

КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное  
бюджетное профессиональное образовательное учреждение  
«Колледж информационных технологий»

## **ОТЧЁТ**

**по модулю МДК 01.03 «Разработка мобильных приложений»**

**Специальность 09.02.07  
«Информационные системы и программирование»**

**Специализация:  
«Программист»**

**Отчет по учебной практике**

Студент группы 493:

Лукьянов И. А.

Преподаватель: Фомин А.В.

Санкт-Петербург 2023

# СОДЕРЖАНИЕ

1. Интерфейс пользователя .....	4
1.1. Авторизация пользователя.....	4
1.2. Регистрация нового пользователя.....	5
1.3. Список мест.....	6
1.4. Место.....	7
1.5. Список счетчиков.....	8
1.6. Счетчик .....	8
1.7. Список тарифов .....	9
1.8. Тариф.....	9
1.9. Список измерений.....	10
1.10. Измерение .....	11
2. Структура базы данных .....	12
2.1. Таблица Settings .....	12
2.2. Таблица APIEndPoint .....	12
2.3. Таблица Session .....	13
3. Описание протокола взаимодействия.....	14
3.1. Контроль сессий .....	16
3.1.1. Функция sign_in .....	16
3.1.2. Функция sign_out.....	17
3.1.3. Функция register_account .....	18
3.1.4. Функция update_password .....	19
3.1.5. Функция delete_account .....	20
3.2. Управление местами.....	20
3.2.1. Функция add_location .....	20
3.2.2. Функция get_locations.....	21
3.2.3. Функция update_location .....	22
3.2.4. Функция delete_location .....	23
3.3. Управление счетчиками .....	24
3.3.1. Функция add_counter.....	24
3.3.2. Функция get_counters .....	26
3.3.3. Функция update_counter .....	28
3.3.4. Функция delete_counter.....	29
3.4. Управление тарифами .....	30
3.4.1. Функция add_rate.....	30
3.4.2. Функция get_rates .....	31

3.4.3.	Функция update_rate .....	32
3.4.4.	Функция delete_rate.....	33
3.5.	Управление измерениями.....	33
3.5.1.	Функция add_measurement.....	33
3.5.2.	Функция get_measurements .....	35
3.5.3.	Функция update_measurement .....	36
3.5.4.	Функция delete_measurement.....	36
Работа приложения .....		38

# 1. Интерфейс пользователя

Приложение состоит из основных Activity:

- Authorization (авторизация) – стартовая Activity, которая служит для ввода данных учетной записи пользователя;
- Registration (регистрация) – можно запустить из Activity авторизации, служит для регистрации учетной записи пользователя;
- Menu (меню) – запускается после успешной авторизации, служит для выбора соответствующего списка.
- Locations (список мест) – служит для отображения списка мест пользователя, с возможностью добавления нового места;
- Location (место) – запускается после выбора места из списка, служит для отображения данных одного места с возможностью редактирования и удаления.
- Counters (список счетчиков) – служит для отображения списка мест пользователя, с возможностью добавления нового счетчика;
- Counter (счетчик) – запускается после выбора счетчика из списка, служит для отображения данных одного счетчика с возможностью редактирования и удаления.
- Rates (список тарифов) – служит для отображения списка тарифов пользователя, с возможностью добавления нового тарифа;
- Rate (тариф) – запускается после выбора тарифа из списка, служит для отображения данных одного тарифа с возможностью редактирования и удаления.
- Measurements (список измерений) – служит для отображения списка измерений пользователя, с возможностью добавления нового измерения;
- Measurement (измерение) – запускается после выбора измерения из списка, служит для отображения данных одного измерения с возможностью редактирования и удаления.

## 1.1. Авторизация пользователя

На рисунке 1 показан макет внешнего вида формы авторизации пользователя.

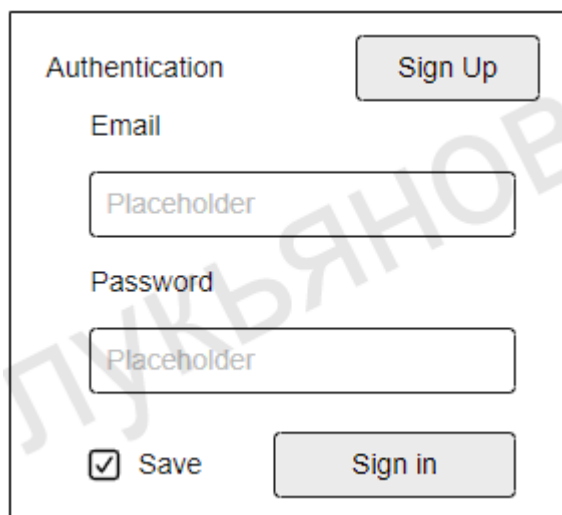
A mockup of a user authentication form. It features a title 'Authentication' at the top left and a 'Sign Up' button at the top right. Below the title are two input fields: 'Email' and 'Password', each with a 'Placeholder' text. At the bottom left, there is a checked checkbox labeled 'Save'. At the bottom right, there is a 'Sign in' button.

Рисунок 1 – Activity для авторизации пользователя

На рисунке 2 показан внешний вид формы авторизации в приложении.

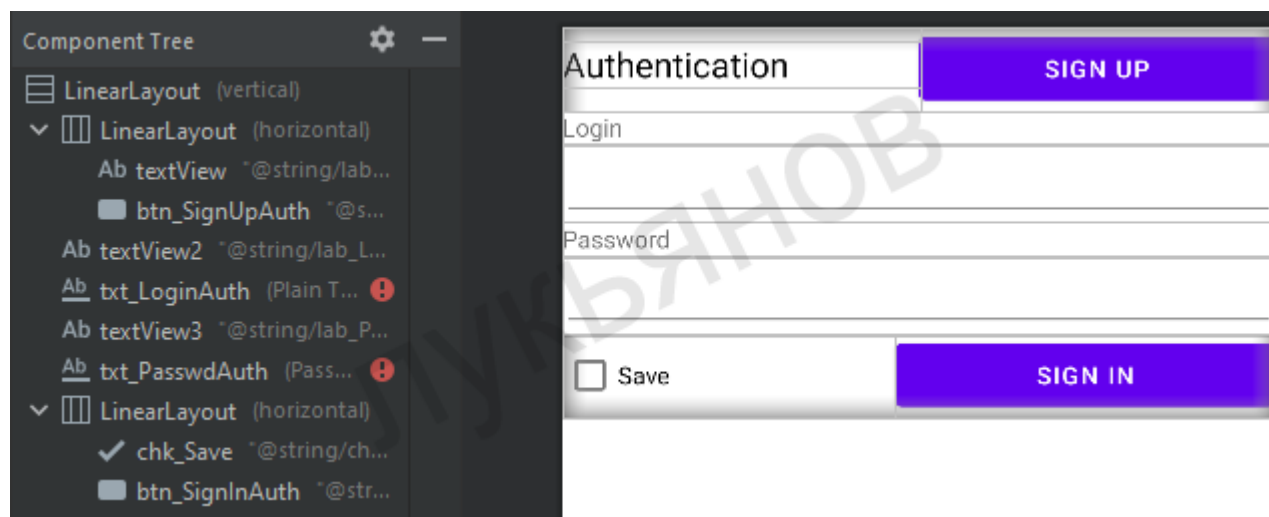


Рисунок 2 – Activity для авторизации пользователя в приложении

На экране присутствуют компоненты EditText для ввода имени пользователя и пароля, компонент CheckBox для возможности сохранения данных учетной записи в локальной базе данных SQLite3 и кнопки входа и регистрации.

## 1.2. Регистрация нового пользователя

На рисунке 3 показан макет внешнего вида формы регистрации пользователя.

Registration

Sign In

Email

Placeholder

Password

Placeholder

Sign Up

Рисунок 3 – Activity для регистрации пользователя

На рисунке 4 показан внешний вид формы регистрации в приложении.

Registration

SIGN IN

Login

Password

SIGN UP

Рисунок 4 – Activity для регистрации пользователя в приложении

На экране присутствуют компоненты EditText для ввода имени пользователя, пароля, а также кнопки регистрации и входа.

### 1.3. Список мест

На рисунке 5 показан макет внешнего вида формы списка мест.

LOCATIONS	
ADD	
ID	TITLE
1	Placeholder
2	Placeholder
3	Placeholder
4	Placeholder

Рисунок 5 – Activity для списка мест

На рисунке 6 показан внешний вид формы списка мест в приложении.



Рисунок 6 – Activity для списка мест в приложении

На экране присутствуют компонент `ListView` для отображения списка мест и выбора места, а также кнопки добавления нового места.

## 1.4. Место

На рисунке 7 показан макет внешнего вида формы места.



Рисунок 7 – Activity места

На рисунке 8 показан внешний вид формы места в приложении.

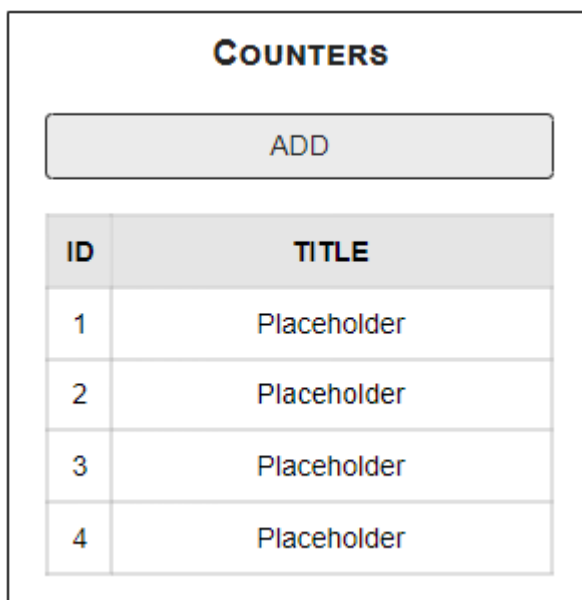


Рисунок 8 – Activity места в приложении

На экране присутствуют компоненты EditText для ввода названия места, а также кнопки для закрытия, сохранения и удаления места.

## 1.5. Список счетчиков

На рисунке 9 показан макет внешнего вида формы списка счетчиков.



COUNTERS	
ADD	
ID	TITLE
1	Placeholder
2	Placeholder
3	Placeholder
4	Placeholder

Рисунок 9 – Activity для списка счетчиков

На рисунке 10 показан внешний вид формы списка счетчиков в приложении.



Рисунок 10 – Activity для списка счетчиков в приложении

На экране присутствуют компонент ListView для отображения списка счетчиков и выбора счетчика, а также кнопки добавления нового счетчика

## 1.6. Счетчик

На рисунке 11 показан макет внешнего вида формы счетчика.



Рисунок 11 – Activity счетчика

На рисунке 12 показан внешний вид формы заметки в приложении.





Рисунок 12 – Activity счетчика в приложении

На экране присутствуют компоненты для редактирования счетчика, а также кнопки для закрытия, сохранения и удаления счетчика.

## 1.7. Список тарифов

На рисунке 13 показан макет внешнего вида формы списка тарифов.

RATES	
ADD	
ID	TITLE
1	Placeholder
2	Placeholder
3	Placeholder
4	Placeholder

Рисунок 13 – Activity для тарифов заметок

На рисунке 14 показан внешний вид формы списка заметок в приложении.



Рисунок 14 – Activity для списка тарифов в приложении

На экране присутствуют компонент ListView для отображения списка тарифов и выбора тарифа, а также кнопки добавления нового тарифа.

## 1.8. Тариф

На рисунке 15 показан макет внешнего вида формы тарифа.



Рисунок 15 – Activity тарифа

На рисунке 16 показан внешний вид формы тарифа в приложении.



Рисунок 16 – Activity тарифа в приложении

На экране присутствуют компоненты для редактирования тарифа, а также кнопки для закрытия, сохранения и удаления тарифа.

## 1.9. Список измерений

На рисунке 17 показан макет внешнего вида формы списка измерений.

MEASUREMENTS	
ADD	
ID	TITLE
1	Placeholder
2	Placeholder
3	Placeholder
4	Placeholder

Рисунок 17 – Activity для списка измерений

На рисунке 18 показан внешний вид формы списка измерений в приложении.



Рисунок 19 – Activity для списка измерений в приложении

На экране присутствуют компонент ListView для отображения списка измерений и выбора измерений, а также кнопки добавления нового измерения.

### **1.10. Измерение**

На рисунке 20 показан макет внешнего вида формы измерения.



Рисунок 20 – Activity измерения

На рисунке 21 показан внешний вид формы измерения в приложении.



Рисунок 21 – Activity измерения в приложении

На экране присутствуют компоненты для редактирования измерения, а также кнопки для закрытия, сохранения и удаления измерения.

## 2. Структура базы данных

База данных состоит из 3 таблиц:

1. Settings – сохраненные логин и пароль пользователя.
2. Session – сохраненная сессия.
3. APIEndPoint – сохраненная конечная точки.

ER-диаграмма базы данных представлена на рисунке 22.

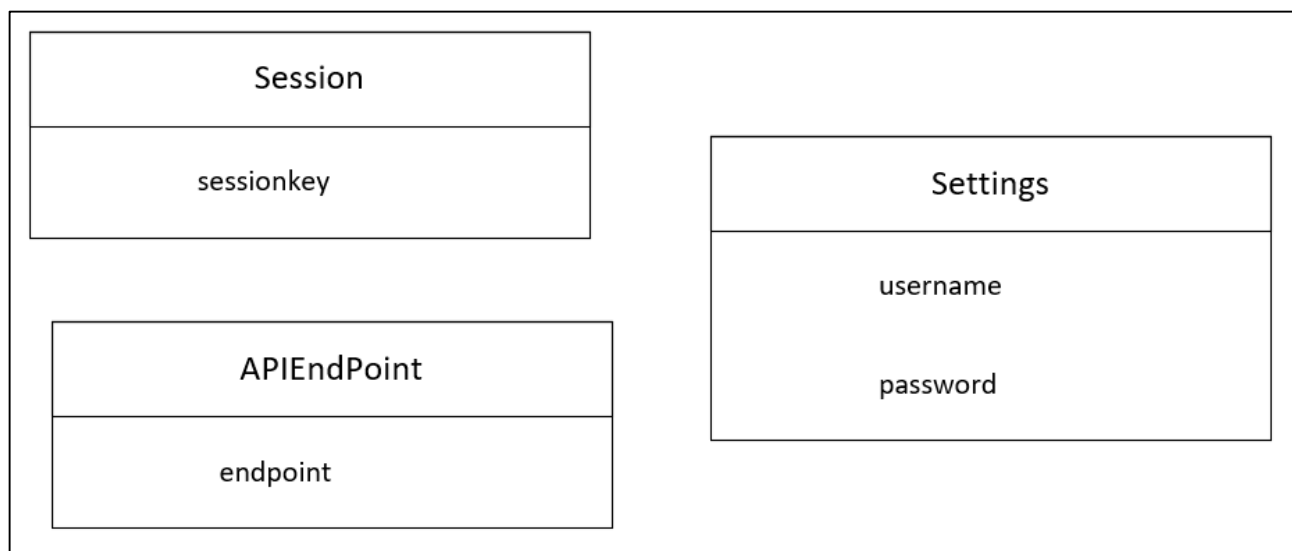


Рисунок 22 – ER-диаграмма базы данных

### 2.1. Таблица Settings

Содержит сведения о сохраненных учетных данных приложения. Таблица состоит из 2 столбцов:

1. username – логин пользователя.
2. password – пароль пользователь.

Подробное описание столбцов представлено на рисунке 23.

Settings		save auth data						
#	name	type	size	default	primary	foreign	unique	description
1	username	text	-		no	-	no	user login
2	password	text	-		no	-	no	user password

Рисунок 23 – Описание столбцов таблицы Settings

### 2.2. Таблица APIEndPoint

Содержит сведения о сохраненной конечной точке приложения. Таблица состоит из 1 столбца:

1. endpoint – логин пользователя.

Подробное описание столбцов представлено на рисунке 24.

<i>APIEndPoint</i>		<i>endpoint of api</i>						
#	name	type	size	default	primary	foreign	unique	description
1	endpoint	text	-		no	-	no	api endpoint

Рисунок 24 – Описание столбцов таблицы APIEndPoint

## 2.3. Таблица Session

Содержит сведения о сохраненном ключе сессии приложения. Таблица состоит из 1 столбца:

1. sessionkey – ключ сессии.

Подробное описание столбцов представлено на рисунке 25.

<i>Session</i>		<i>saved session key</i>						
#	name	type	size	default	primary	foreign	unique	description
1	sessionkey	text	-		no	-	no	key of session

Рисунок 25 – Описание столбцов таблицы Settings

### 3. Описание протокола взаимодействия

Для взаимодействия с базой данных мобильное приложение использует API на основе хранимых процедур СУБД Postgres и сервера PostgREST, который организует вызов процедур по протоколу HTTP и обмен данными в формате JSON.

Всего доступно 21 функция, список которых показан на рисунке 26:

1. `sign_in` – авторизация пользователя.
2. `sign_out` – выход из сессии.
3. `register_account` – регистрация пользователя.
4. `update_password` – изменение пароля пользователя.
5. `delete_account` – удаление пользователя.
6. `add_location` – добавление места.
7. `get_locations` – получение списка мест пользователя.
8. `update_location` – обновление места.
9. `delete_location` – удаление места.
10. `add_counter` – добавление счетчика.
11. `get_counters` – получение списка счетчиков пользователя.
12. `update_counter` – обновление счетчика.
13. `delete_counter` – удаление счетчика.
14. `add_rate` – добавление тарифа.
15. `get_rates` – получение списка тарифов пользователя.
16. `update_rate` – обновление тарифа.
17. `delete_rate` – удаление тарифа.
18. `add_measurement` – добавление измерения.
19. `get_measurements` – получение списка измерений пользователя.
20. `update_measurement` – обновление измерения.
21. `delete_measurement` – удаление измерения.

(rpc) delete_measurement	^
POST /rpc/delete_measurement remove existing measurement sample by id	v
(rpc) get_counters	^
POST /rpc/get_counters list all existing counters for given location	v
(rpc) get_locations	^
POST /rpc/get_locations list all existing locations for current account	v
(rpc) delete_counter	^
POST /rpc/delete_counter delete given counter by id	v
(rpc) get_rates	^
POST /rpc/get_rates list all existing rates for given counter	v
(rpc) update_password	^
POST /rpc/update_password change currently logged in account password	v
(rpc) delete_account	^
POST /rpc/delete_account delete existing account by active session token	v
(rpc) delete_rate	^
POST /rpc/delete_rate remove existing counter rate by id	v
(rpc) get_measurements	^
POST /rpc/get_measurements list all existing measurements for given counter	v
(rpc) delete_location	^
POST /rpc/delete_location delete given location by id	v
(rpc) add_location	^
POST /rpc/add_location add new location to account	v
(rpc) update_counter	^
POST /rpc/update_counter update existing counter parameters	v
(rpc) add_counter	^
POST /rpc/add_counter add new counter to specified location	v
(rpc) add_measurement	^
POST /rpc/add_measurement add new data sample for given counter	v
(rpc) add_rate	^
POST /rpc/add_rate add new rate value for given counter	v
(rpc) register_account	^
POST /rpc/register_account create new user account with given name and password	v
(rpc) sign_out	^
POST /rpc/sign_out dispose an active session token	v
(rpc) update_location	^
POST /rpc/update_location update existing location name	v
(rpc) sign_in	^
POST /rpc/sign_in authenticate user with name and password, start a new session returning token	v

Рисунок 26 – Список хранимых процедур

## 3.1. Контроль сессий

Данный набор функций позволяет управлять сессией пользователя.

### 3.1.1. Функция `sign_in`

Позволяет пройти авторизацию с имеющимся логином и паролем, возвращает ключ сессии.

*Входные параметры:*

- `name1` – логин пользователя.
- `password1` – пароль пользователя.

*Выходные параметры:*

- ключ сессии в формате строки.

Пример вызова функции показаны на рисунках 27 и 28.

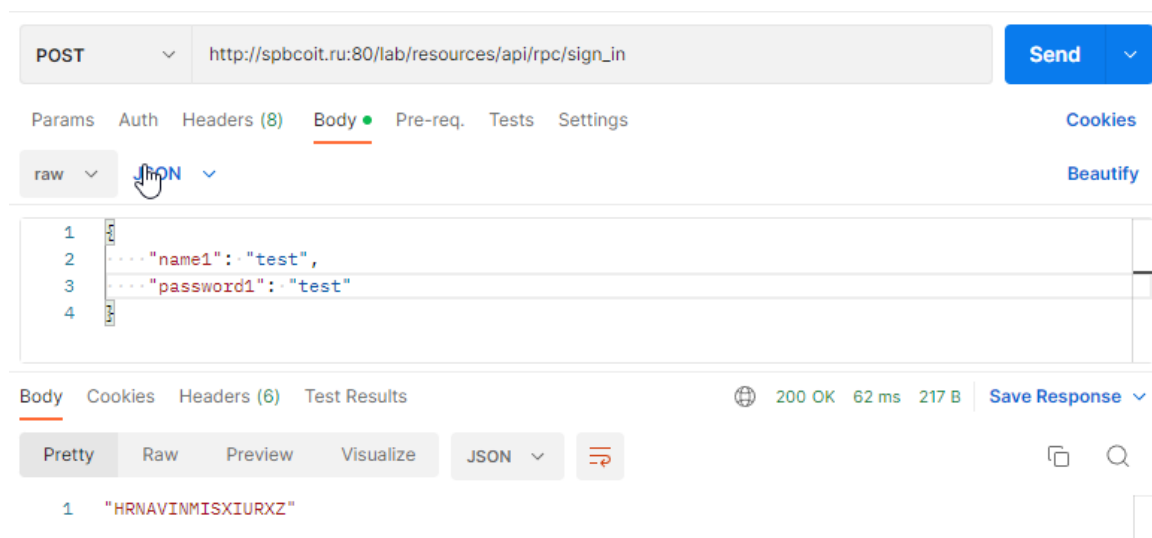


Рисунок 27 – Пример работы функции `sign_in`

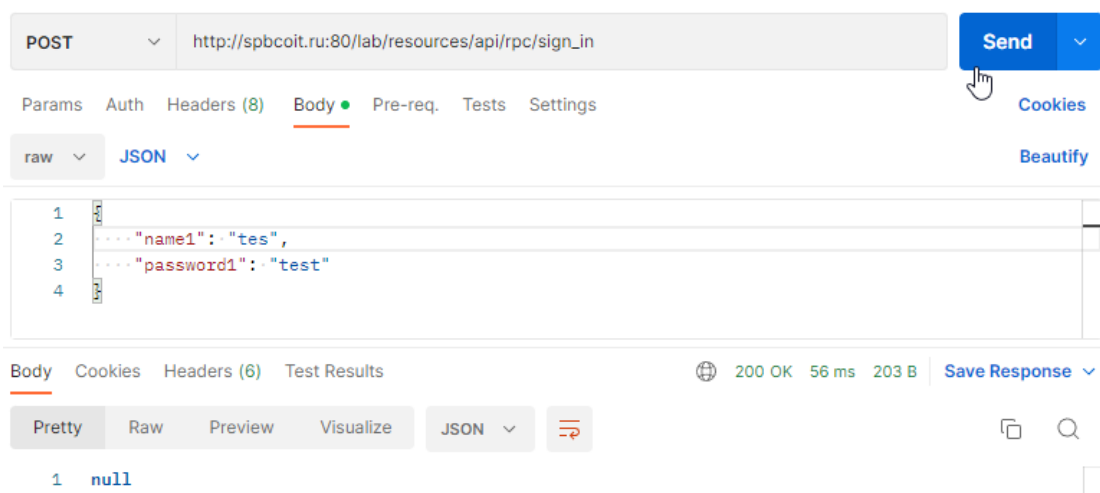


Рисунок 28 – Вызов функции `sign_in` с неверными данными



### 3.1.2. Функция sign\_out

Позволяет закрыть сессию по ключу.

*Входные параметры:*

— key1 – ключ сессии.

*Выходные параметры:*

— true или false в зависимости от правильности ключа сессии.

Пример вызова функции показаны на рисунках 29 и 30.

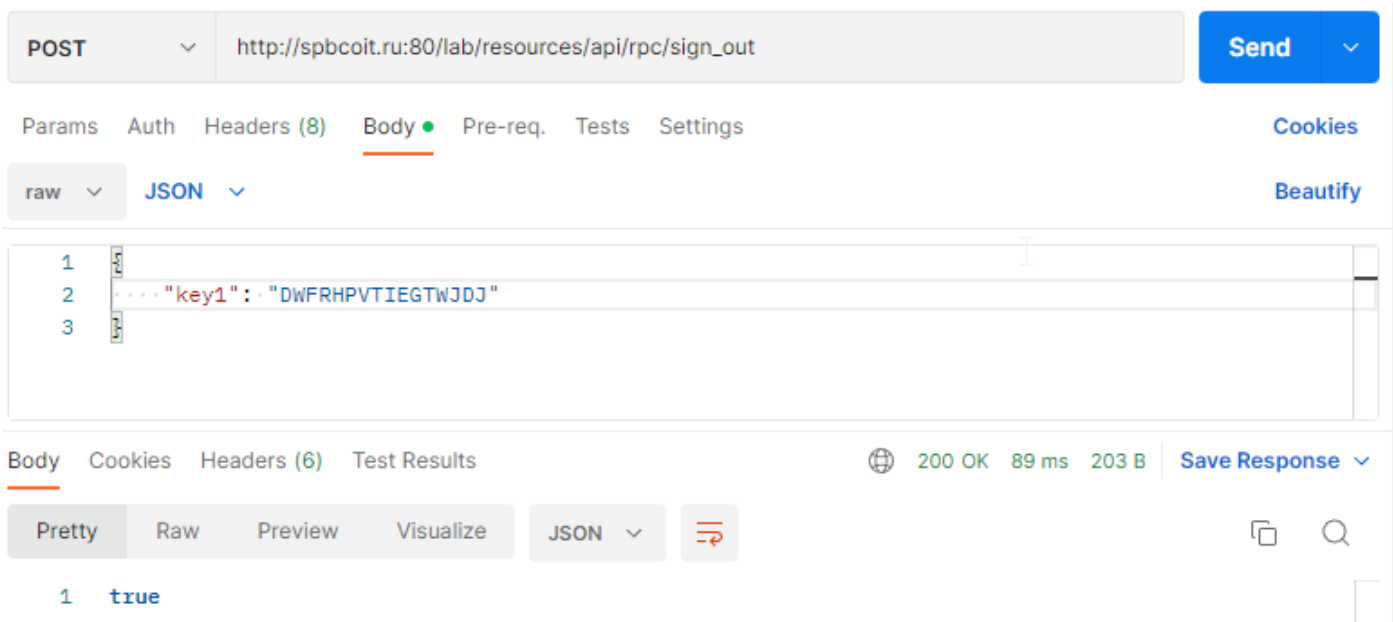


Рисунок 29 – Пример работы функции sign\_out

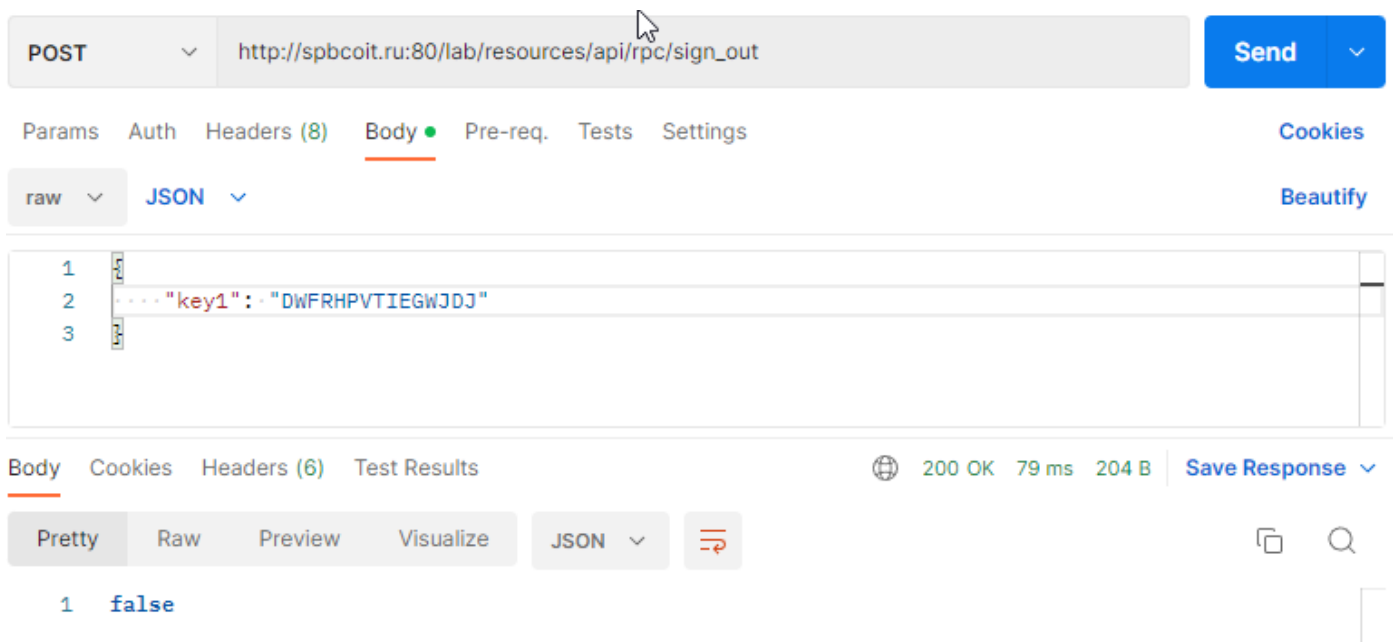


Рисунок 30 – Вызов функции sign\_out с неверными данными

### 3.1.3. Функция register\_account

Позволяет зарегистрировать новую учетную запись по ранее не существующему логину и паролю.

*Входные параметры:*

- name1 – несуществующее ранее в системе имя пользователя.
- password1 – пароль пользователя.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 31 и 32.

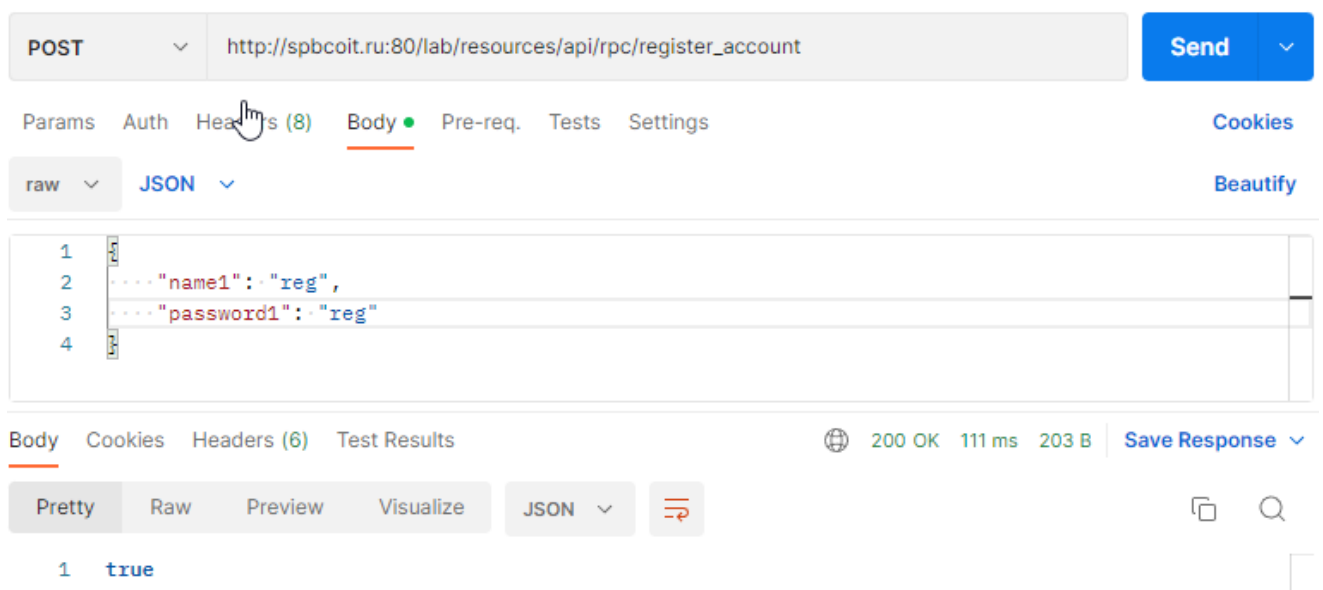


Рисунок 31 – Пример работы функции register\_account

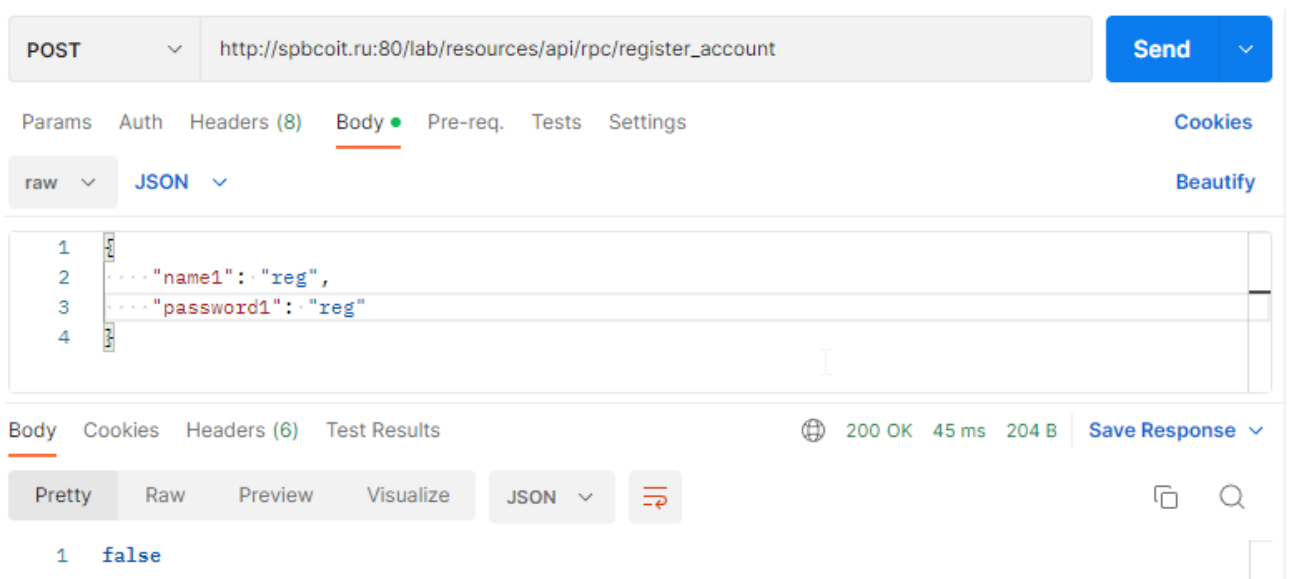


Рисунок 32 – Вызов функции register\_account с неверными данными

### 3.1.4. Функция update\_password

Позволяет изменить пароль пользователя с указанием ключа сессии.

*Входные параметры:*

- key1 – ключ сессии.
- password1 – новый пароль.

*Выходные параметры:*

- true или false в зависимости от правильности ключа сессии.

Пример вызова функции показаны на рисунках 33 и 34.

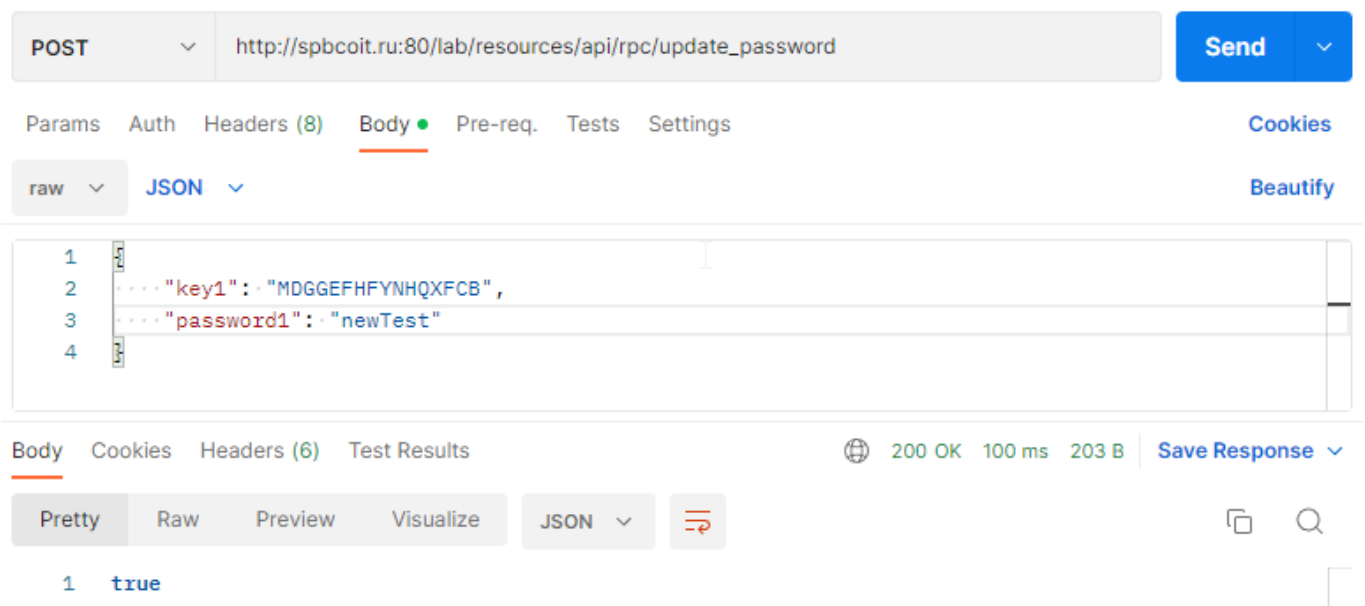


Рисунок 33 – Пример работы функции update\_password

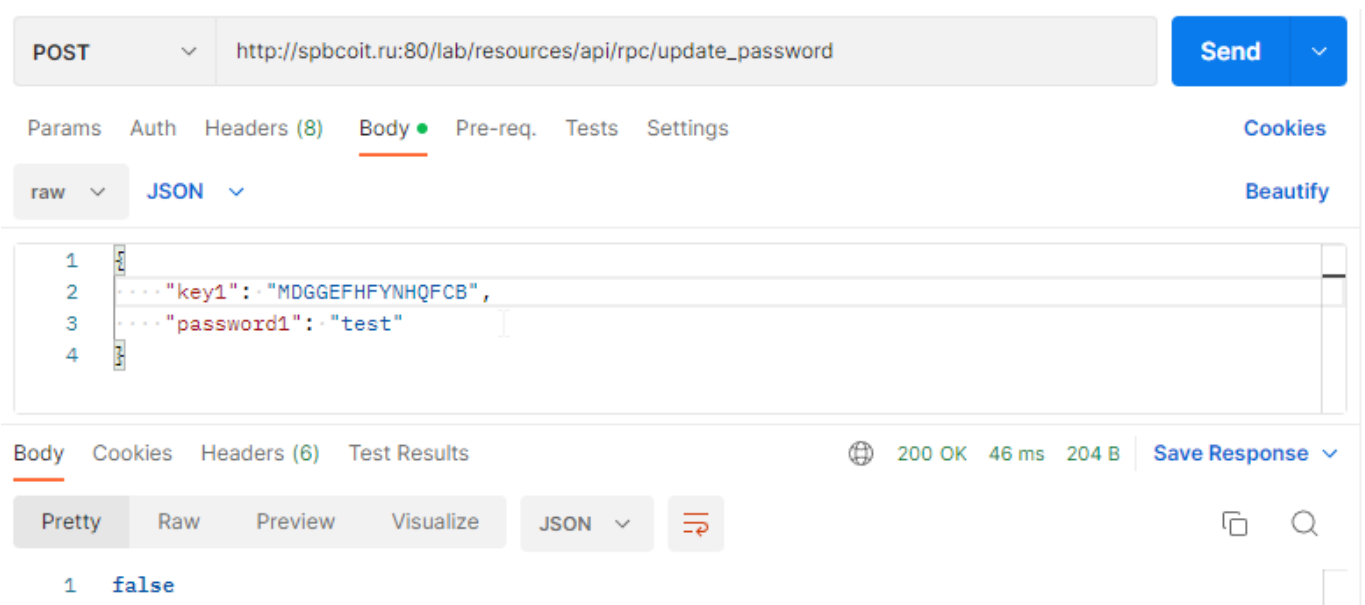


Рисунок 34 – Вызов функции update\_password с неверными данными

### 3.1.5. Функция delete\_account

Позволяет удалить пользователя по ключу сессии.

*Входные параметры:*

— key1 – ключ сессии.

*Выходные параметры:*

— true или false в зависимости от правильности ключа сессии.

Пример вызова функции показаны на рисунках 35 и 36.



Рисунок 35 – Пример работы функции delete\_account

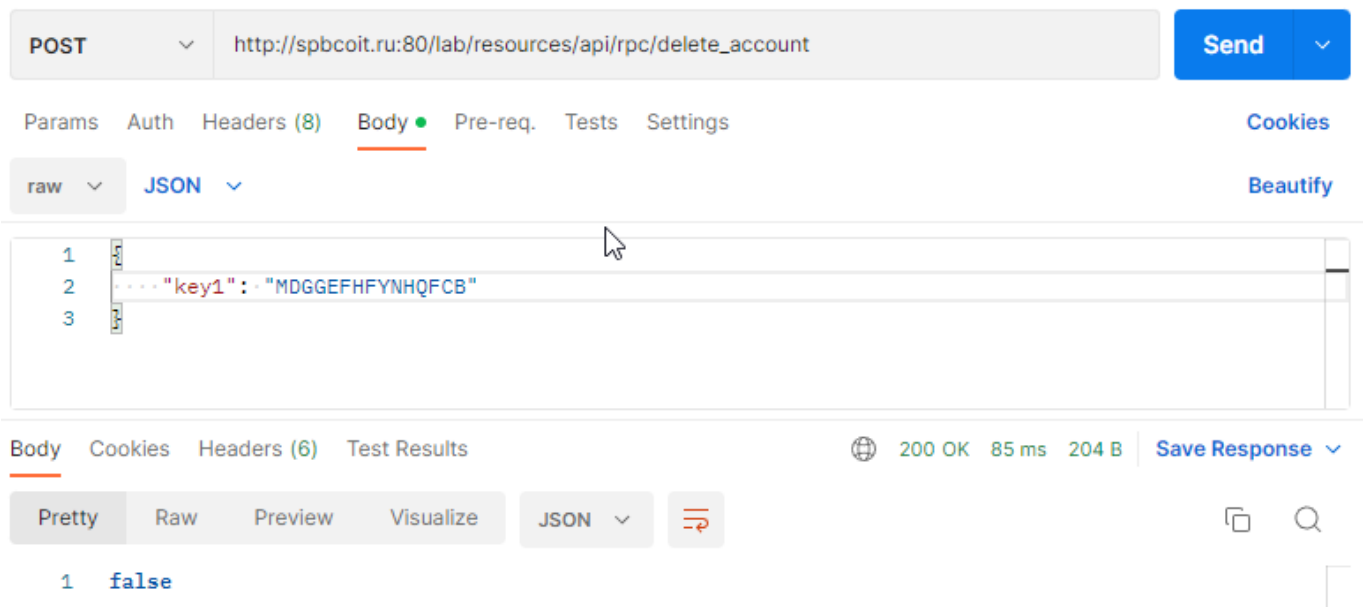


Рисунок 36 – Вызов функции delete\_account с неверными данными

## 3.2. Управление местами

Данный набор функций позволяет управлять местами пользователя.

### 3.2.1. Функция add\_location

Позволяет добавить новое место с заданным именем для пользователя.

*Входные параметры:*

— key1 – ключ сессии.

— name1 – имя нового места.

*Выходные параметры:*

— true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 37 и 38.

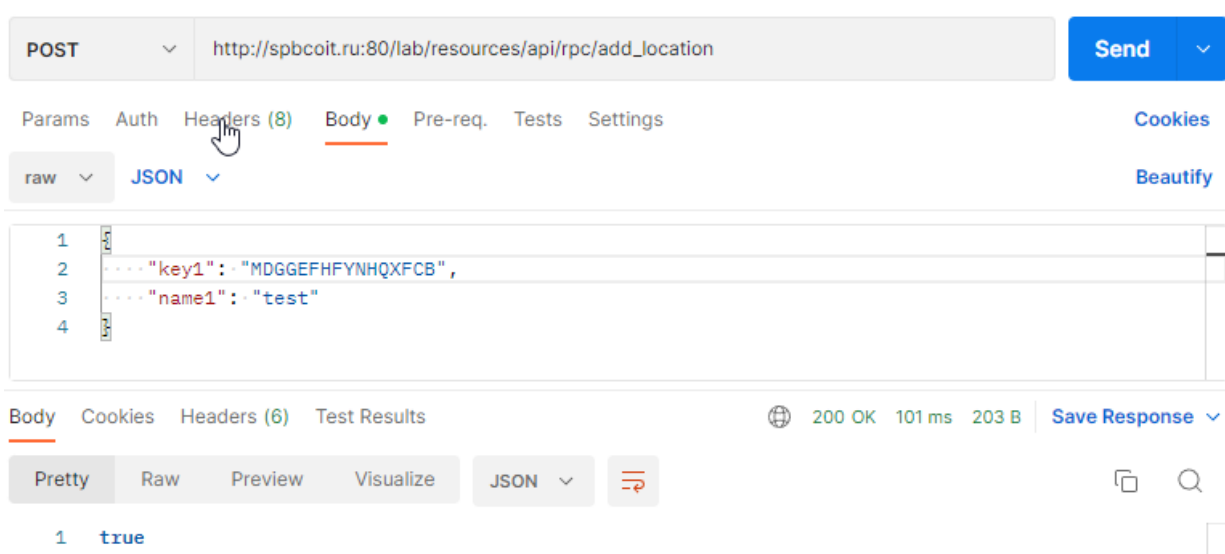


Рисунок 37 – Пример работы функции add\_location

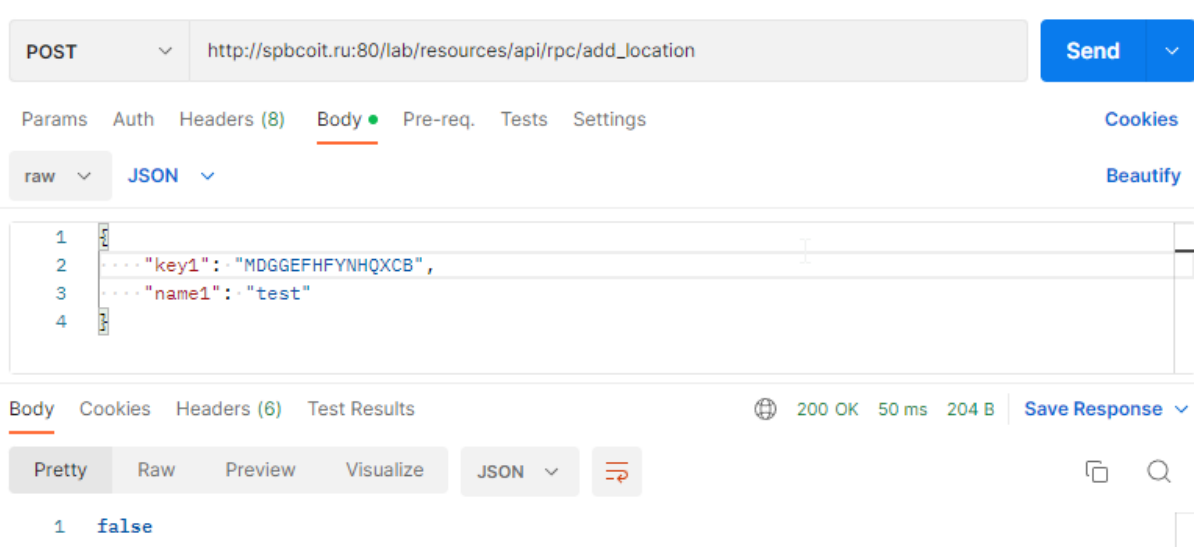


Рисунок 38 – Вызов функции add\_location с неверными данными

### 3.2.2. Функция get\_locations

Позволяет получить список мест пользователя по ключу сессии.

*Входные параметры:*

— key1 – ключ сессии.

*Выходные параметрами будет массив данных со следующими атрибутами:*

— id2 – уникальный идентификатор места.

— name2 – название места.

Пример вызова функции показаны на рисунках 39 и 40.

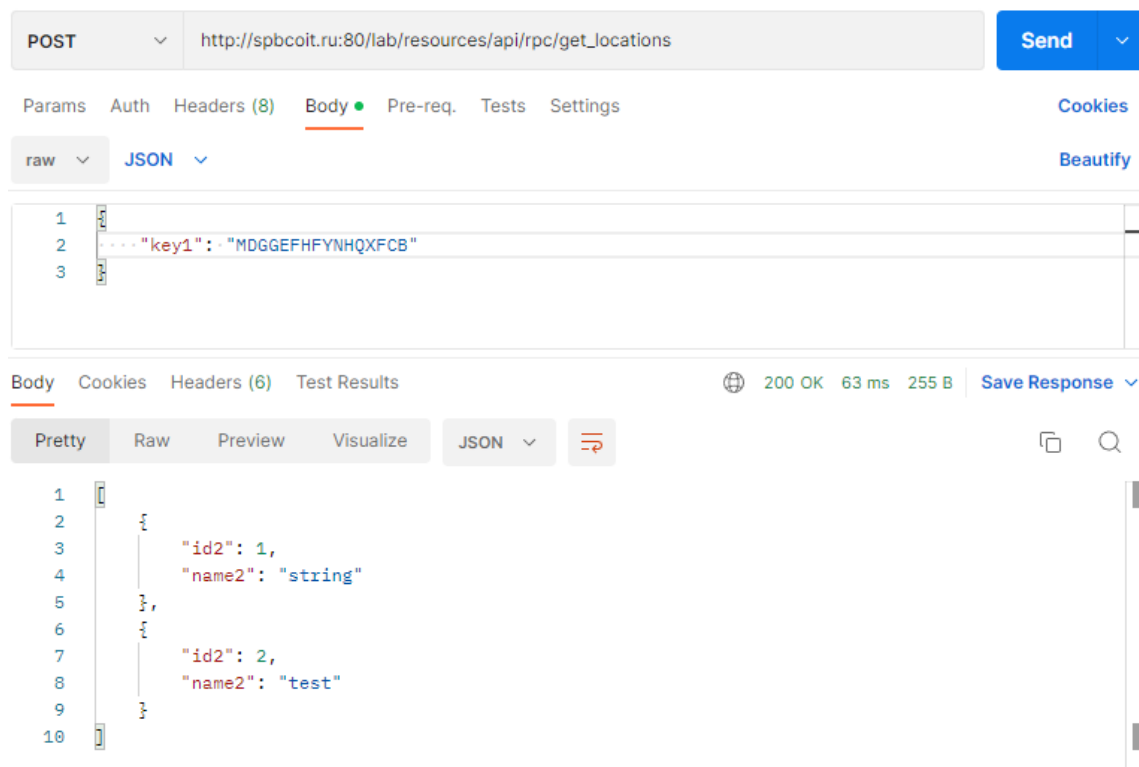


Рисунок 39 – Пример работы функции `get_locations`

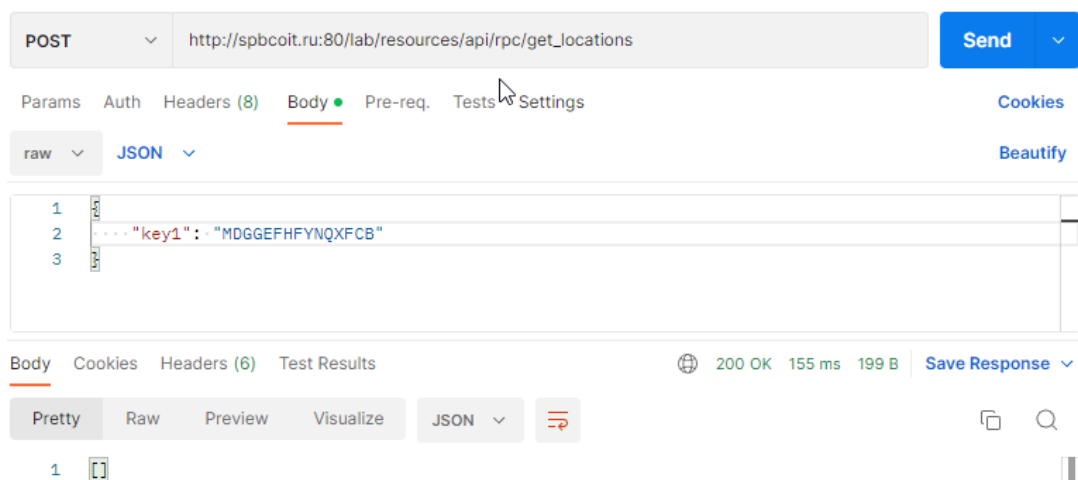


Рисунок 40 – Вызов функции `get_locations` с неверными данными

### 3.2.3. Функция `update_location`

Позволяет обновить свойства места с указанием его идентификатора, ключа сессии и новых данных.

*Входные параметры:*

- `id1` – идентификатор места.
- `key1` – ключ сессии.

— name1 – новое имя места.

*Выходные параметры:*

— true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 41 и 42.

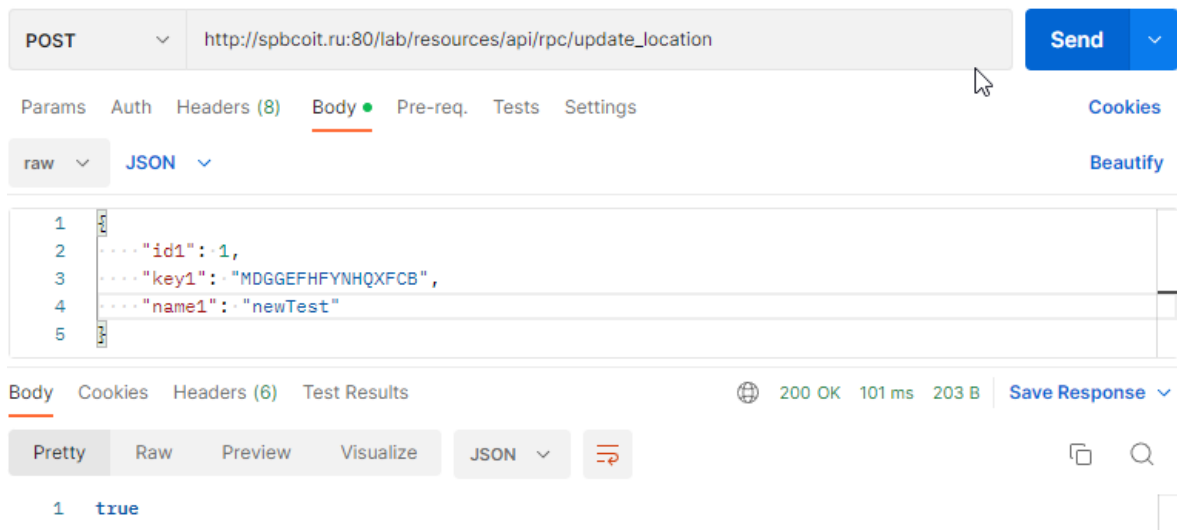


Рисунок 41 – Пример работы функции update\_location

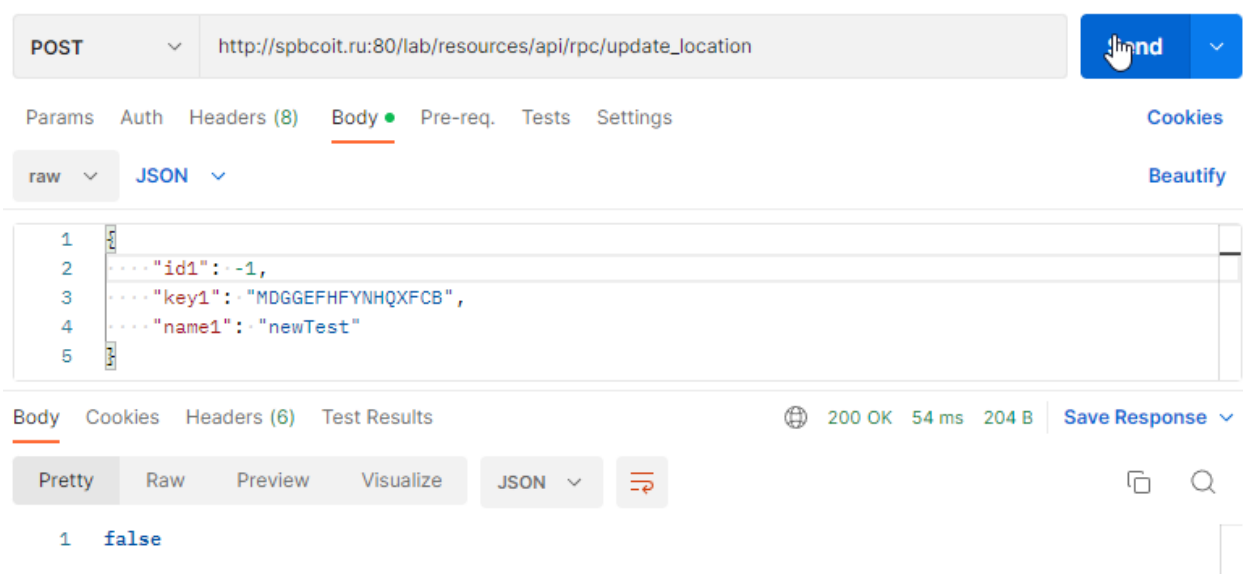


Рисунок 42 – Вызов функции update\_location с неверными данными

### 3.2.4. Функция delete\_location

Позволяет удалить место с указанием его идентификатора и ключа сессии.

*Входные параметры:*

— id1 – идентификатор места.

— key1 – ключ сессии.

*Выходные параметры:*

— true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 43 и 44.

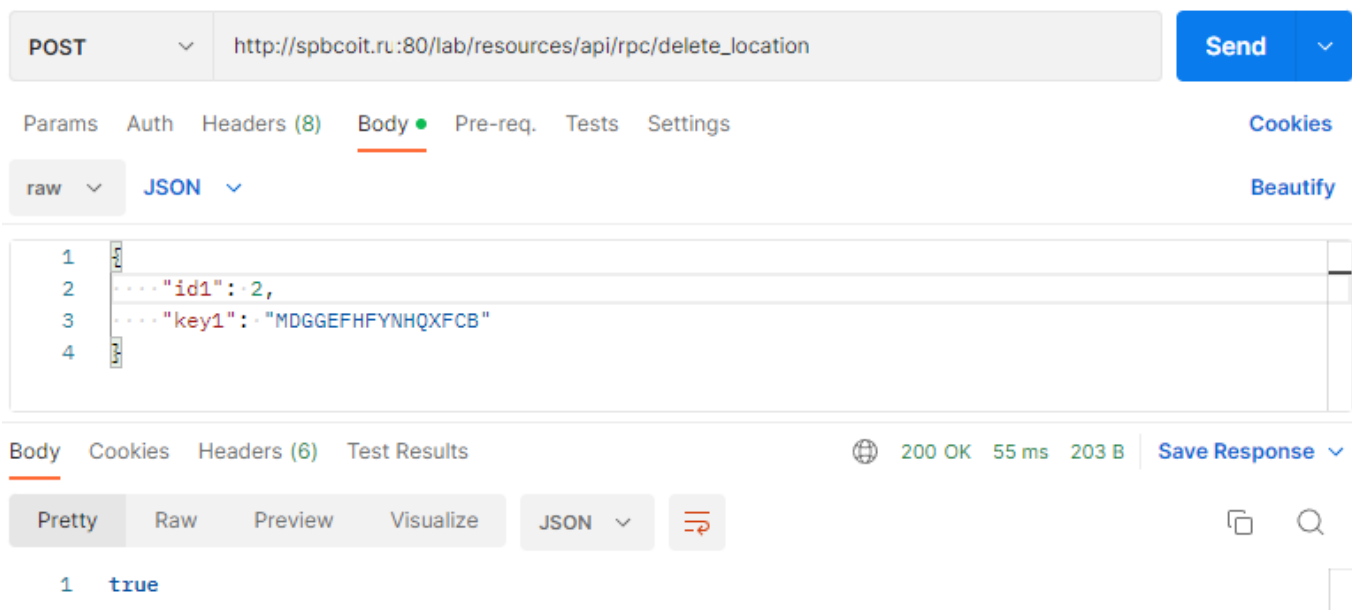


Рисунок 43 – Пример работы функции delete\_location

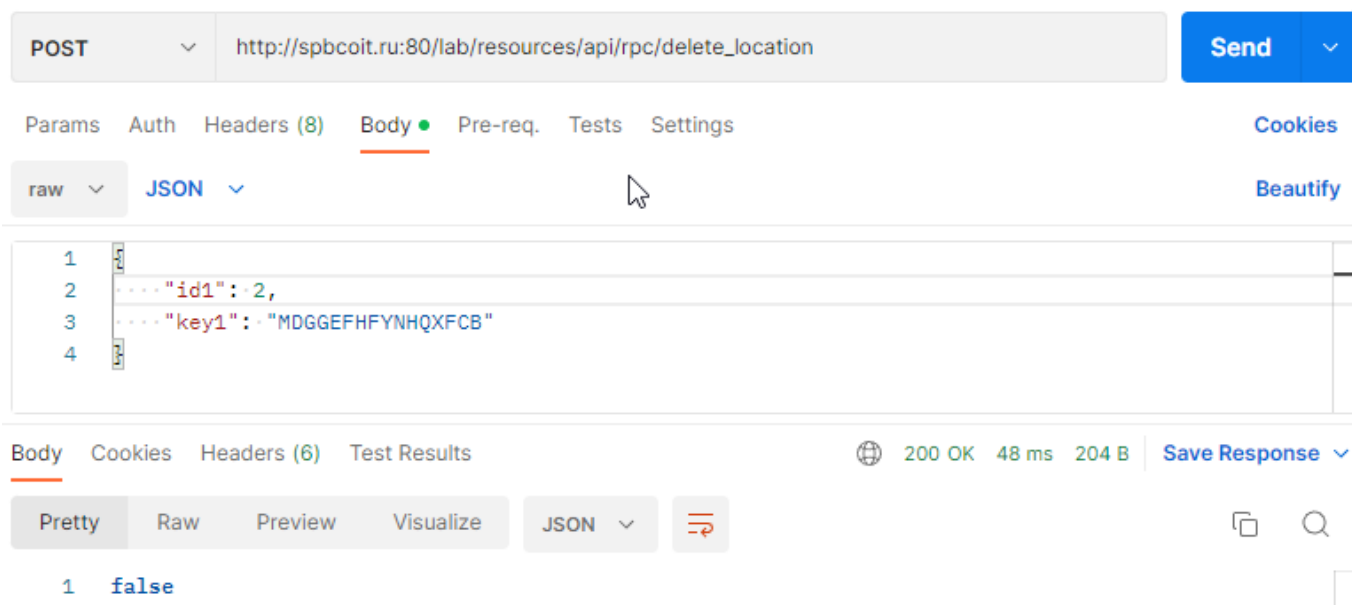


Рисунок 44 – Вызов функции delete\_location с неверными данными

### 3.3. Управление счетчиками

Данный набор функций позволяет управлять счетчиками пользователя.

#### 3.3.1. Функция add\_counter

Позволяет добавить новый счетчик с заданными свойствами для места пользователя.



*Входные параметры:*

- icon1 – иконка счетчика.
- key1 – ключ сессии.
- location1 – место, к которому будет привязан счетчик.
- name1 – имя счетчика.
- unit1 – единицы измерения счетчика

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 45 и 46.

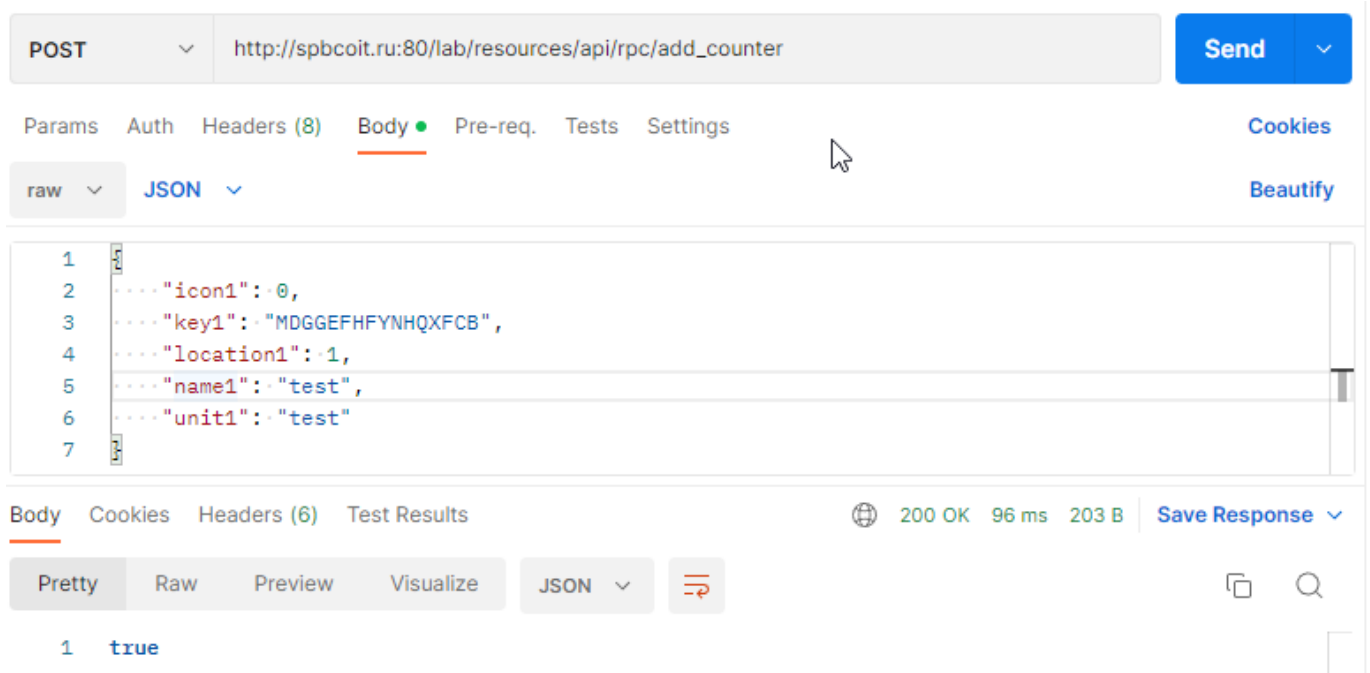


Рисунок 45 – Пример работы функции add\_counter

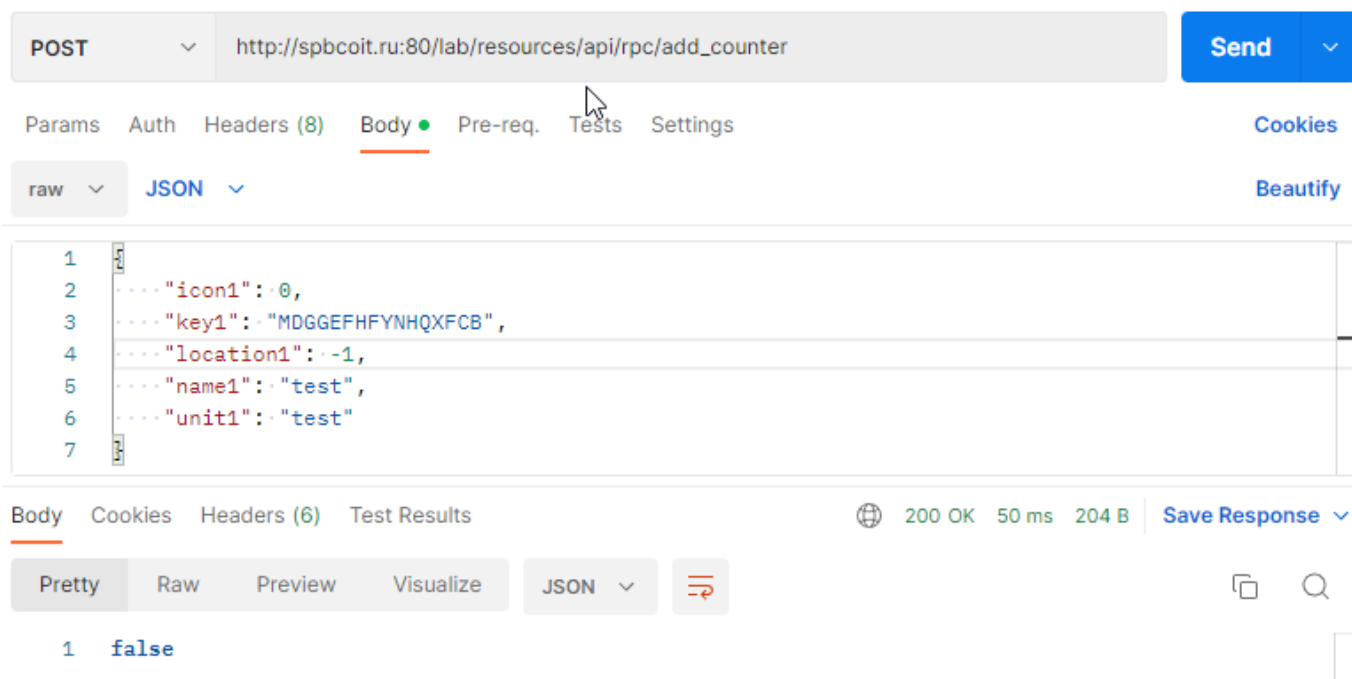


Рисунок 46 – Вызов функции add\_counter с неверными данными

### 3.3.2. Функция get\_counters

Позволяет получить список счетчиков для места пользователя по ключу сессии.

*Входные параметры:*

- key1 – ключ сессии.
- location1 – место счетчиков.

*Выходные параметрами будет массив данных со следующими атрибутами:*

- id2 – уникальный идентификатор счетчика.
- location2 – место счетчика.
- name2 – название счетчика.
- unit2 – единицы измерения счетчика
- icon2 – иконка счетчика

Пример вызова функции показаны на рисунках 47 и 48.

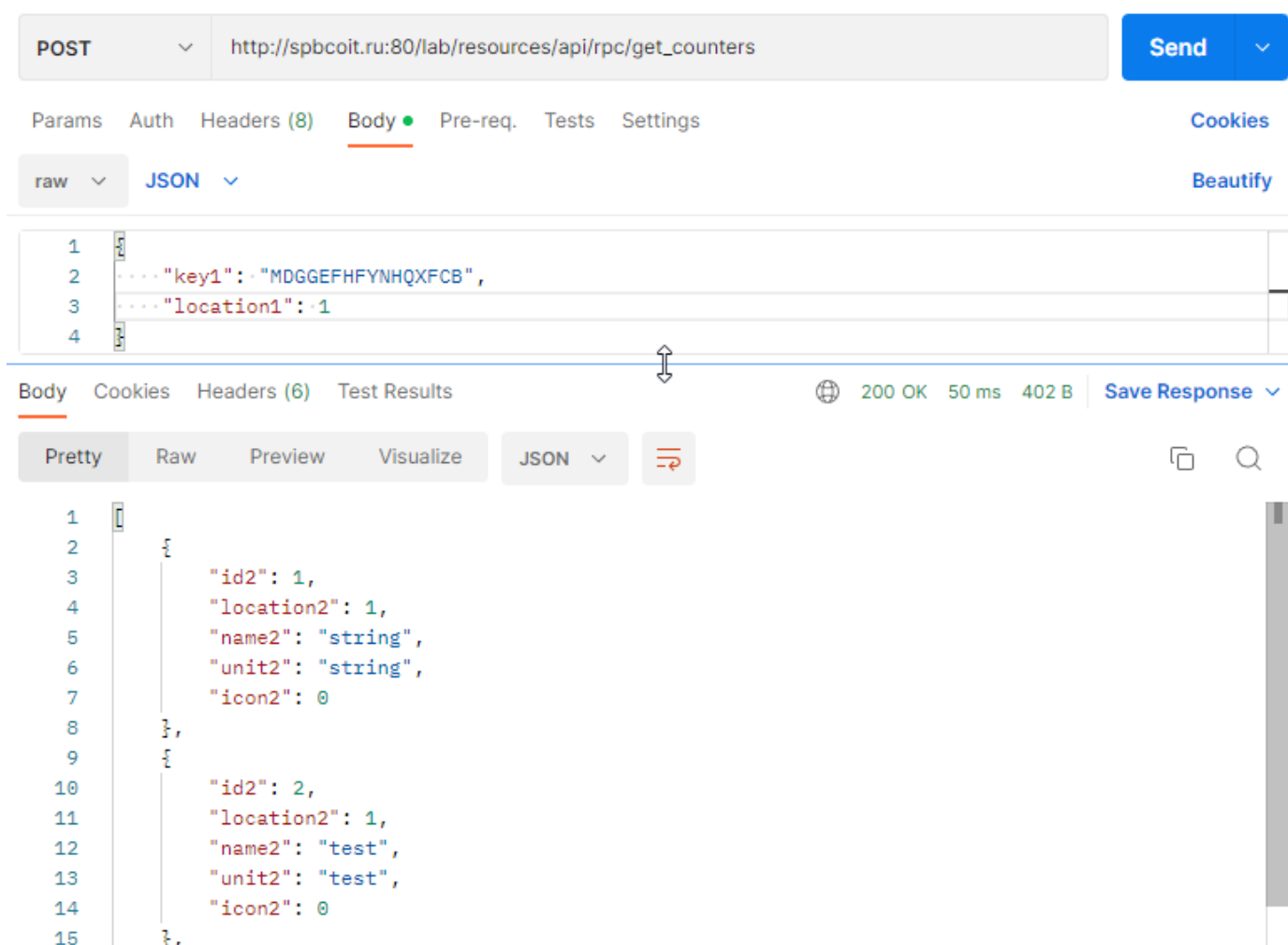


Рисунок 47 – Пример работы функции `get_counters`

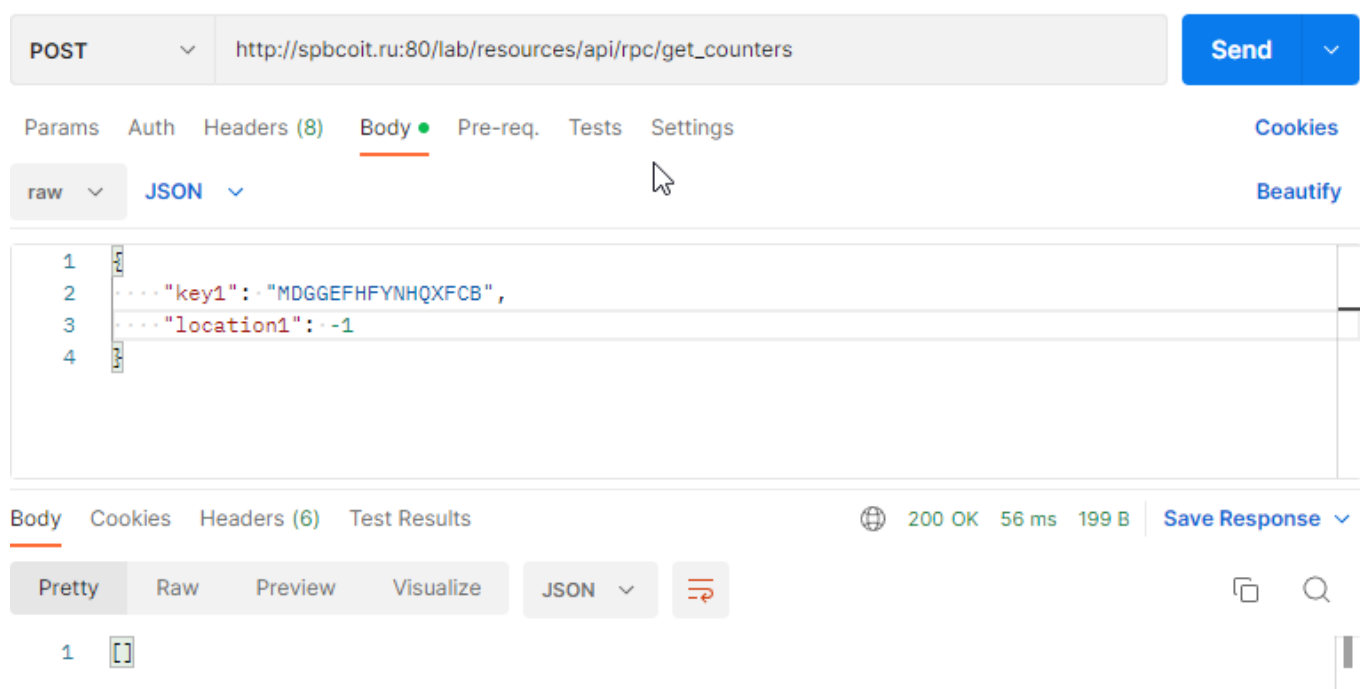


Рисунок 48 – Вызов функции `get_counters` с неверными данными

### 3.3.3. Функция update\_counter

Позволяет обновить свойства счетчика с указанием его идентификатора, ключа сессии и новых данных.

*Входные параметры:*

- counter1 – идентификатор счетчика.
- icon1 – иконка счетчика.
- key1 – ключ сессии.
- location1 – идентификатор места, к которому будет привязан счетчик.
- name1 – новое названия счетчика.
- unit1 – новые единицы измерения счетчика.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 49 и 50.

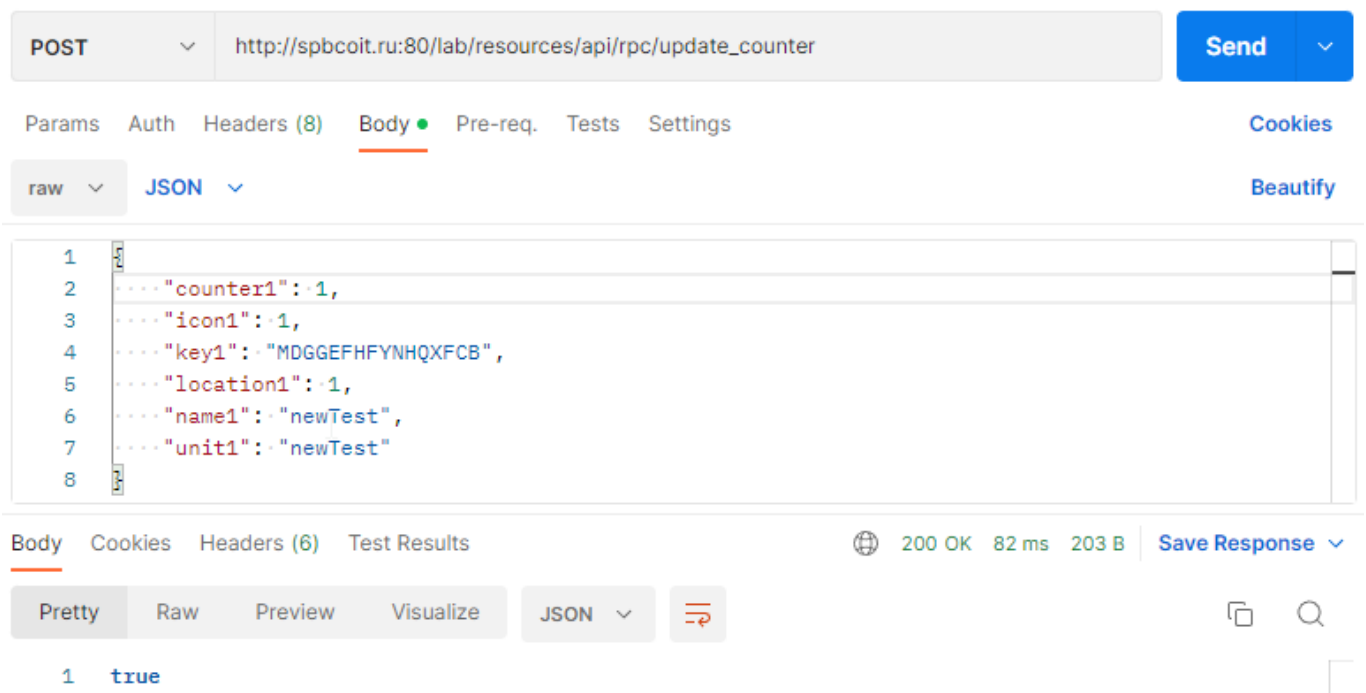


Рисунок 49 – Пример работы функции update\_counter

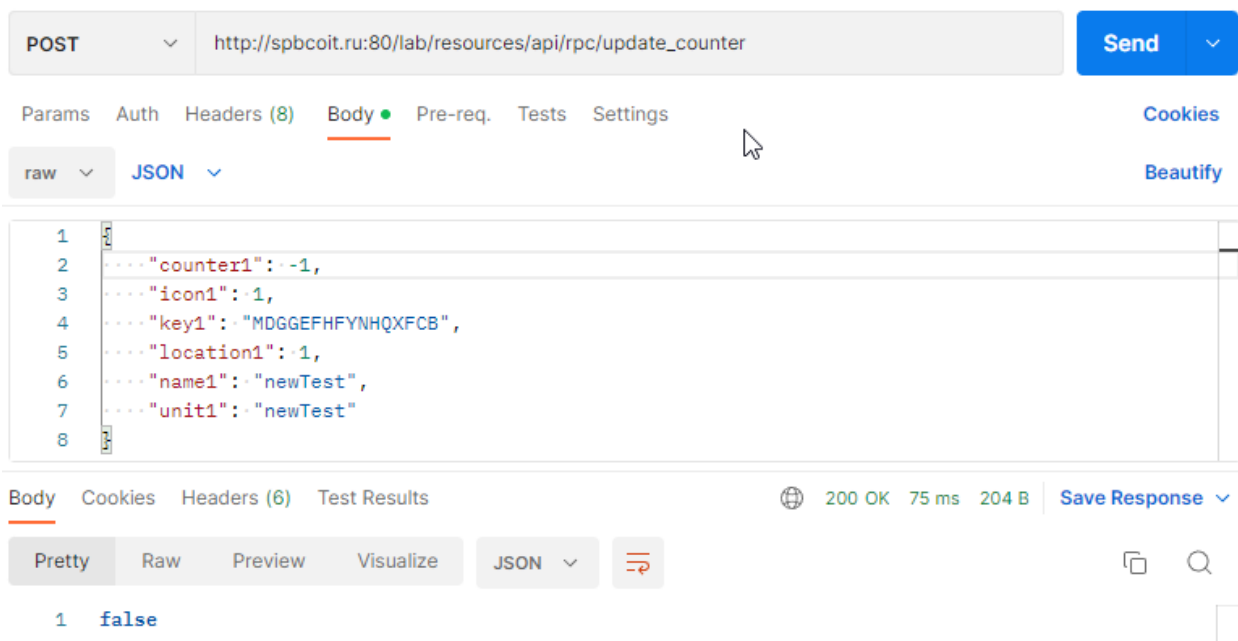


Рисунок 50 – Вызов функции update\_counter с неверными данными

### 3.3.4. Функция delete\_counter

Позволяет удалить счетчик с указанием его идентификатора и ключа сессии.

*Входные параметры:*

- counter1 – идентификатор счетчика.
- key1 – ключ сессии.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 51 и 52.

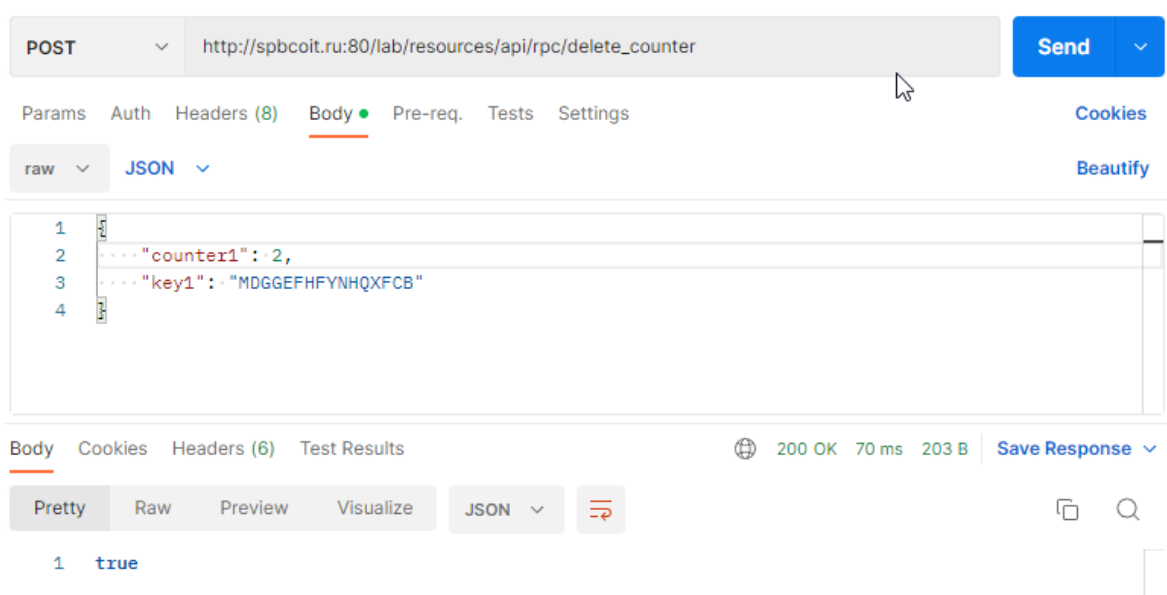


Рисунок 51 – Пример работы функции delete\_counter

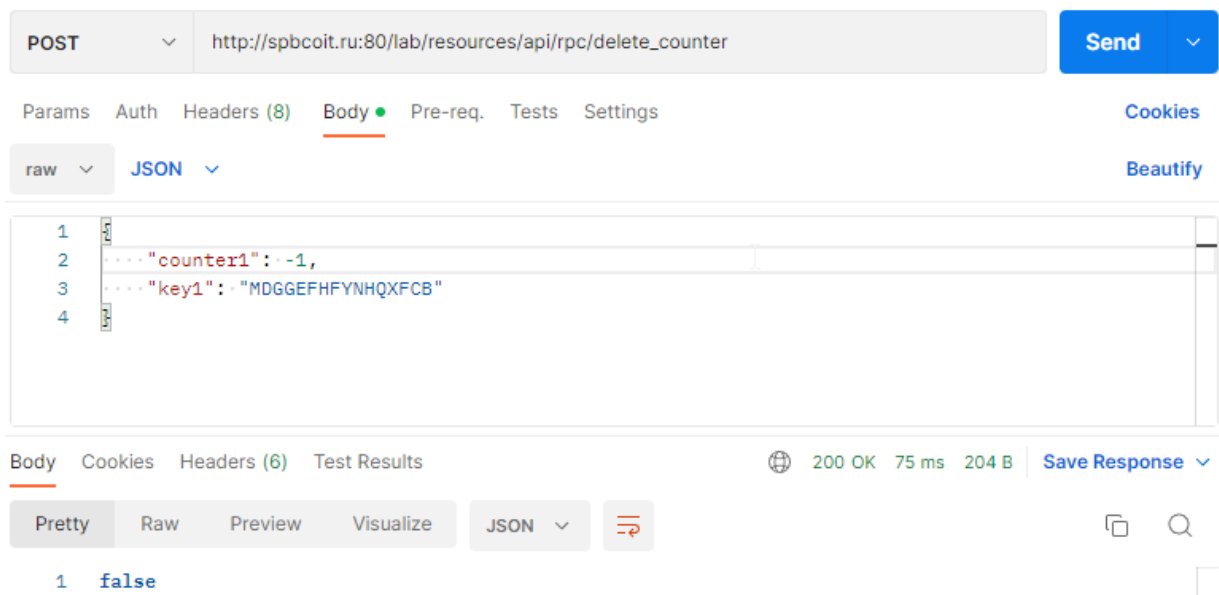


Рисунок 52 – Вызов функции delete\_counter с неверными данными

### 3.4. Управление тарифами

Данный набор функций позволяет управлять тарифами пользователя.

#### 3.4.1. Функция add\_rate

Позволяет добавить новый тариф с заданными свойствами для счетчика пользователя.

*Входные параметры:*

- counter1 – идентификатор счетчика, к которому будет привязан тариф.
- key1 – ключ сессии.
- ts1 – время вычисления тарифа.
- value1 – значение за единицу времени тарифа.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 53 и 54.

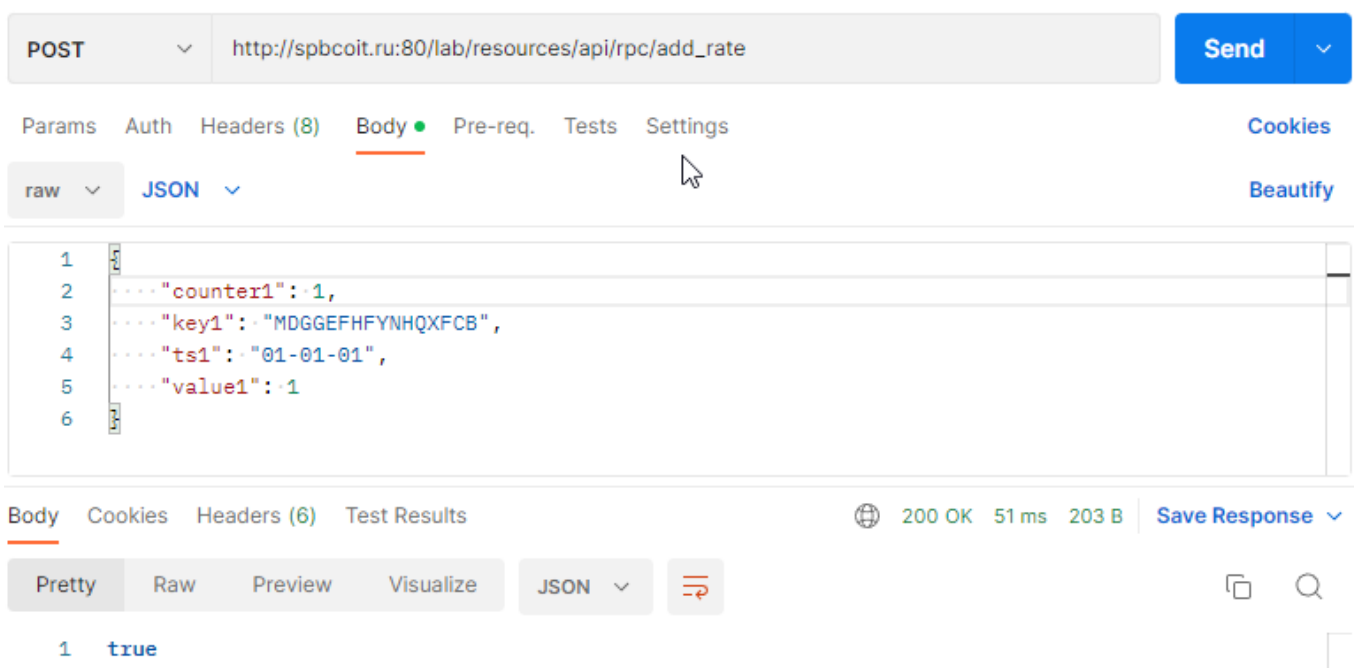


Рисунок 53 – Пример работы функции add\_rate

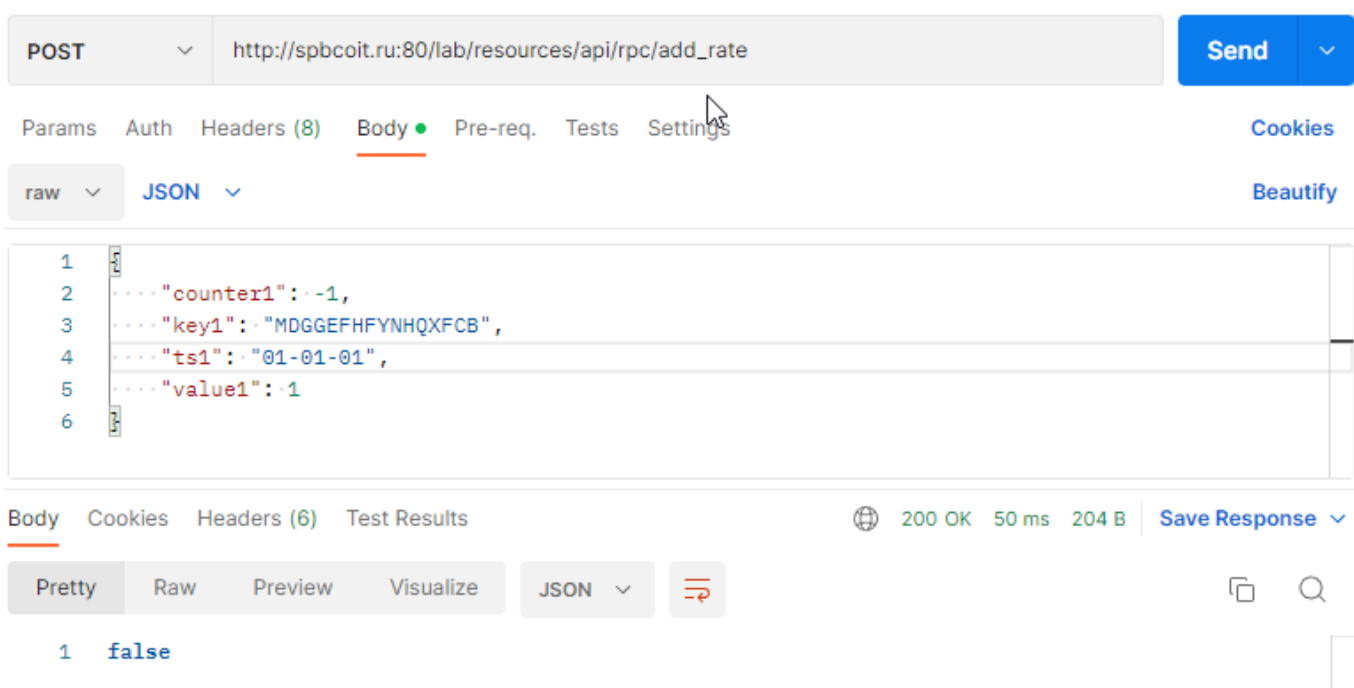


Рисунок 54 – Вызов функции add\_rate с неверными данными

### 3.4.2. Функция get\_rates

Позволяет получить список тарифов для счетчика пользователя по ключу сессии.

*Входные параметры:*

- counter1 – счетчик, к которому привязаны тарифы.
- key1 – ключ сессии.

*Выходные параметрами будет массив данных со следующими атрибутами:*

- id2 – уникальный идентификатор счетчика.
- counter2 – счетчик, к которому привязан тариф.
- ts2 – время измерения счетчика.
- value2 – значение за единицу времени счетчика.

Пример вызова функции показаны на рисунках 55 и 56.



Рисунок 55 – Пример работы функции get\_rates



Рисунок 56 – Вызов функции get\_rates с неверными данными

### **3.4.3. Функция update\_rate**

Позволяет обновить свойства тарифа с указанием его идентификатора, ключа сессии и новых данных.

*Входные параметры:*

- Pattern – Pattern.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 57 и 58.



Рисунок 57 – Пример работы функции update\_rate





Рисунок 58 – Вызов функции update\_rate с неверными данными

#### 3.4.4. Функция delete\_rate

Позволяет удалить тариф с указанием его идентификатора и ключа сессии.

*Входные параметры:*

- key1 – ключ сессии.
- rate1 – идентификатор тарифа.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 59 и 60.



Рисунок 59 – Пример работы функции delete\_rate



Рисунок 60 – Вызов функции delete\_rate с неверными данными

### 3.5. Управление измерениями

Данный набор функций позволяет управлять измерениями пользователя.

#### 3.5.1. Функция add\_measurement

Позволяет добавить новое измерения с заданными свойствами для счетчика пользователя.

*Входные параметры:*

- counter1 – идентификатор счетчика.
- image1 – изображение измерения.

- key1 – ключ сессии.
- ts1 – дата и время измерения.
- value1 – значение измерения.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 61 и 62.

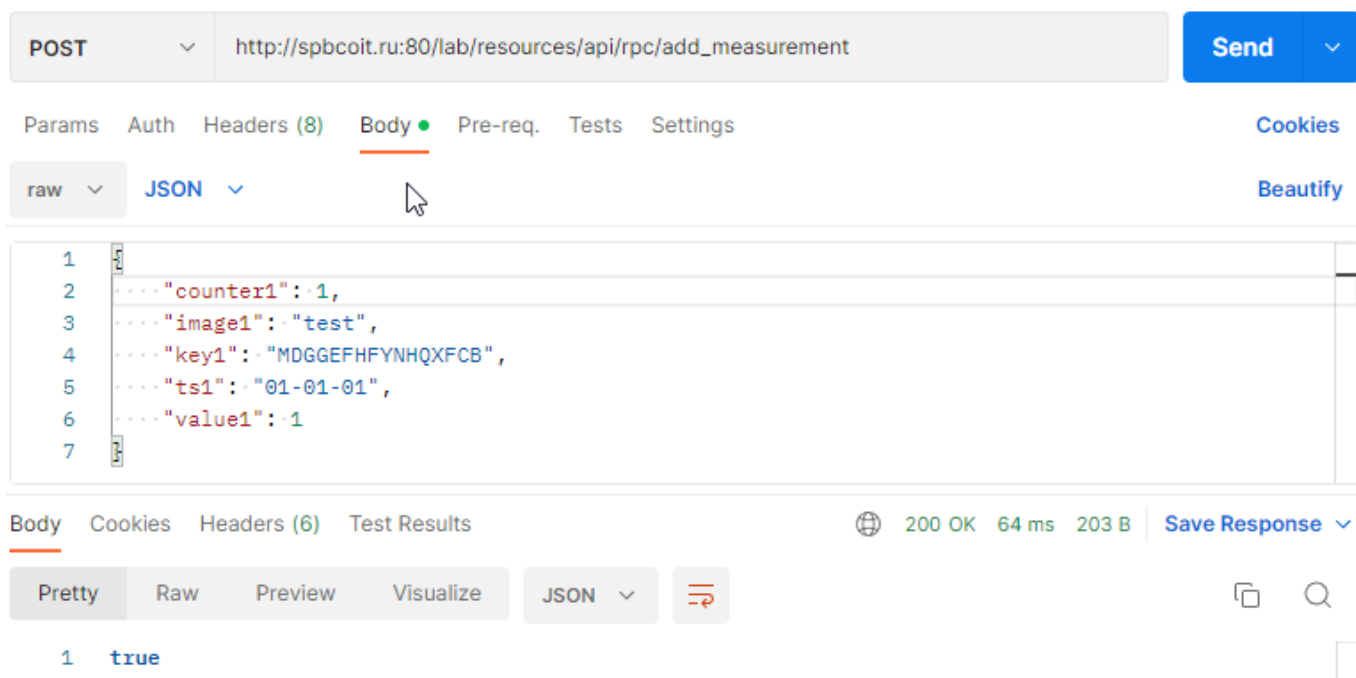


Рисунок 61 – Пример работы функции add\_measurement

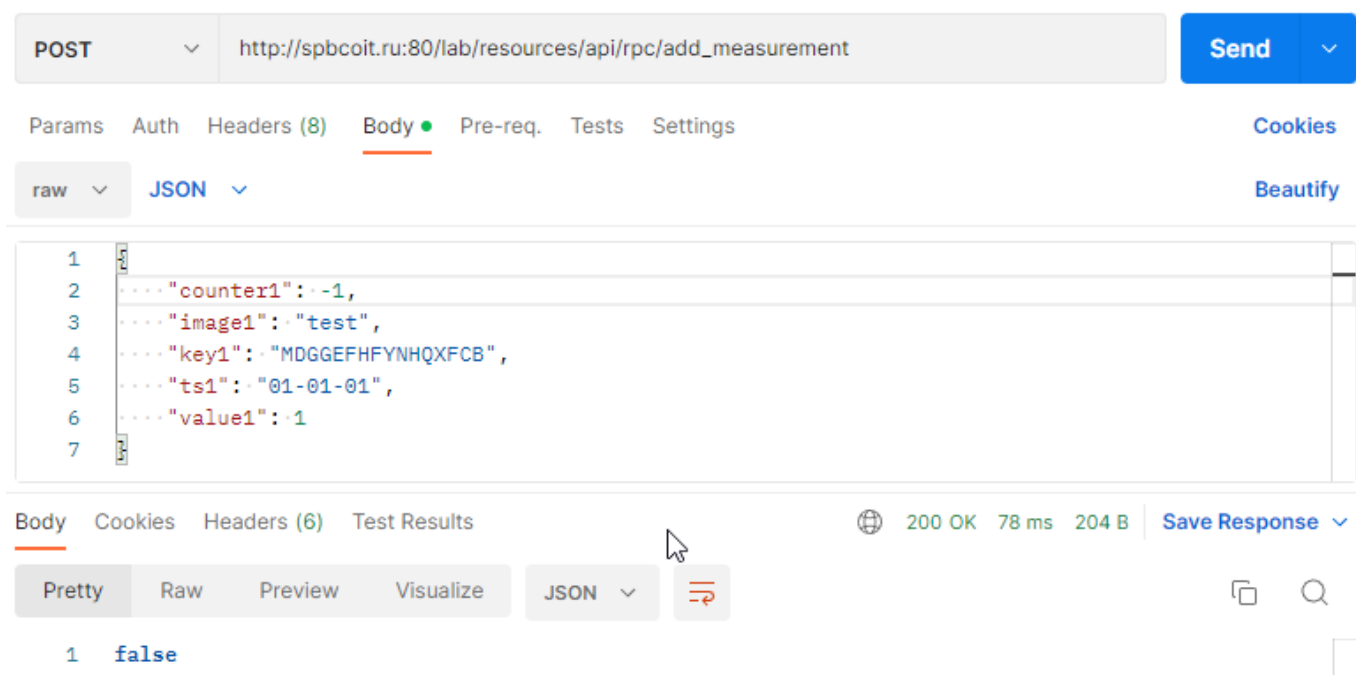


Рисунок 62 – Вызов функции add\_measurement с неверными данными

### 3.5.2. Функция `get_measurements`

Позволяет получить список измерений для счетчика пользователя по ключу сессии.

*Входные параметры:*

- `counter1` – счетчик, к которому привязаны измерения.
- `key1` – ключ сессии.

*Выходные параметрами будет массив данных со следующими атрибутами:*

- `id2` – уникальный идентификатор измерения.
- `ts2` – дата и время измерения.
- `value2` – значение измерения.
- `image2` – изображение измерения,

Пример вызова функции показаны на рисунках 63 и 64.

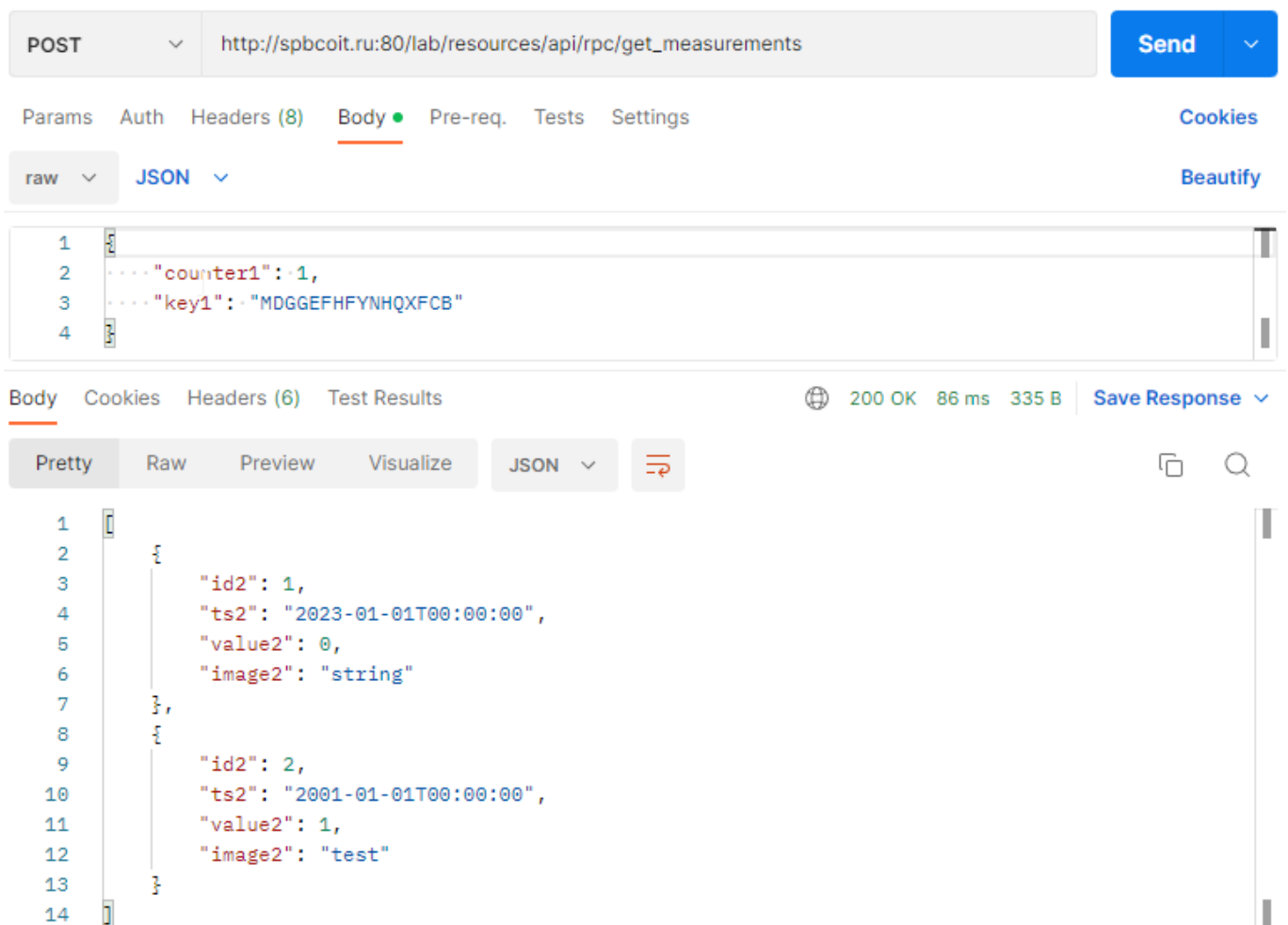


Рисунок 63 – Пример работы функции `get_measurements`

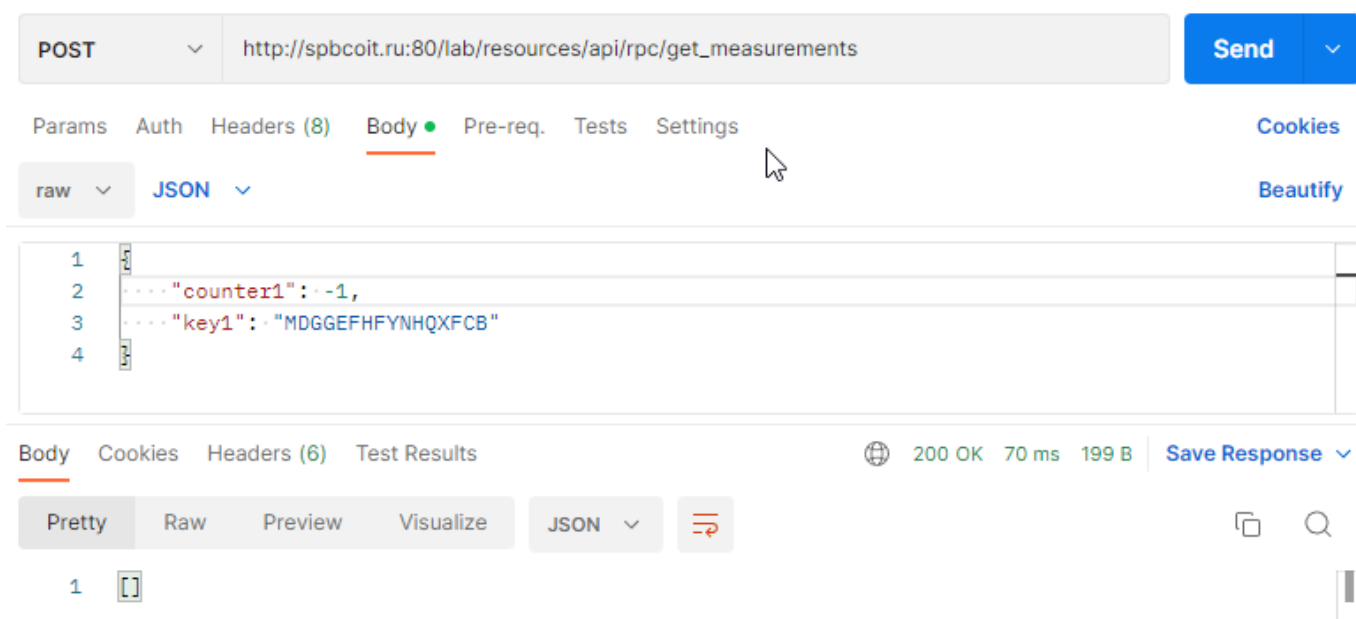


Рисунок 64 – Вызов функции get\_measurements с неверными данными

### 3.5.3. Функция update\_measurement

Позволяет обновить свойства измерения с указанием его идентификатора, ключа сессии и новых данных.

*Входные параметры:*

— Pattern – pattern.

*Выходные параметры:*

— true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 65 и 66.



Рисунок 65 – Пример работы функции update\_measurement



Рисунок 66 – Вызов функции update\_measurement с неверными данными

### 3.5.4. Функция delete\_measurement

Позволяет удалить измерение с указанием его идентификатора и ключа сессии.

*Входные параметры:*

- key1 – ключ сессии.
- measurement1 – идентификатор измерения.

*Выходные параметры:*

- true или false в зависимости от успеха операции.

Пример вызова функции показаны на рисунках 67 и 68.



Рисунок 67 – Пример работы функции delete\_measurement

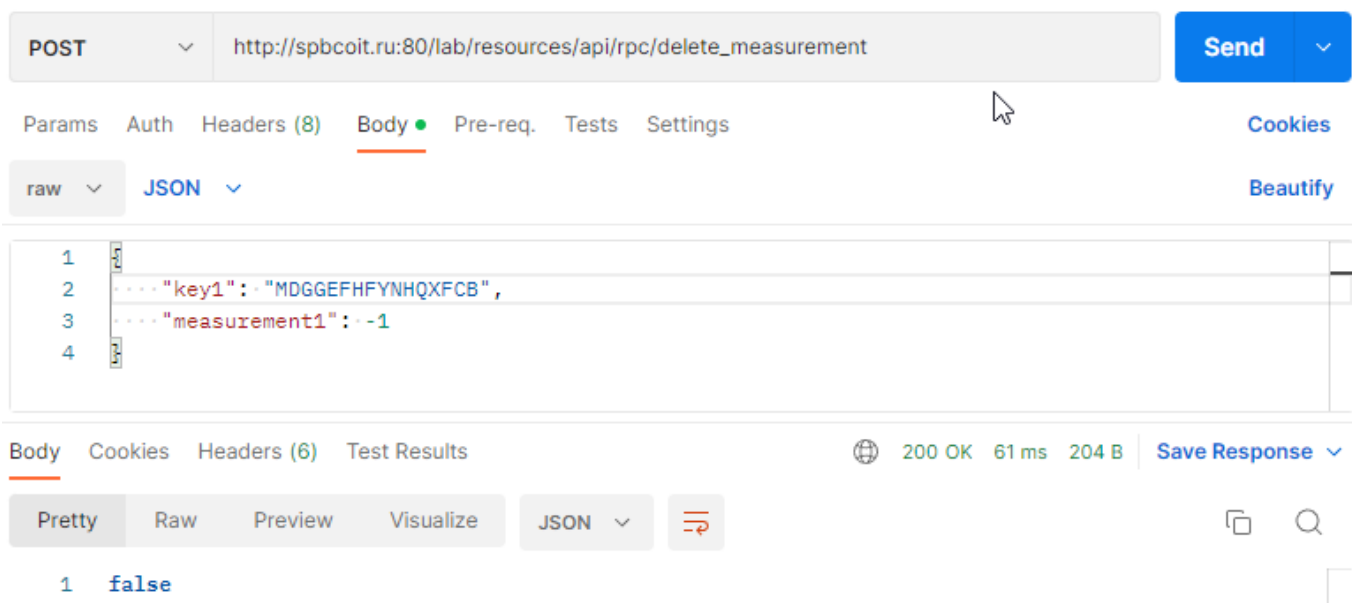


Рисунок 68 – Вызов функции delete\_measurement с неверными данными

## Работа приложения

Pattern.



Рисунок 61 – Pattern