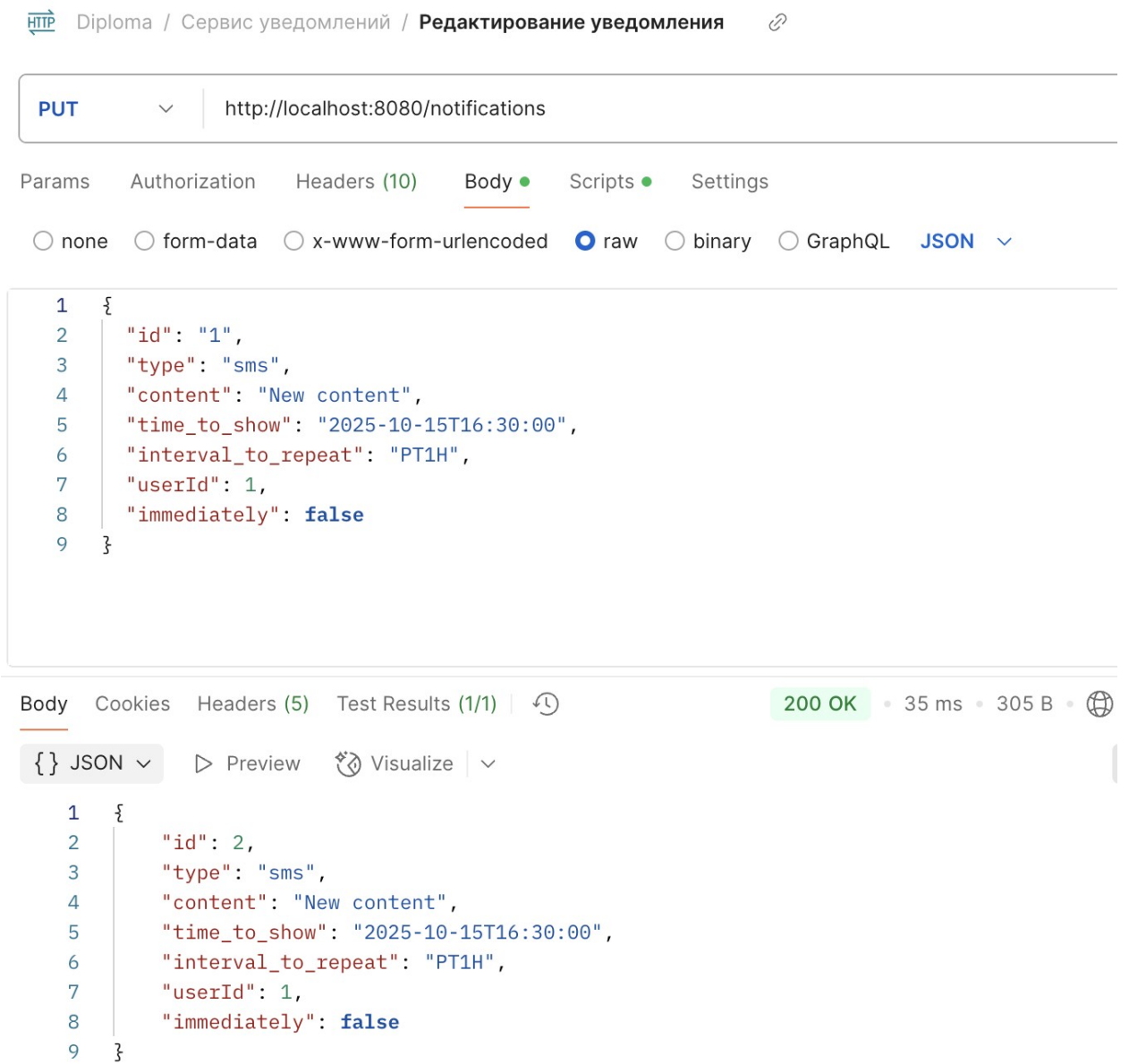


Рисунок 18. Сервис уведомлений. Тест на редактирование уведомления.

Ограничения на входные данные:

- type:
 - допустимые значения: «sms» и «email» (если присутствует)
- time_to_show (если присутствует):
 - формат: "yyyy-MM-dd'T'HH:mm:ss"
- недопустимо, чтобы immediately = false И time_to_show = null

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[HTTP](#) Diploma / Сервис уведомлений / **Получение уведомлений**

GET

⌵

http://localhost:8080/notifications/1,2,3

Params

Authorization

Headers (10)

Body ●

Scripts ●

Settings

Query Params

	Key	Value	Description
	Key	Value	Description

Body

Cookies

Headers (5)

Test Results (1/1)

🕒

200 OK

• 39 ms

• 495 B

🌐

{ } JSON ⌵

▶ Preview

🔗 Visualize ⌵

☰

1

[

2

{

3

" id ": 1,

4

" type ": " email ",

5

" content ": " Уведомление о новом сообщении. ",

6

" time_to_show ": " 2024-10-15T16:30:00 ",

7

" interval_to_repeat ": " PT1H ",

8

" user_id ": 1,

9

" immediately ": true

10

},

11

{

12

" id ": 2,

13

" type ": " sms ",

14

" content ": " New content ",

Рисунок 19. Сервис уведомлений. Тест на получение уведомлений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

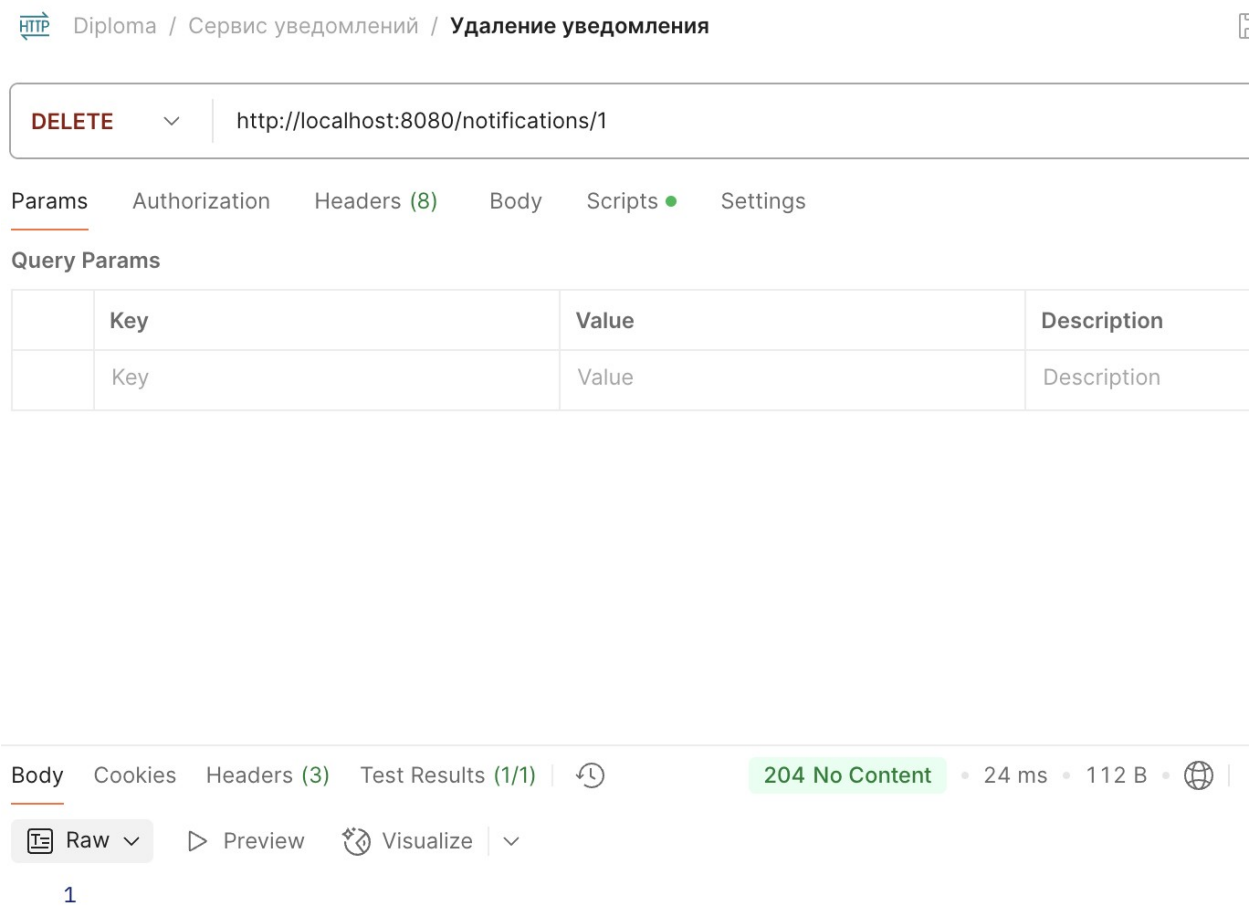


Рисунок 20. Сервис уведомлений. Тест на удаление уведомления.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

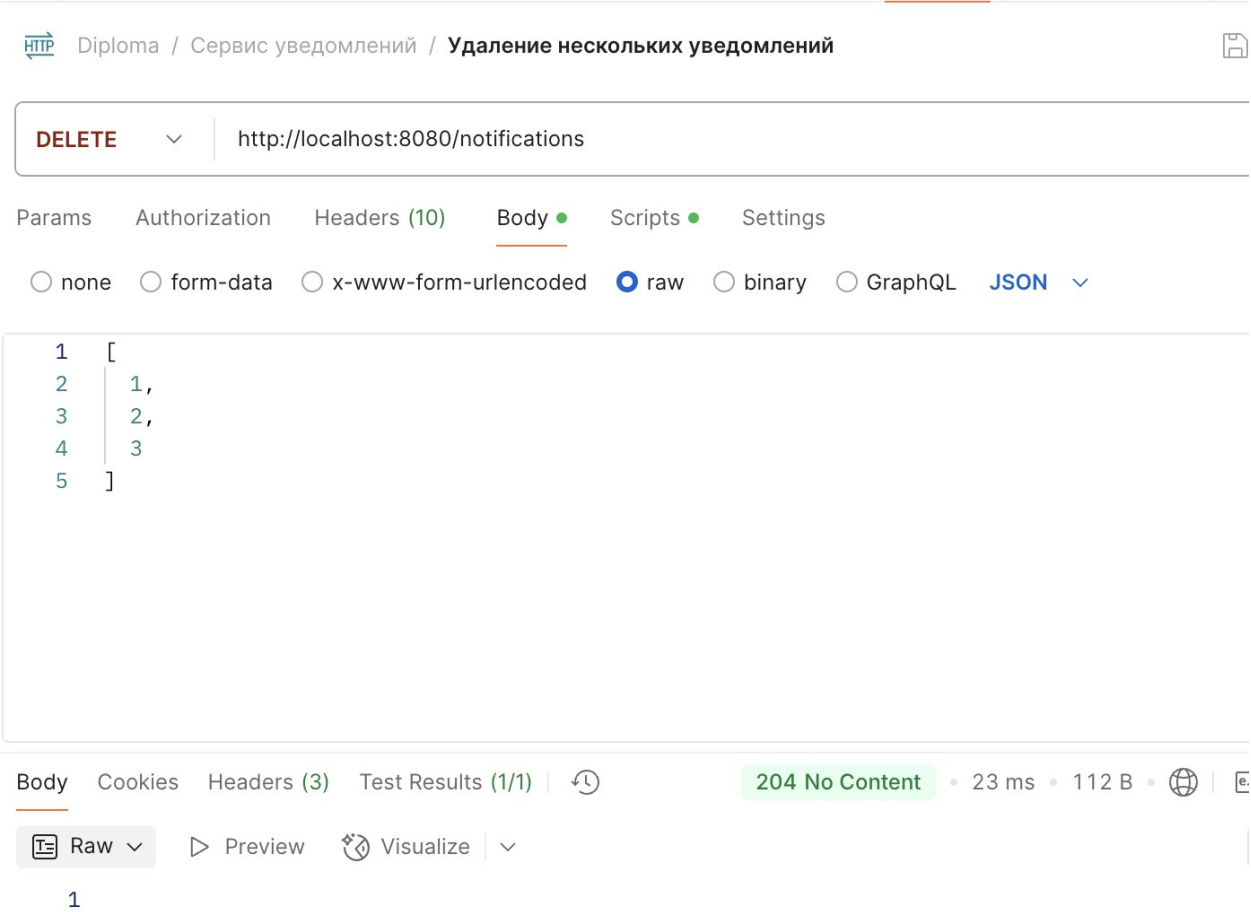


Рисунок 21. Сервис уведомлений. Тест на удаление нескольких уведомлений.

Результат выполнения тестов: все тесты успешно пройдены.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Diploma - Run results

Run Again

Automate Run

+ New Run

Export Results

Ran today at 08:57:13 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	none	1	626ms	5	54 ms

All Tests

Passed (5)

Failed (0)

Skipped (0)

View Summary

Iteration 1

POST

Создание нового уведомления

http://localhost:8080/notifications

201

218 ms

356 B

PASS

Ответ должен содержать код статуса 201 и данные в формате json

PUT

Редактирование уведомления

http://localhost:8080/notifications

200

28 ms

305 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

GET

Получение уведомлений

http://localhost:8080/notifications/1,2,3

200

9 ms

495 B

PASS

Ответ должен содержать код статуса 200 и данные в формате json

DELETE

Удаление уведомления

http://localhost:8080/notifications/1

204

7 ms

112 B

PASS

Ответ должен содержать код статуса 204 и данные в формате json

DELETE

Удаление нескольких уведомлений

http://localhost:8080/notifications

204

9 ms

112 B

PASS

Ответ должен содержать код статуса 204

Рисунок 22. Сервис уведомлений. Результат запуска всей коллекции тестов: успешно (все).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.3. Испытание выполнения требований к интерфейсу

Требования к интерфейсу проверяются визуально. Представленный интерфейс приложения в предыдущем разделе удовлетворяет всем требованиям.

6.4. Испытание выполнения требований к надёжности

По файлам логов было зафиксировано, что в процессе проведения тестирования ни один сервис не был завершён аварийно.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.05-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.301-79 Программа и методика испытаний. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.401-78 Текст программы. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.505-79 Руководство оператора. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.04-01 51				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

[illegible]