Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

Лабораторная работа №3

по дисциплине "Тестирование программного обеспечения"

ТЕМА РАБОТЫ:

Тестирование REST API

Выполнил:

студент гр. ПРИ-120

Парахин К.В.

Приняла:

Ассистент кафедры ИСПИ

Породникова П.А.

Владимир 2022 г.

Цель работы:

Научиться проектировать тестовые сценарии для тестирования REST API. Познакомиться с инструментом тестирования REST API - программой Postman.

Ход работы:

**API** расшифровывается как «интерфейс программирования приложений». Он позволяет осуществлять связь и обмениваться данными между двумя отдельными модулями программы. Система программного обеспечения, реализующая API, содержит функции/подпрограммы, которые могут быть выполнены с помощью другого программного обеспечения.

Выполнение работы:

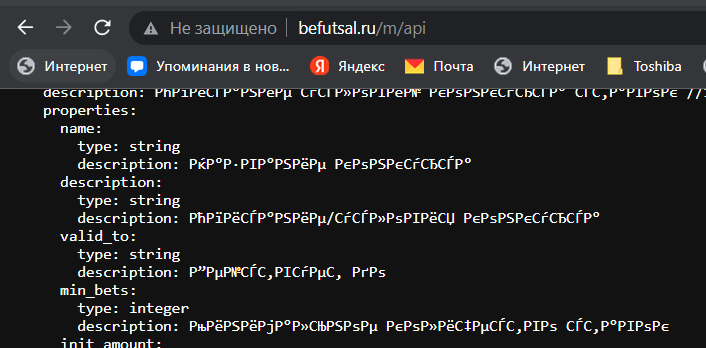


Рисунок 1. Копирование содержимого Befutsal API

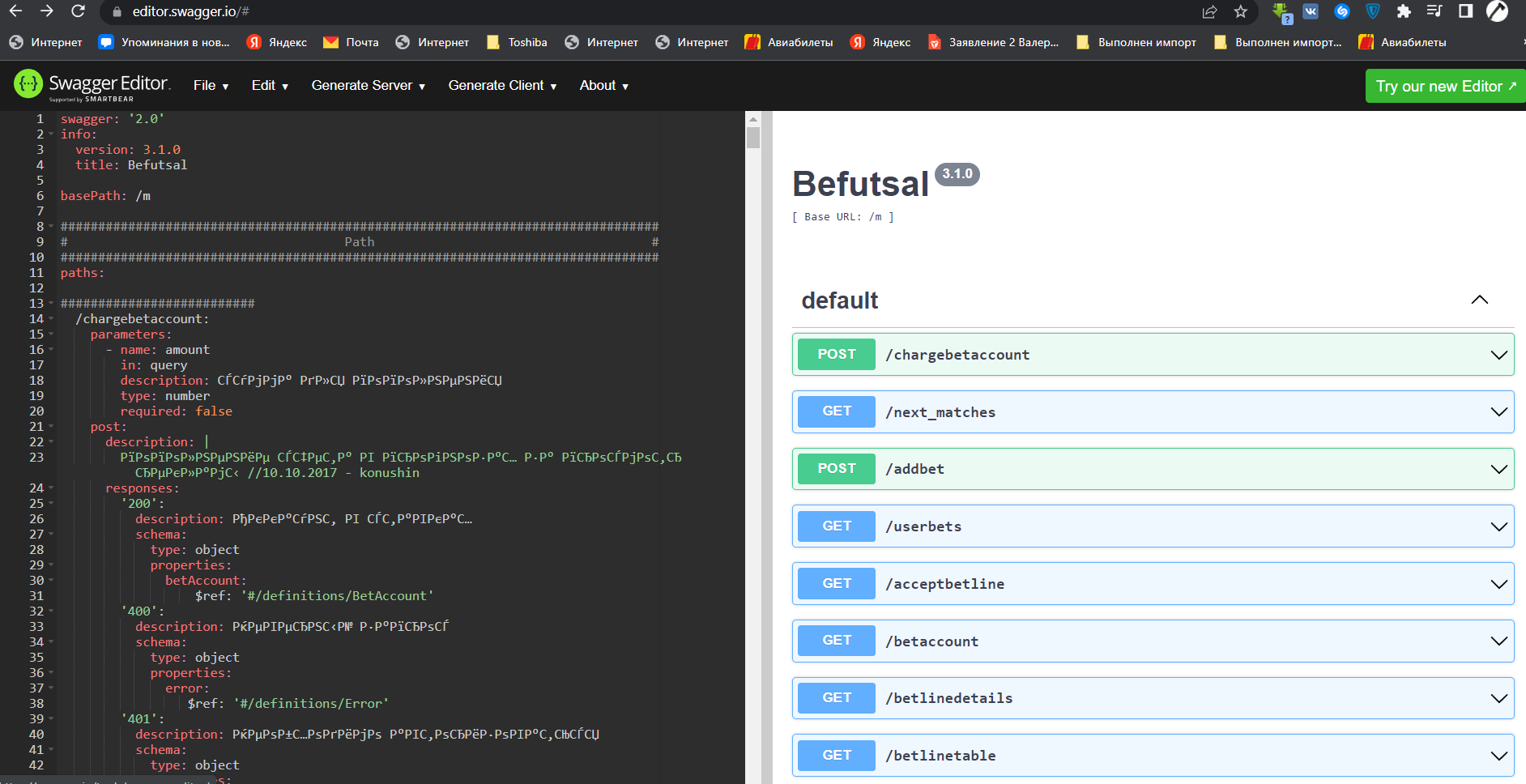


Рисунок 2. Вставка содержимого Befutsal API в Swagger Editor

Скриншоты составления примеров запросов к Befutsal API (примеры некорректных запросов с некорректными ответами):

Пример POST – запроса на списание денег с аккаунта (ChargeBetAccount):

Ниже (к тому же) приведен некорректный запрос (который был специально взят неправильным, например, из возможного ответа API):

Ожидаемый ответ: Так как пользователь не авторизован, то ему не доступен функционал API – поэтому вне зависимости от содержания запроса мы получим ответ HTTP 401 – Unathorized.

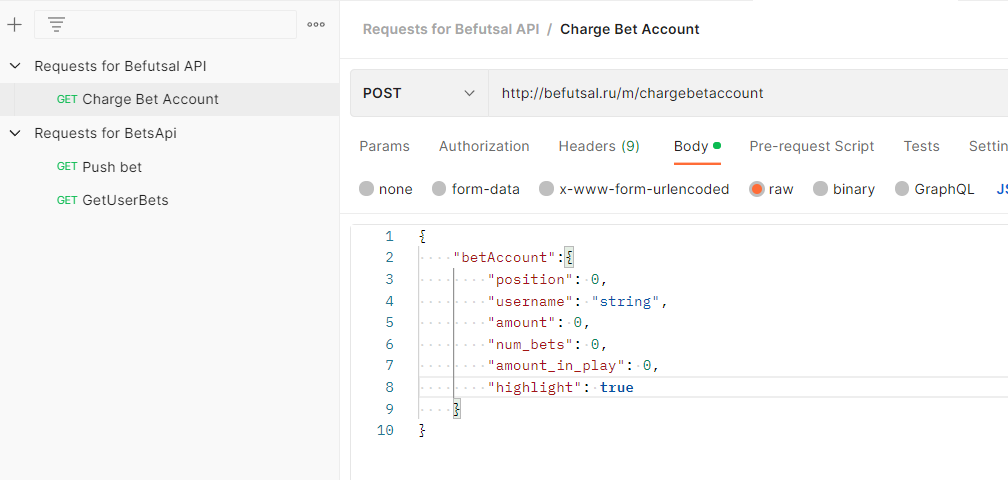
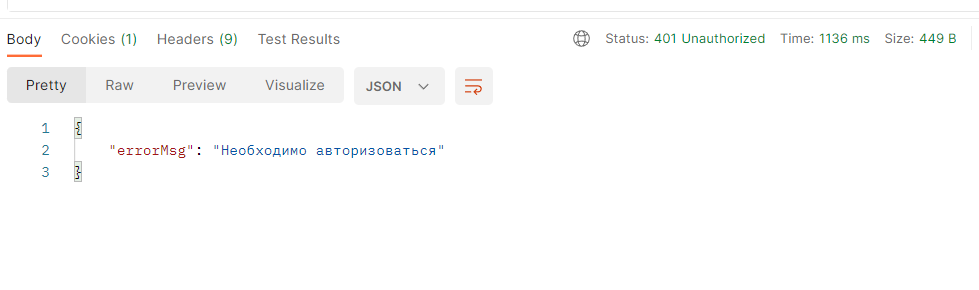


Рисунок 3.1. Составление запроса ChargeBetAccount

Рисунок 3.2. Фактический ответ (совпал с ожидаемым)

Пример GET – запроса на получение следующих матчей (GetNextMatches):

Ниже (к тому же) приведен некорректный запрос (который был специально взят неправильным, например, из возможного ответа API):

Ожидаемый ответ: Так как пользователь не авторизован, то ему не доступен функционал API – поэтому вне зависимости от содержания запроса мы получим ответ HTTP 401 – Unathorized.

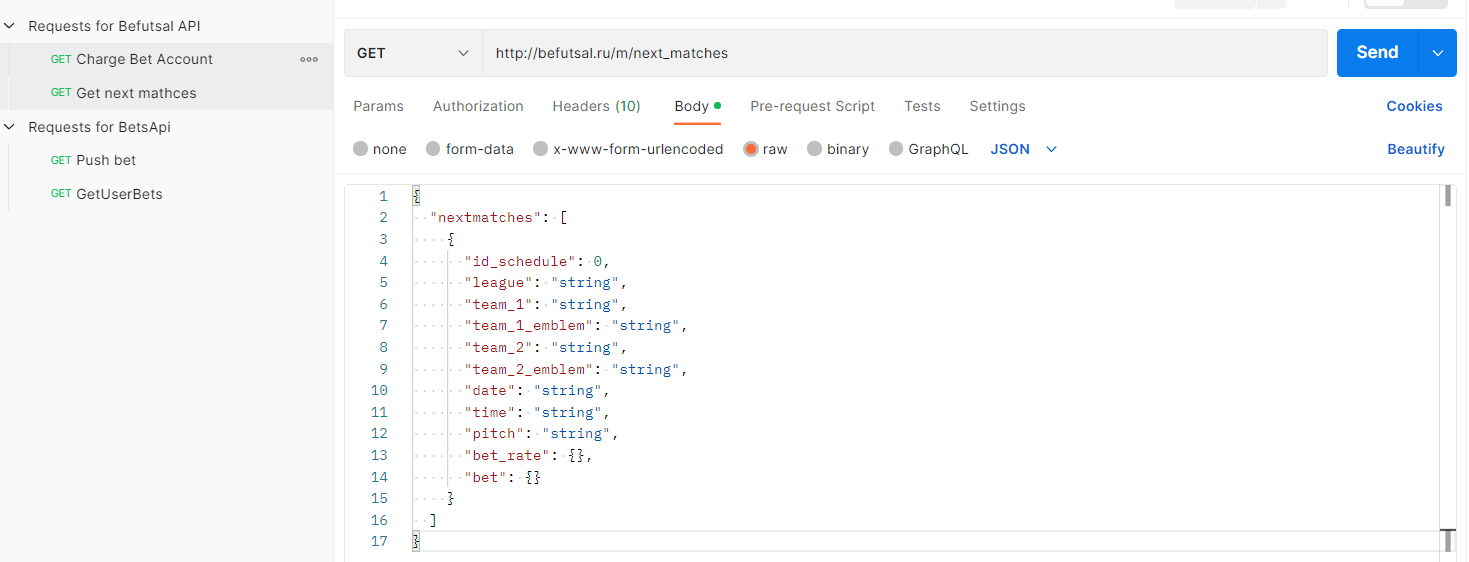


Рисунок 4. Составление запроса GetNextMatches

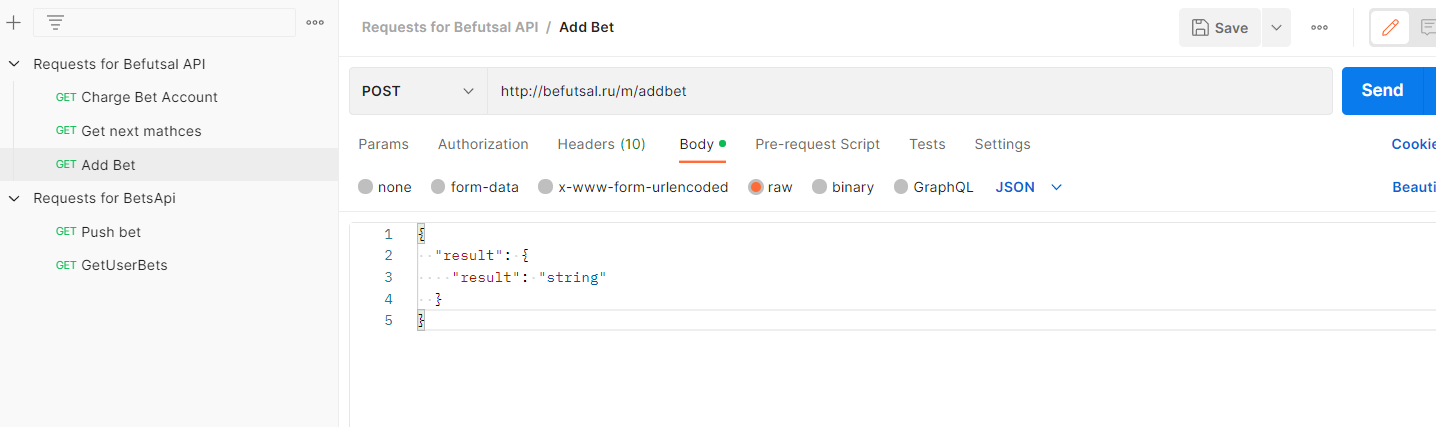


Рисунок 5. Составление запроса AddBet

Ниже приведено еще 2 POST-запроса с некорретными данными (запрос на авторизацию – Login и запрос на регистрацию – Register):

Ожидаемый ответ от POST – запроса Login: Так были введены некорректные данные (список корректных данных описан далее в тестовых сценариях для конкретных заданий с использованием запроса Login ), то будет получен некорректный ответ – HTTP 400 Bad Request.

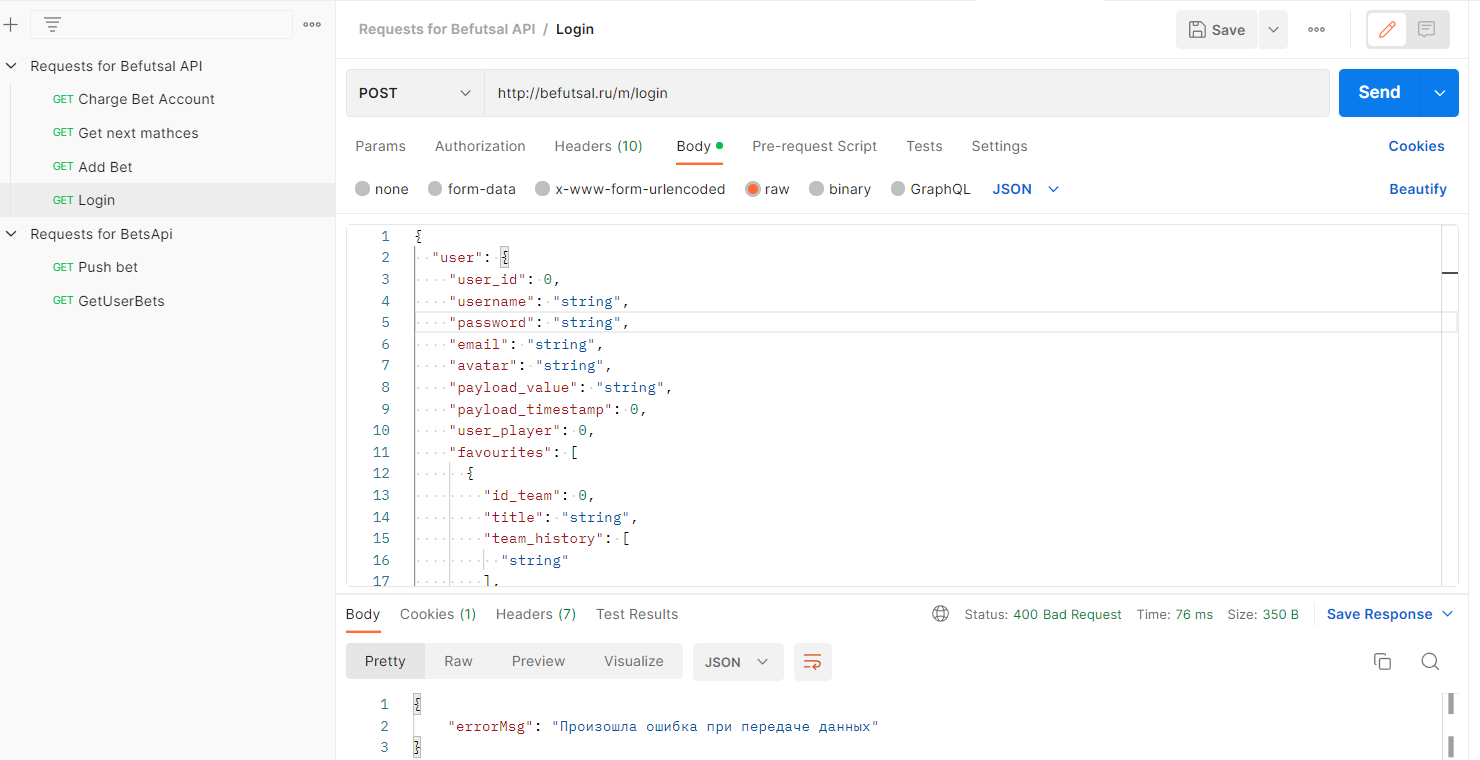


Рисунок 6. Фактический ответ на запрос Login (совпал с ожидаемым)

Ожидаемый ответ от POST – запроса Register: Так были введены некорректные данные (список корректных данных описан далее в тестовых сценариях для конкретных заданий с использованием запроса Register), то будет получен некорректный ответ – HTTP 400 Bad Request.

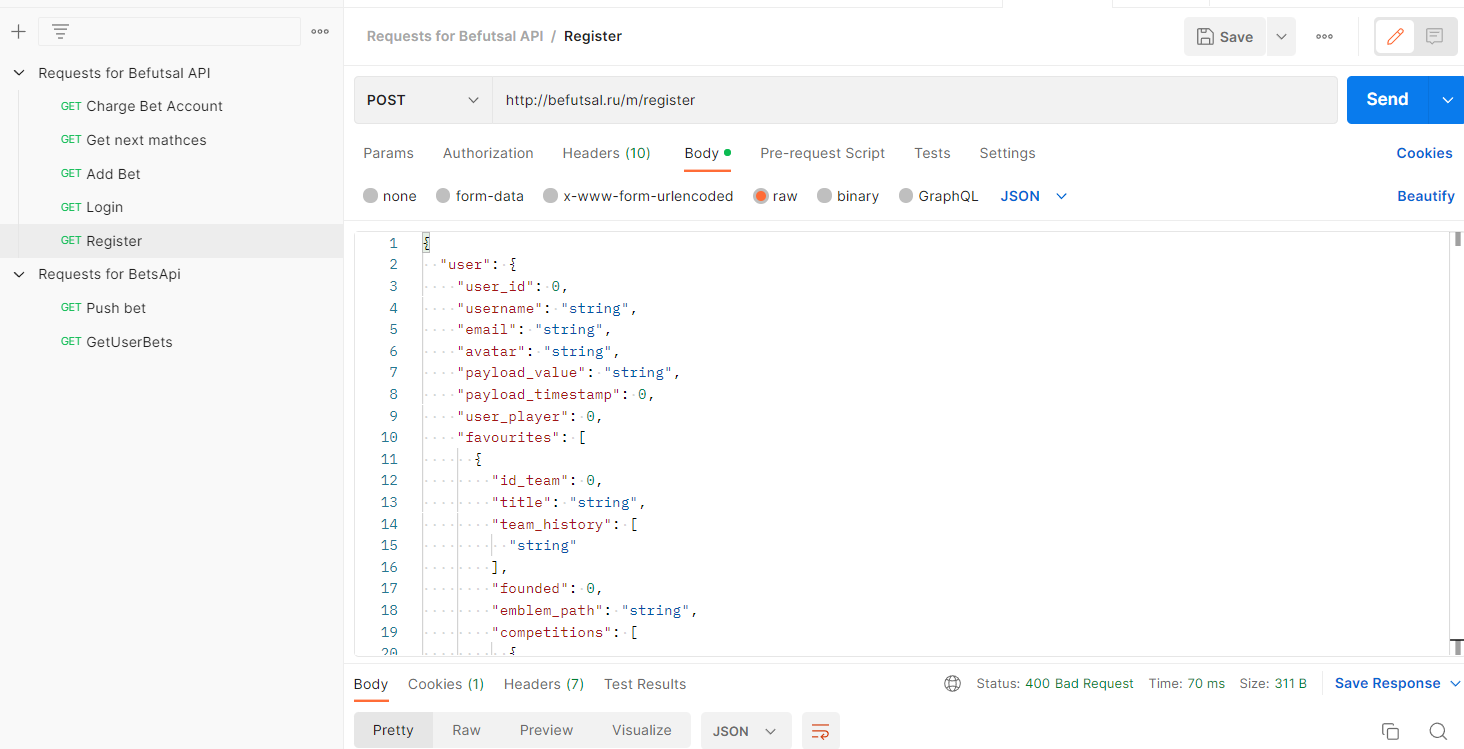


Рисунок 7. Фактический ответ на запрос Register (совпал с ожидаемым)

Сначала была изучена спецификация API – портала befutsal/api.ru, по которй ниже было составлено описание тестовых сценариев.

Описание 6 изученных тестовых сценариев:

Тест-кейс № 1: Зарегистрироваться, получить список всех команд, добавить в избранное 2 команды.

|  |  |
| --- | --- |
| Идентификатор | 1 |
| Название | Регистрация |
| Описание | Регистрация нового пользователя в системе |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/register>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на регистрацию; 6. Нажать Send; |
| Входные данные | "username": "KirillParahin",      "email": "parahinkirill@gmail.com",      "password": "kirill2002",      "uid": "1" |
| Ожидаемый результат | Произошла успешная регистрация пользователя. |
|  | |
| Идентификатор | 2 |
| Название | Получить список всех команд |
| Описание | Получение полного списка команд с их описанием |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/teams> 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | Входные данные не требуются |
| Ожидаемый результат | Были выгружены все команды с их описанием |
|  | |
| Идентификатор | 3 |
| Название | Добавить в избранное команду |
| Описание | Происходит добавление команды в список избранных команд пользователя |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/favourites>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на вход в систему; 6. Нажать Send. |
| Входные данные | "add": "true",      "team\_id": "224668488"  "add": "true",      "team\_id": "224668080" |
| Ожидаемый результат | Произошло успешное добавление команды в избранное (для данного тестового сценария действие проводится 2 раза с разными данными) |

Тест-кейс 2: Войти, получить список всех команд, получить список предстоящих игр для любой команды.

|  |  |
| --- | --- |
| Идентификатор | 4 |
| Название | Войти |
| Описание | Вход пользователя с данными, которые были указаны им при регистрации |
| Шаги | * 1. Открыть Postman;   2. Установить тип запроса POST;   3. Указать ссылку: <http://befutsal.ru/m/login> ;   4. В разделе Body выбрать raw и JSON;   5. В текстовом поле раздела Body прописать параметры на регистрацию;   6. Нажать Send; |
| Входные данные | "email": "parahinkv@gmail.com",      "password": "kirill2002",  "uid": "1" |
| Ожидаемый результат | Произошла успешная авторизация пользователя |
|  | |
| Идентификатор | 5 |
| Название | Получить список всех команд |
| Описание | Получение полного списка команд с их описанием |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/teams>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | Входные данные не требуются |
| Ожидаемый результат | Были выгружены данные о всех командах |
|  | |
| Идентификатор | 6 |
| Название | Получить список предстоящих игр для любой команды |
| Описание | Происходит выгрузка всех следующих матчей для определенной команды, указанной в параметрах |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/next_matches>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на вход в систему; 6. Нажать Send. |
| Входные данные | "team\_id": 224668080 |
| Ожидаемый результат | Были выгружены все предстоящие матчи для указанной команды |

Тест-кейс 3: Войти, получить список всех команд, получить турнирную таблицу любой команды.

|  |  |
| --- | --- |
| Идентификатор | 7 |
| Название | Войти |
| Описание | Вход пользователя с данными, которые были указаны им при регистрации |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/login> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на регистрацию; 6. Нажать Send; |
| Входные данные | "email": "parahinkirill@gmail.com",      "password": "kirill2002" |
| Ожидаемый результат | Произошла успешная авторизация пользователя. |
|  | |
| Идентификатор | 8 |
| Название | Получить список всех команд |
| Описание | Получение полного списка команд с их описанием |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/teams> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | Входные данные не требуются |
| Ожидаемый результат | Были выгружены данные о всех командах |
|  | |
| Идентификатор | 9 |
| Название | Получить турнирную таблицу любой команды |
| Описание | Для выбранного соревнования и выбранной команды выгружается информация из турнирной таблицы |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/leaguetable> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на вход в систему; 6. Нажать Send. |
| Входные данные | "competition\_id": 224668368,      "team\_id": "1438" |
| Ожидаемый результат | Произошла выгрузка информации о выбранной команде из турнирной таблицы |

Тест-кейс № 4: Войти, получить список всех команд, получить результаты игр любой команды.

|  |  |
| --- | --- |
| Идентификатор | 10 |
| Название | Войти |
| Описание | Вход пользователя с данными, которые были указаны им при регистрации |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/login> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на регистрацию; 6. Нажать Send; |
| Входные данные | "email": "parahinkirill@gmail.com",      "password": "kirill2002" |
| Ожидаемый результат | Произошла успешная авторизация пользователя. |
|  | |
| Идентификатор | 11 |
| Название | Получить список всех команд |
| Описание | Получение полного списка команд с их описанием |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/teams>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | Входные данные не требуются |
| Ожидаемый результат | Были выгружены данные о всех командах |
|  | |
| Идентификатор | 12 |
| Название | Получить результаты игр любой команды |
| Описание | Происходит выгрузка всех результатов для сыгранных игр указанной команды |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/leaguetable/teamdetails> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на вход в систему; 6. Нажать Send. |
| Входные данные | "competition\_id": 224668368,      "team\_id": 1438 |
| Ожидаемый результат | Произошла выгрузка всех результатов для выбранной команды и соревнования |

Тест-кейс №5: Войти, получить информацию о контактах федераций.

|  |  |
| --- | --- |
| Идентификатор | 13 |
| Название | Войти |
| Описание | Вход пользователя с данными, которые были указаны им при регистрации |
| Шаги | 1.Открыть Postman;   1. Установить тип запроса POST; 2. Указать ссылку: <http://befutsal.ru/m/login> ; 3. В разделе Body выбрать raw и JSON; 4. В текстовом поле раздела Body прописать параметры на регистрацию; 5. Нажать Send; |
| Входные данные | "email": "parahinkirill@gmail.com",      "password": "kirill2002" |
| Ожидаемый результат | Произошла успешная авторизация пользователя. |
|  | |
| Идентификатор | 14 |
| Название | Получить информацию о контактах федераций |
| Описание | Происходит выгрузка всех данных о контактах федераций для таблицы лиги |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/leaguetable/contacts> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать |
| Входные данные | Входные данные не требуются |
| Ожидаемый результат | Произошла успешная выгрузка данных о контактах федераций, имеющихся с таблице |

Тест кейс № 6 (мой собственный сценарий): Зарегистрироваться, войти, посмотреть протоколы команды, подробно посмотреть протокол, выйти с аккаунта

|  |  |
| --- | --- |
| Идентификатор | 15 |
| Название | Регистрация |
| Описание | Регистрация нового пользователя в системе |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/register>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на регистрацию; 6. Нажать Send; |
| Входные данные | "username": "KirillParahin",      "email": "parahinkirill@gmail.com",      "password": "kirill2002",      "uid": "2" |
| Ожидаемый результат | Произошла успешная регистрация пользователя. |
|  | |
| Идентификатор | 16 |
| Название | Войти |
| Описание | Вход пользователя с данными, которые были указаны им при регистрации |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса POST; 3. Указать ссылку: <http://befutsal.ru/m/changepassword>; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | "email": "parahinkirill@gmail.com",      "password": "kirill2002" |
| Ожидаемый результат | Произошла успешная авторизация пользователя |
|  | |
| Идентификатор | 17 |
| Название | Посмотреть протоколы команды |
| Описание | Получение всех записанных протоколов для указанной команды |
| Шаги | 1.Открыть Postman;  2.Установить тип запроса GET;   1. Указать ссылку: <http://befutsal.ru/m/protocols> ; 2. В разделе Body выбрать raw и JSON; 3. В текстовом поле раздела Body прописать параметры на вход в систему; 4. Нажать Send. |
| Входные данные | "team\_id": 1438 |
| Ожидаемый результат | Произошла выгрузка всех протоколов указанной команды |
| Идентификатор | 18 |
| Название | Подробно посмотреть протокол |
| Описание | Подробный просмотр конкретного протокола команды среди остальных протоколов |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/detailed_protocol> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на вход в систему; 6. Нажать Send. |
| Входные данные | "protocol\_id": 224708241 |
| Ожидаемый результат | Была показана подробная информация по указанному в параметрах протоколу |
| Идентификатор | 19 |
| Название | Выйти из аккаунта |
| Описание | Выход пользователя из текущего аккаунта, сессия которого была загружена ранее |
| Шаги | 1. Открыть Postman; 2. Установить тип запроса GET; 3. Указать ссылку: <http://befutsal.ru/m/logout> ; 4. В разделе Body выбрать raw и JSON; 5. В текстовом поле раздела Body прописать параметры на смену пароля; 6. Нажать Send. |
| Входные данные | Входных параметров нет |
| Ожидаемый результат | Произошел выход из текущей учетной записи |

Тестирование и более подробное описание тестовых сценариев (тест-кейсов):

Сценарий 1 - Зарегистрироваться, получить список всех команд, добавить в избранное 2 команды.

В этот сценарий входит следующая последовательность запросов:

- Регистрация (/register - post-request)

Обязательные параметры для минимального запроса:

- username – string

- email – string

- password – string

- uid – string

В случае отсутствия ввода хотя бы одного параметра в запрос, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Ниже представлен пример минимального запроса (рис.8) и ответа (рис.9):

Ожидаемый ответ от запроса: HTTP 200 Ok

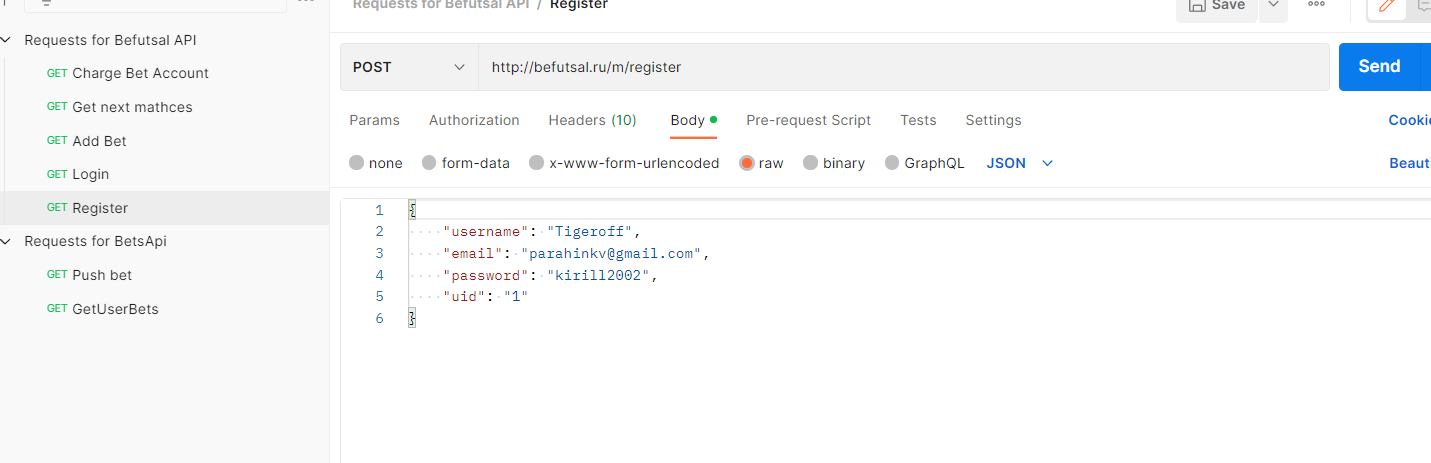


Рисунок 8. Корректный минимальный запрос

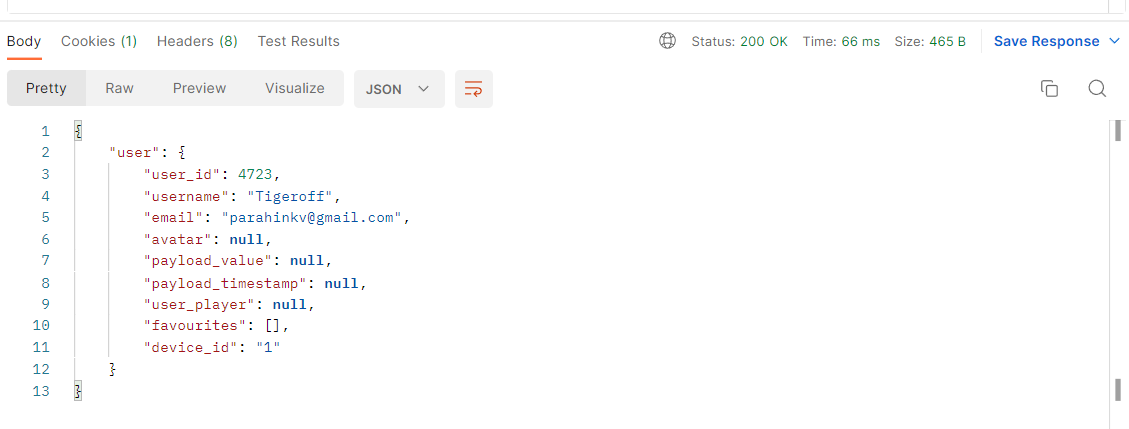


Рисунок 9. Фактический ответ (совпал с ожидаемым)

- Операция получения списка команд (/teams – get-request)

Обязательные параметры отсутствуют – минимальный запрос не содержит параметров

Ниже представлен пример минимального Get-запроса с пустым телом (рис.10) и ответа (рис.11)

Ожидаемый ответ от запроса: HTTP 200 Ok

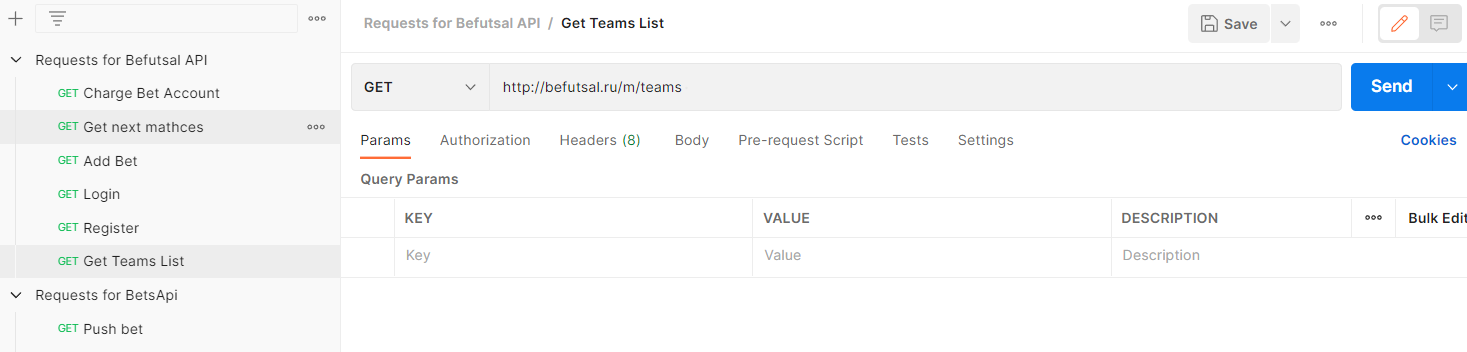


Рисунок 10. Корректный запрос без параметров

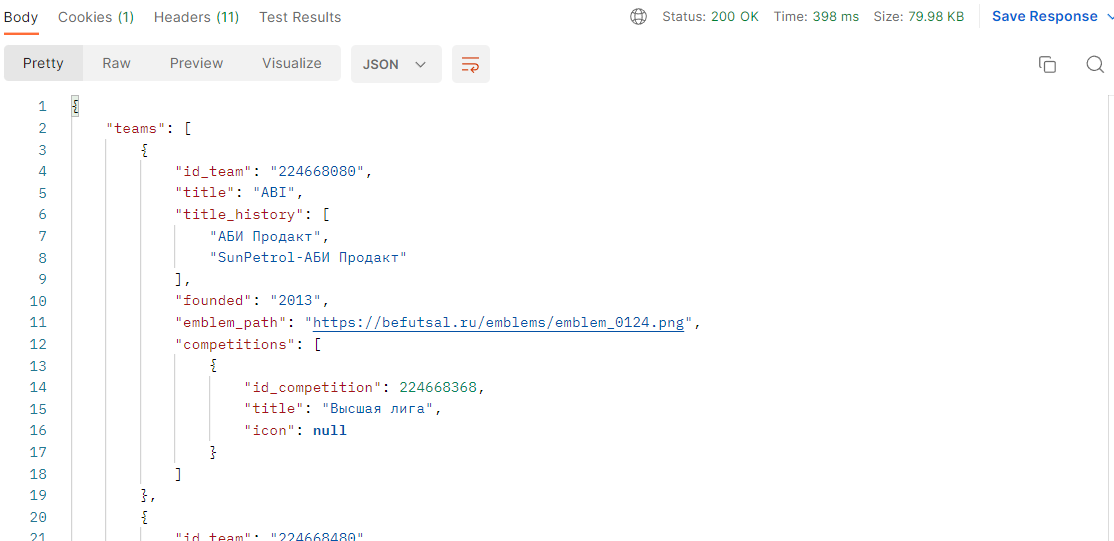


Рисунок 11. Фактический ответ (совпал с ожидаемым)

- Добавление 2 команд в избранное (2 последовательнх запроса /favourites – post-request)

Обязательные параметры для минимального запроса:

- add – boolean

- team\_id – integer

В случае отсутствия ввода хотя бы одного параметра в запрос, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Ниже представлен пример минимального и достаточного запроса на добавление первой команды в избранное (рис.12) и ответа (рис.13):

Ожидаемый ответ от запроса: HTTP 200 Ok

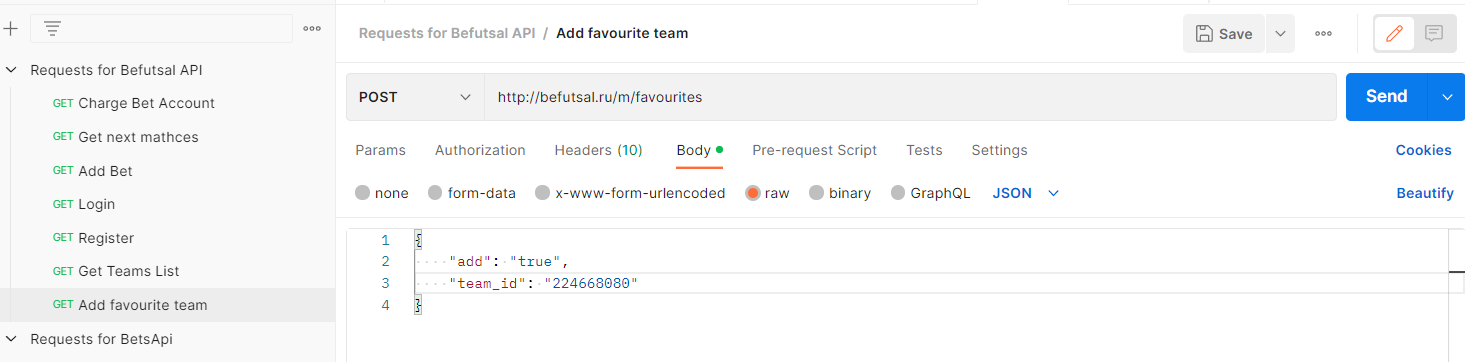


Рисунок 12. Корректный минимальный запрос

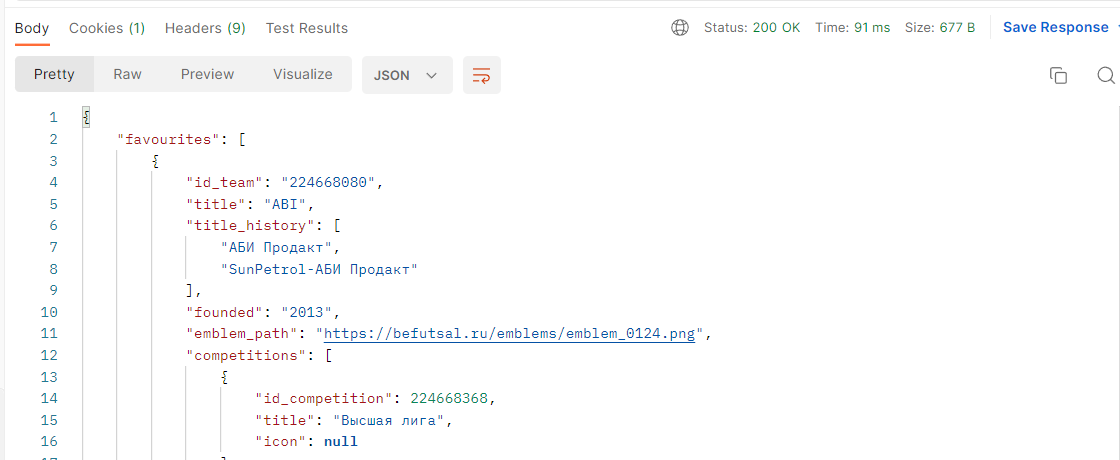


Рисунок 13. Фактический ответ (совпал с ожидаемым)

Ниже представлен пример минимального запроса на добавление второй команды в избранное (рис.14) и ответа (рис.15):

Ожидаемый ответ от запроса: HTTP 200 Ok

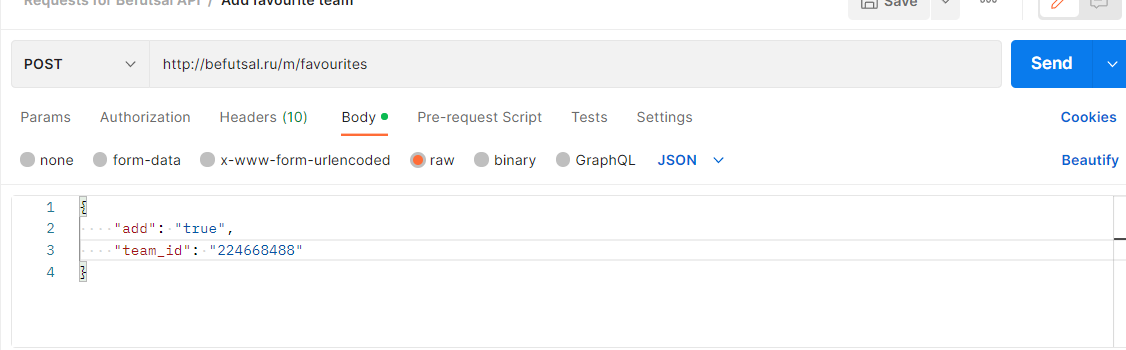


Рисунок 14. Корректный минимальный запрос

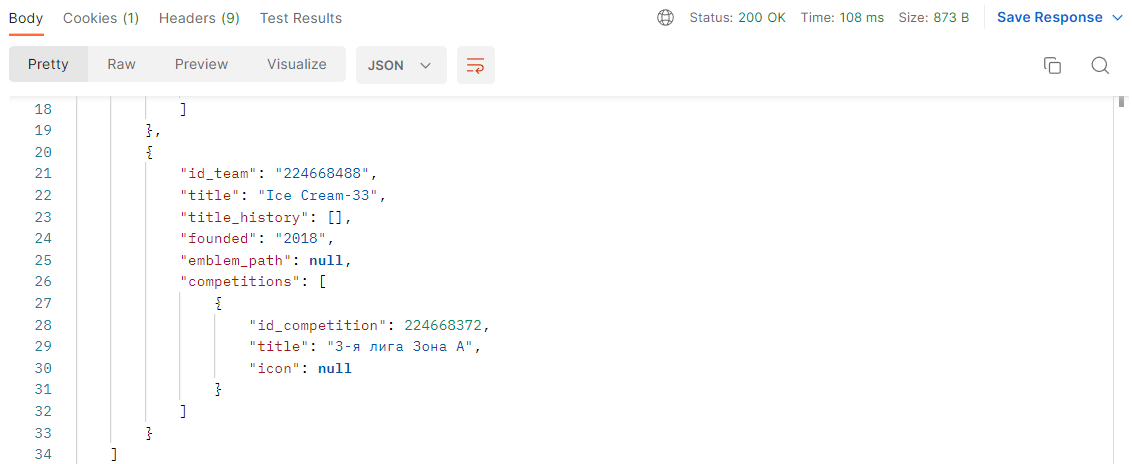


Рисунок 15. Фактический ответ (совпал с ожидаемым)

Сценарий 2 - Войти, получить список всех команд, удалить из избранного 1 команду

В этот сценарий входит следующая последовательность запросов:

- Войти (/login – post-request)

Обязательные параметры для минимального запроса:

- email – string

- password – string

В случае отсутствия ввода хотя бы одного параметра в запрос, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Кроме этого, для максимального запроса можно ввести еще один необязательный параметр:

- uid – string

Ниже представлен пример максимального запроса и ответа (рис.16)

Ожидаемый ответ от запроса: HTTP 200 Ok

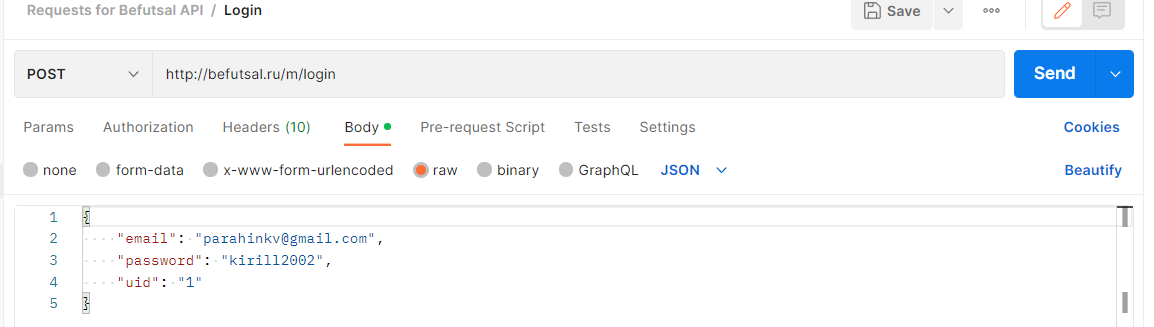




Рисунок 16. Фактический ответ (совпал с ожидаемым)

- Операция получения списка команд (/teams – get-request)

Обязательные параметры отсутствуют – минимальный запрос не содержит параметров

Ниже представлен пример минимального Get-запроса с пустым телом и ответа (рис.17):

Ожидаемый ответ от запроса: HTTP 200 Ok

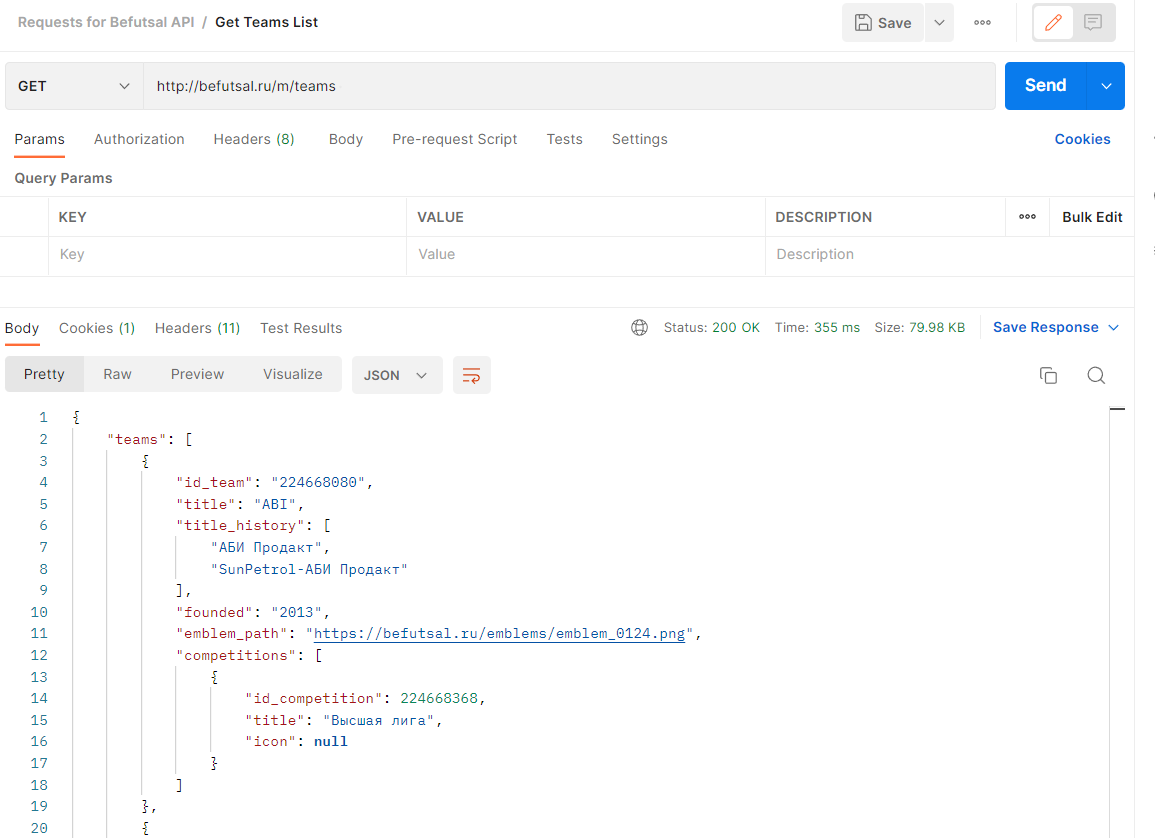


Рисунок 17. Фактический ответ на запрос (совпал с ожидаемым)

* Удалить из избранного 1 команду (/favourites – delete-request)

Обязательные параметры для минимального запроса:

- add – boolean

- team\_id – string

Ниже представлен пример минимального запроса и ответа на него (рис.18):

Ожидаемый ответ от запроса: HTTP 200 Ok

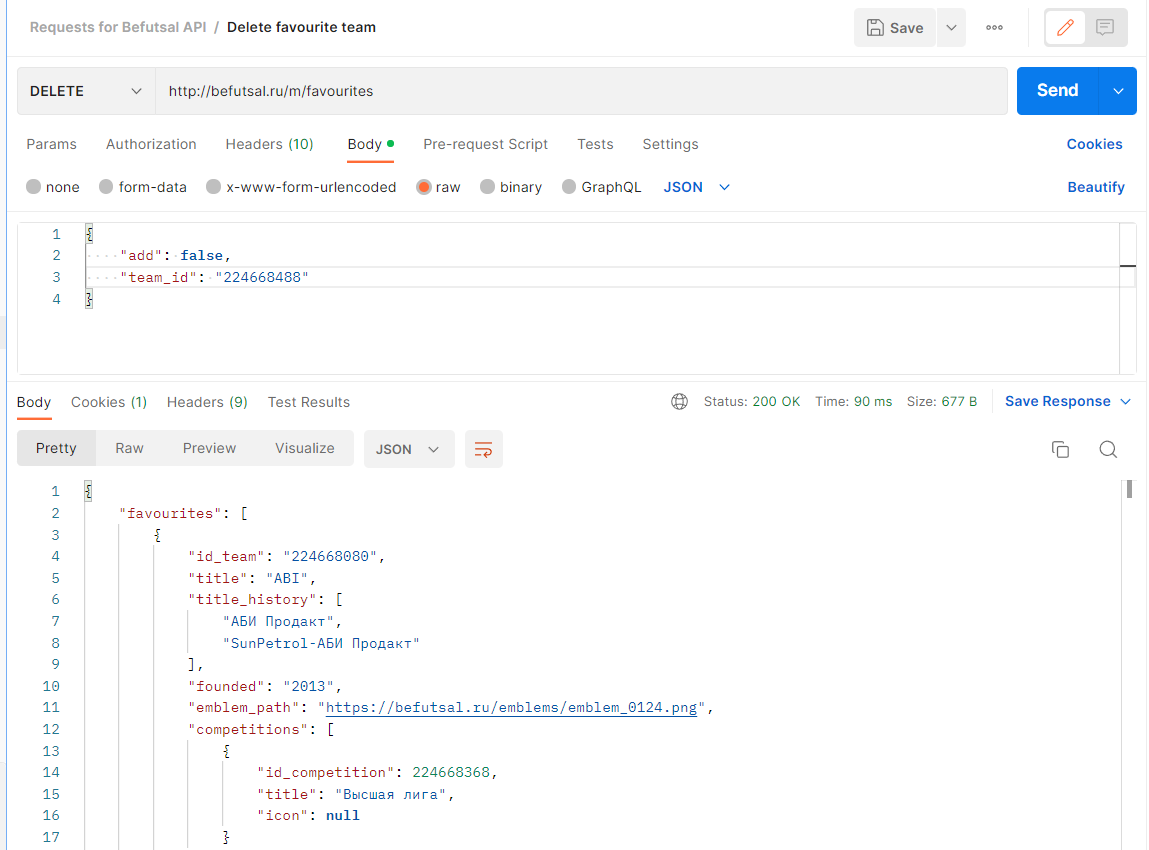


Рисунок 18. Фактический ответ (совпал с ожидаемым)

Сценарий 3 - Войти, получить список всех команд, получить турнирную таблицу любой команды.

В этот сценарий входит следующая последовательность запросов:

- Войти (/login – post-request)

Обязательные параметры для минимального запроса:

- email – string

- password – string

Ниже представлен пример минимального запроса и ответа на него (рис.19):

Ожидаемый ответ от запроса: HTTP 200 Ok

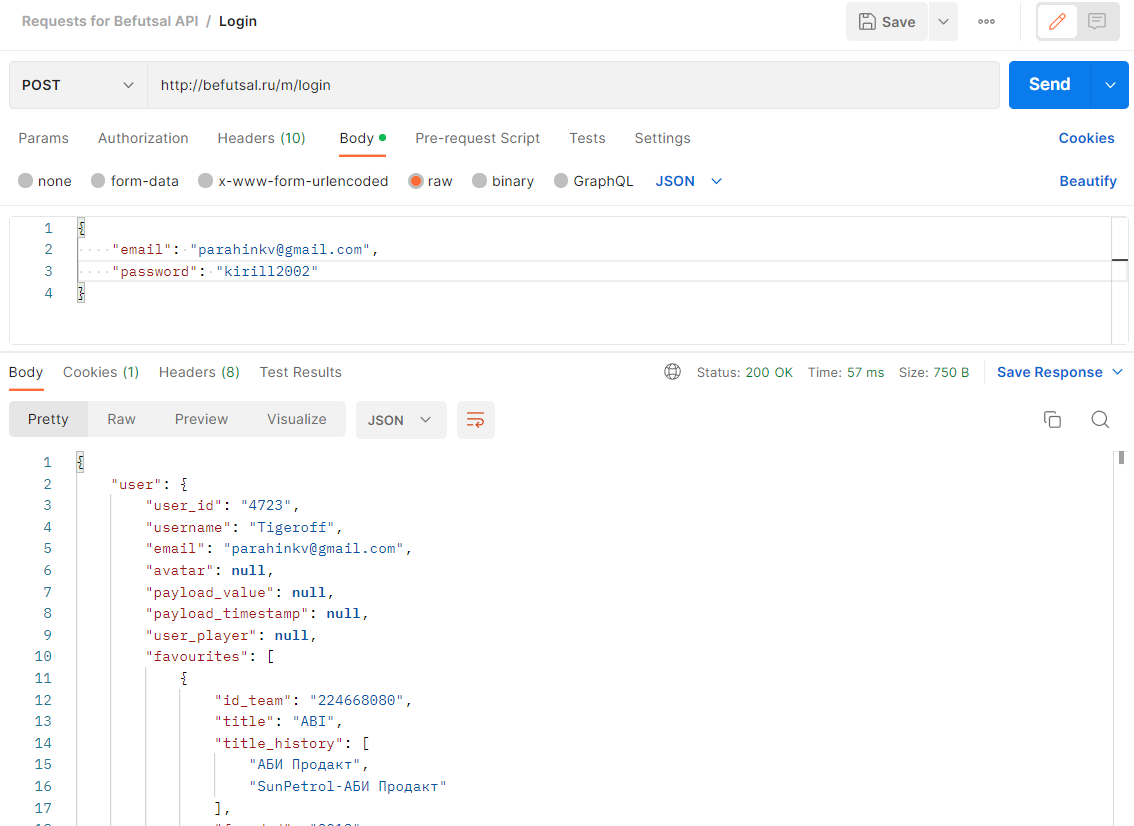


Рисунок 19. Фактический ответ на запрос (совпал с ожидаемым)

- Операция получения списка команд (/teams – get-request)

Обязательные параметры отсутствуют – минимальный запрос не содержит параметров

Ниже представлен пример минимального Get-запроса с пустым телом и ответа на него (рис.20):

Ожидаемый ответ от запроса: HTTP 200 Ok

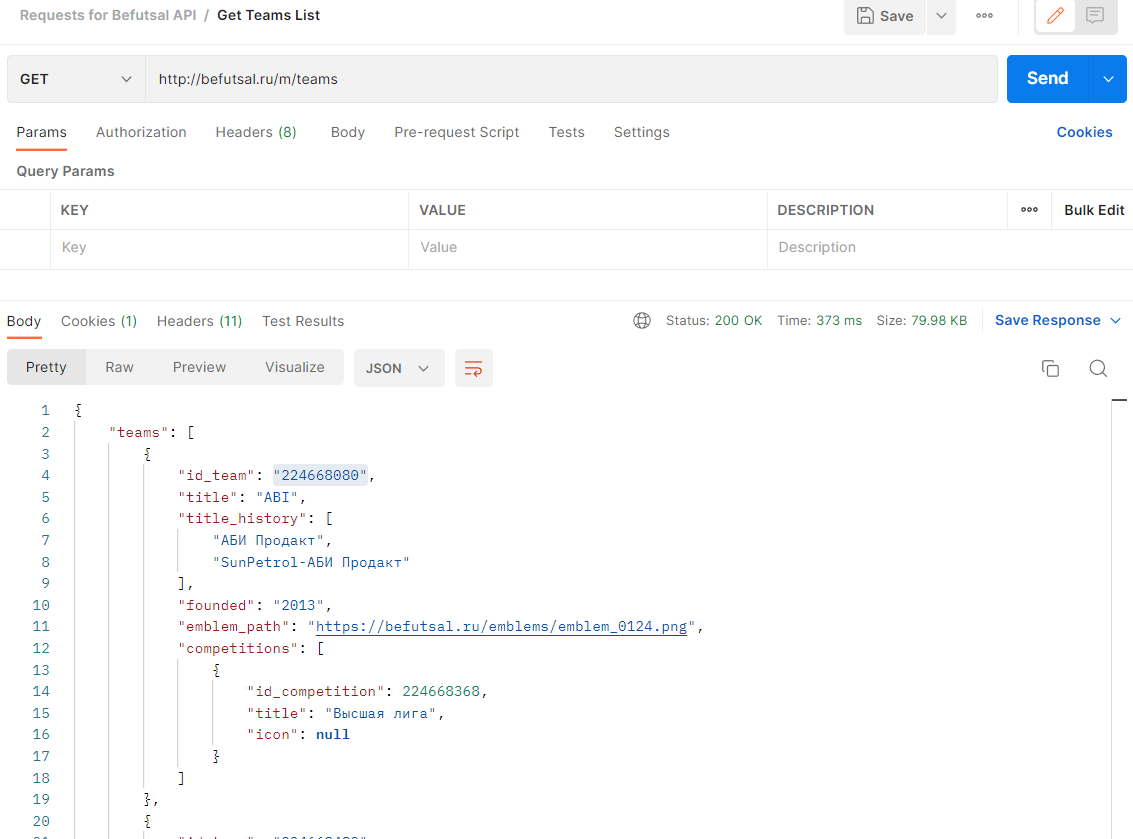


Рисунок 20. Фактический ответ на запрос (совпал с ожидаемым)

- Получение турнирной таблицы (/leaguetable – get-request)

Обязательные параметры для минимального запроса:

- competition\_id – integer

В случае отсутствия этого параметра, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Также можно ввести дополнительный параметр team\_id – integer – чтобы получить положение в турнирной таблице для определенной команды

Ниже представлен пример максимального запроса и ответа на него (рис.21):

Ожидаемый ответ от запроса: HTTP 200 Ok

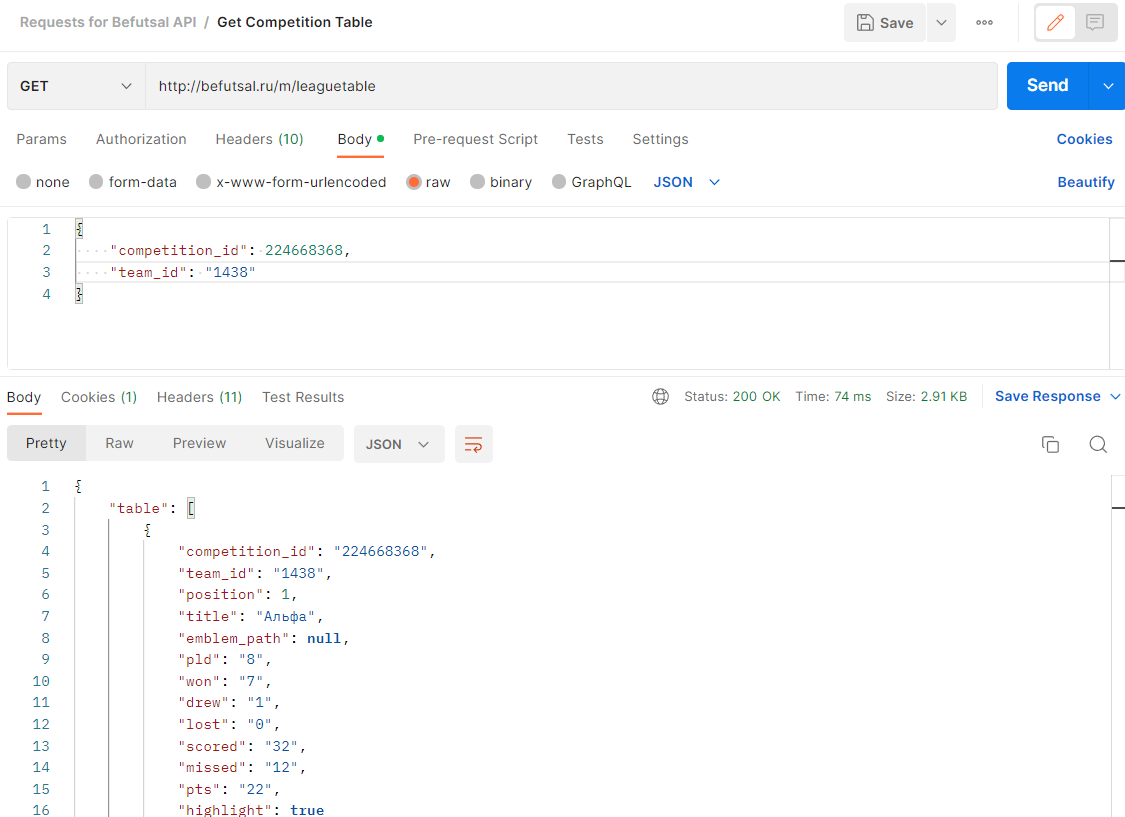


Рисунок 21. Фактический ответ на запрос (совпал с ожидаемым

Сценарий 4 - Войти, получить список всех команд, получить результаты игр любой команды.

В этот сценарий входит следующая последовательность запросов:

- Войти (/login – post-request)

Обязательные параметры для минимального запроса:

- email – string

- password – string

Ниже представлен пример минимального запроса и ответа на него (рис.22):

Ожидаемый ответ от запроса: HTTP 200 Ok

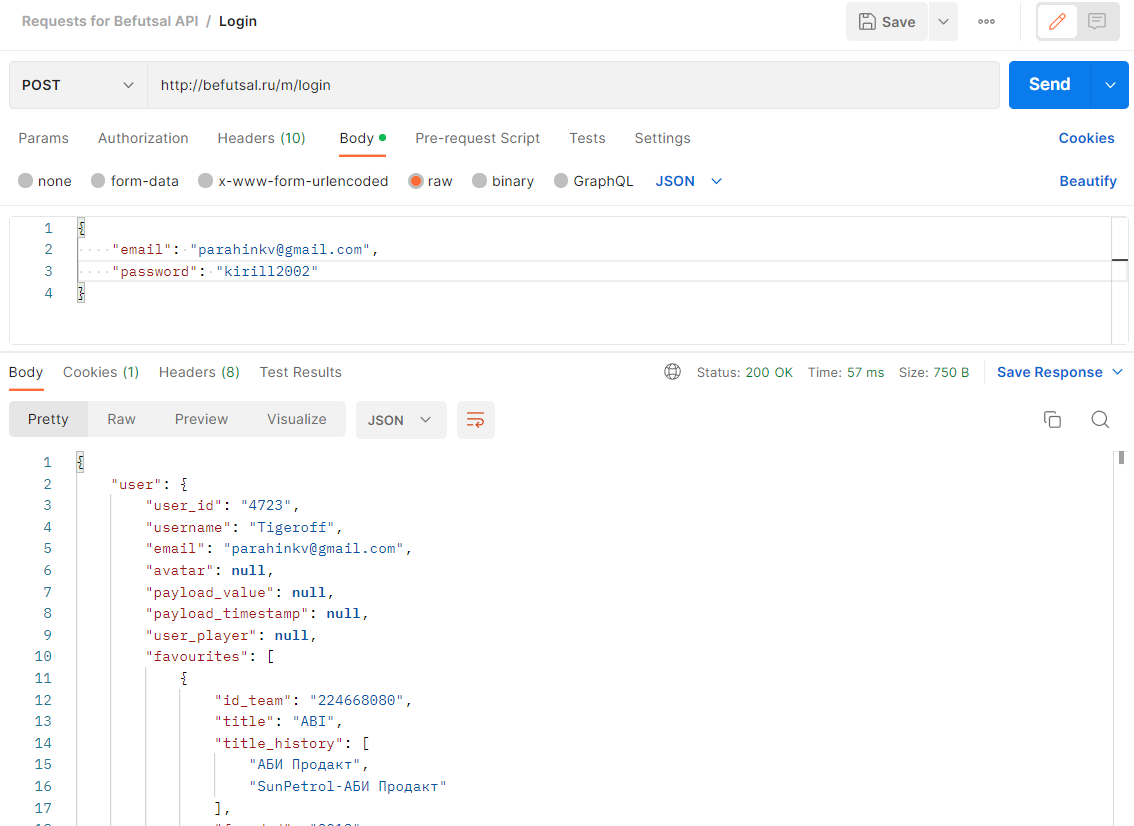


Рисунок 22. Фактический ответ на запрос (совпал с ожидаемым)

- Операция получения списка команд (/teams – get-request)

Обязательные параметры отсутствуют – минимальный запрос не содержит параметров

Ниже представлен пример минимального Get-запроса с пустым телом и ответа на него (рис.23):

Ожидаемый ответ от запроса: HTTP 200 Ok

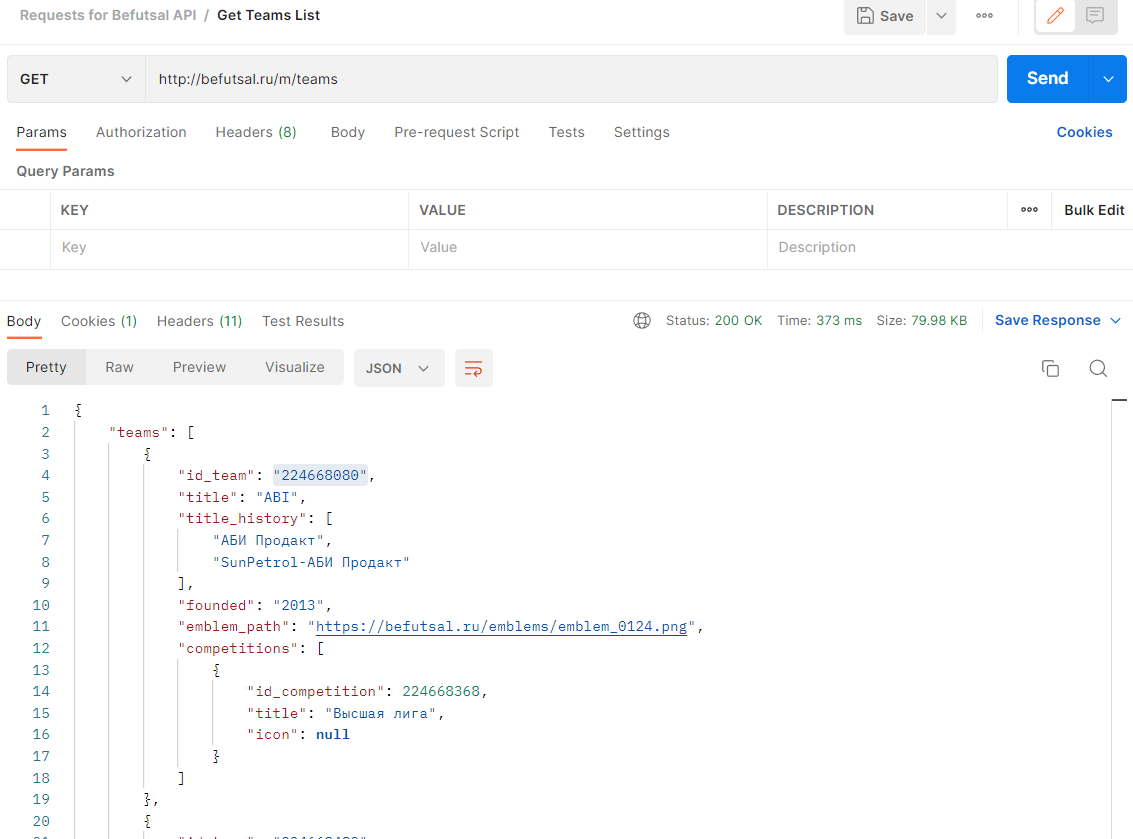


Рисунок 23. Фактический ответ на запрос (совпал с ожидаемым)

- Получить результаты игр любой команды (/teamdetails – get-request)

Обязательные параметры для минимального запроса:

- competition\_id – integer

- team\_id - integer

В случае отсутствия этого параметра, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Ниже представлен пример минимального запроса и ответа на него (рис.24):

Ожидаемый ответ от запроса: HTTP 200 Ok

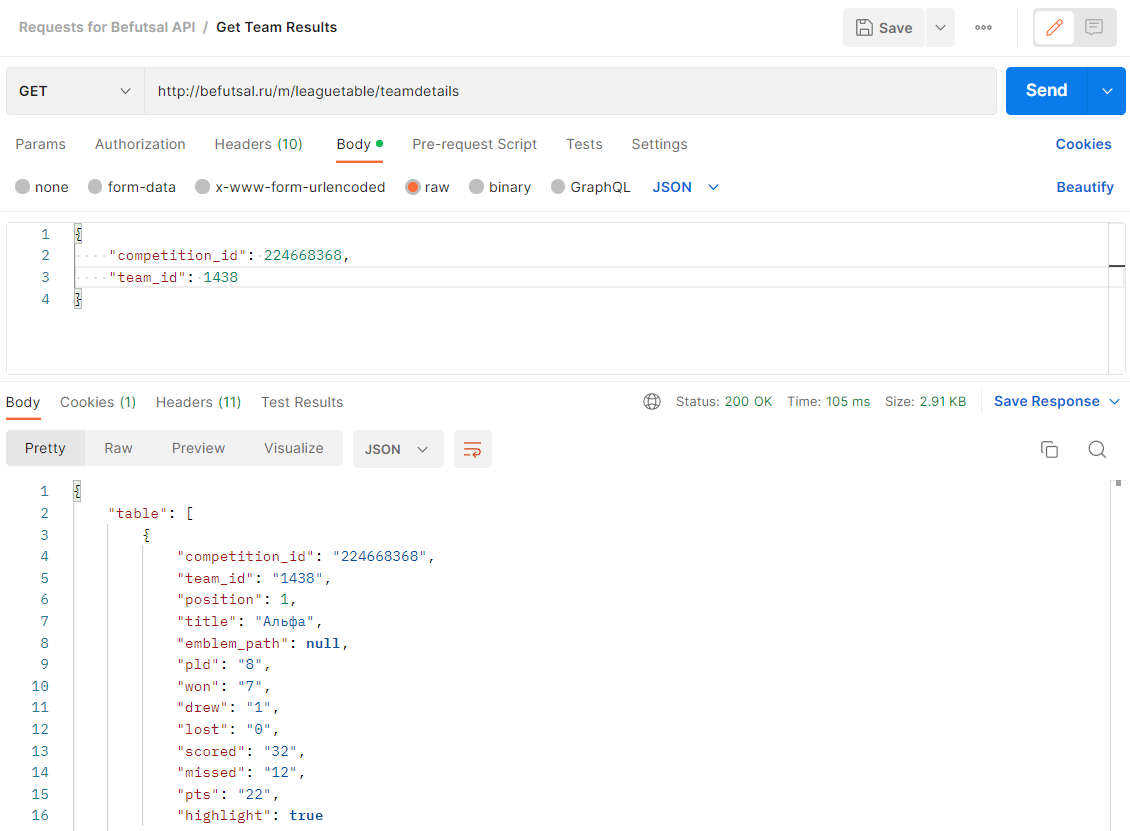


Рисунок 24. Фактический ответ на запрос (совпал с ожидаемым)

Сценарий 5 - Войти, получить информацию о контактах федераций

В этот сценарий входит следующая последовательность запросов:

- Войти (/login – post-request)

Обязательные параметры для минимального запроса:

- email – string

- password – string

Ожидаемый ответ от запроса: HTTP 200 Ok

Ниже представлен пример минимального запроса и ответа на него (рис.25):

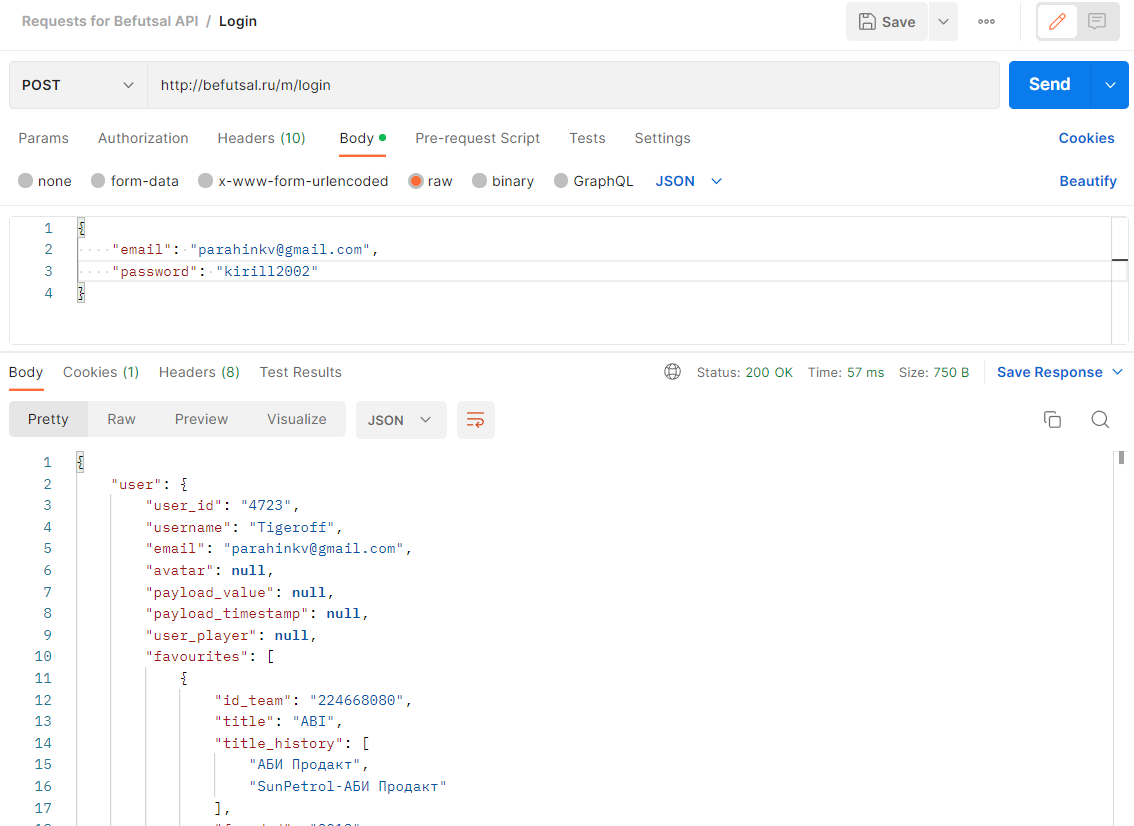


Рисунок 25. Фактический ответ на запрос (совпал с ожидаемым)

- Получить информацию о контактах федераций (/contacts – get-request)

Обязательные параметры для запроса отсуствуют – минимальный запрос пустой

Ниже представлен пример минимального запроса и ответа на него (рис.26):

Ожидаемый ответ от запроса: HTTP 200 Ok

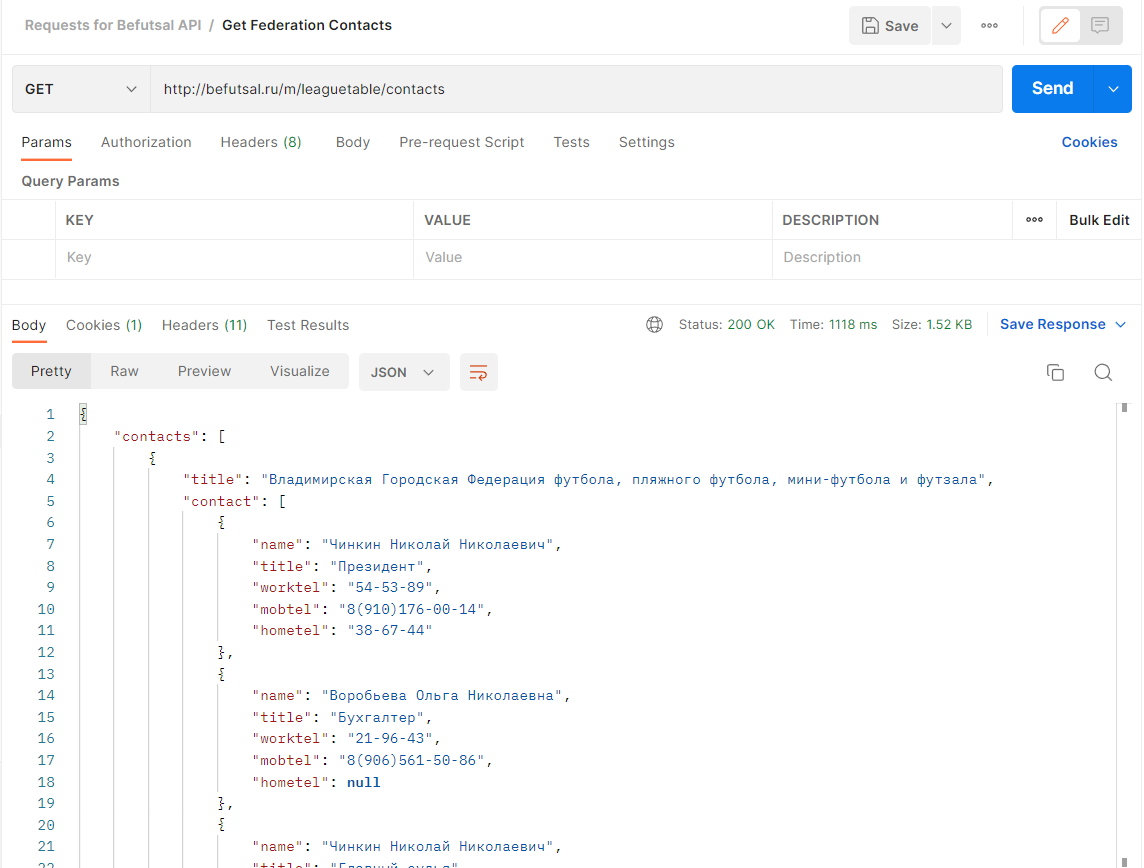


Рисунок 26. Фактический ответ на запрос (совпал с ожидаемым)

Мой сценарий 6 – Зарегистрироваться, войти, посмотреть протоколы команды, подробно посмотреть определенный протокол, выйти с аккаунта

В этот сценарий входит следующая последовательность запросов:

- Регистрация (/register - post-request)

Обязательные параметры для минимального запроса:

- username – string

- email – string

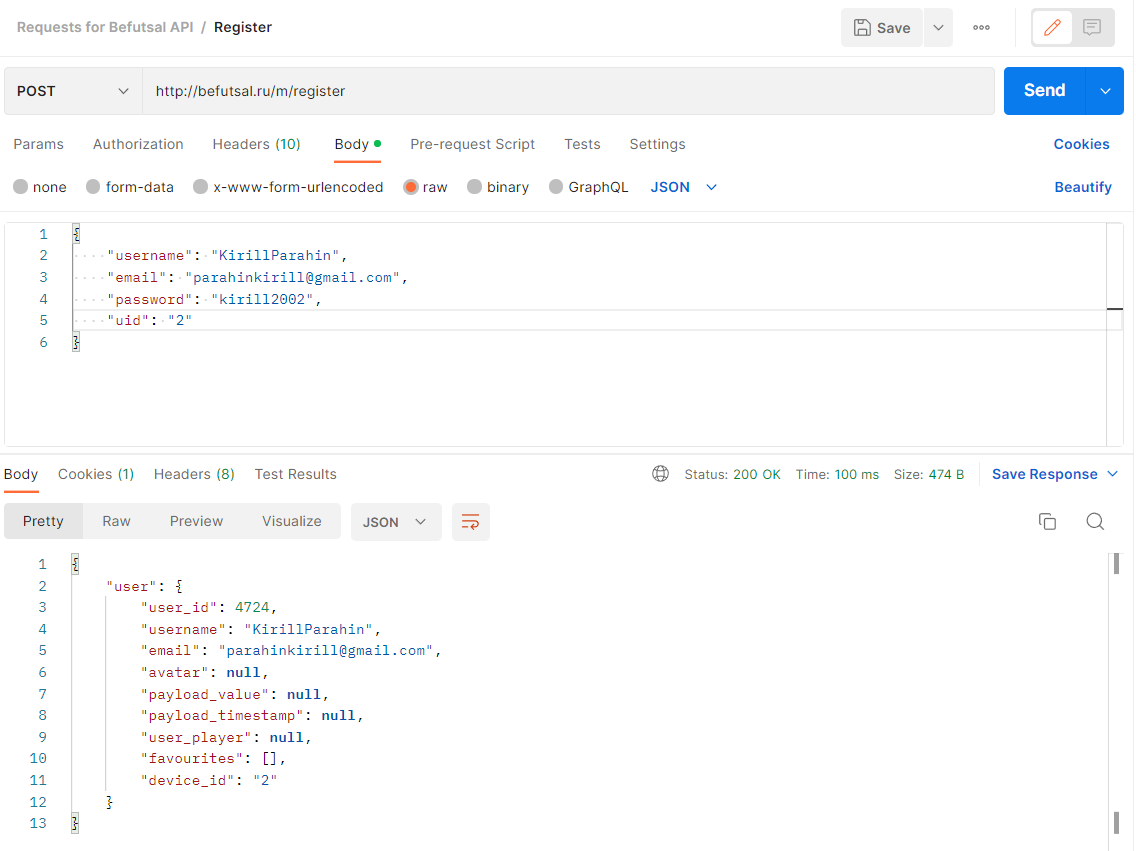
- password – string

- uid – string

В случае отсутствия ввода хотя бы одного параметра в запрос, мы получим ответ с ошибкой 400 – Bad Request (Некорректный запрос)

Ниже представлен пример минимального запроса и ответа на него (рис.27):

Ожидаемый ответ от запроса: HTTP 200 Ok

Рисунок 27. Фактический ответ на запрос (совпал с ожидаемым)

- Войти (/login – post-request)

Обязательные параметры для минимального запроса:

- email – string

- password – string

Ниже представлен пример минимального запроса и ответа на него (рис.28):

Ожидаемый ответ от запроса: HTTP 200 Ok

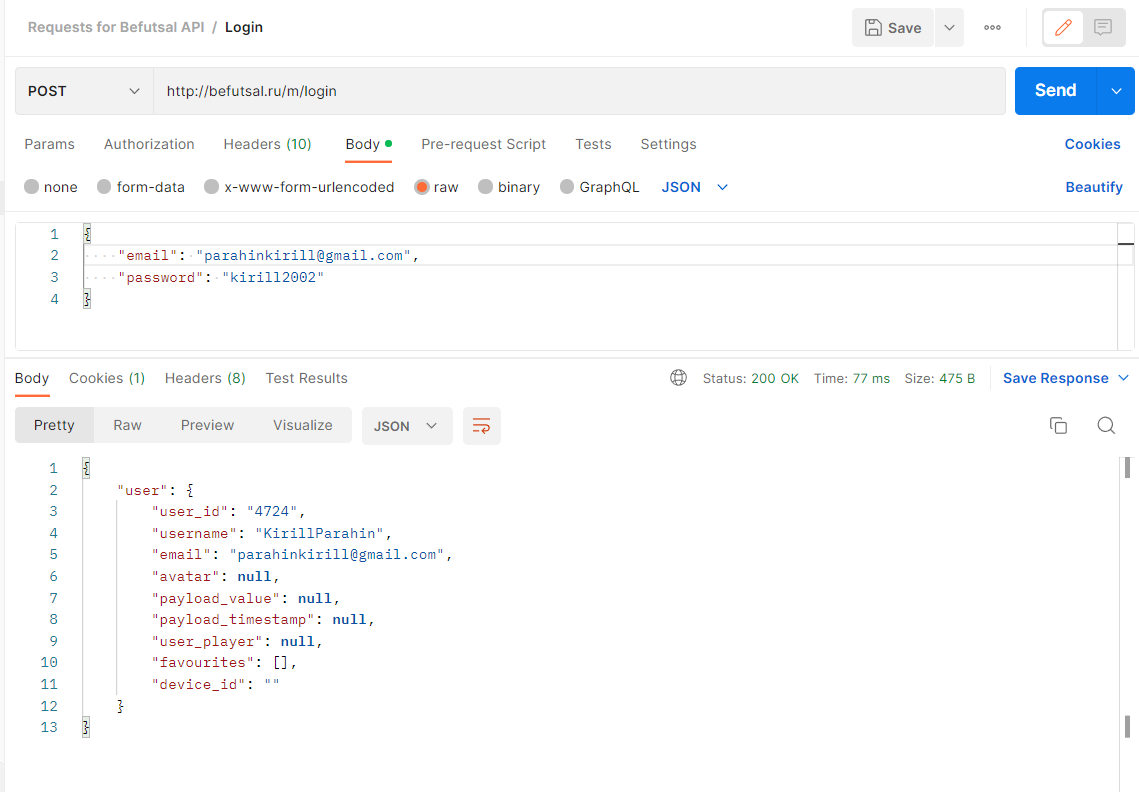


Рисунок 28. Фактический ответ на запрос (совпал с ожидаемым)

- Посмотреть протоколы команды (/protocols – get-request)

Обязательные параметры для минимального запроса:

- team\_id - integer

Ниже представлен пример минимального запроса и ответа на него (рис.29):

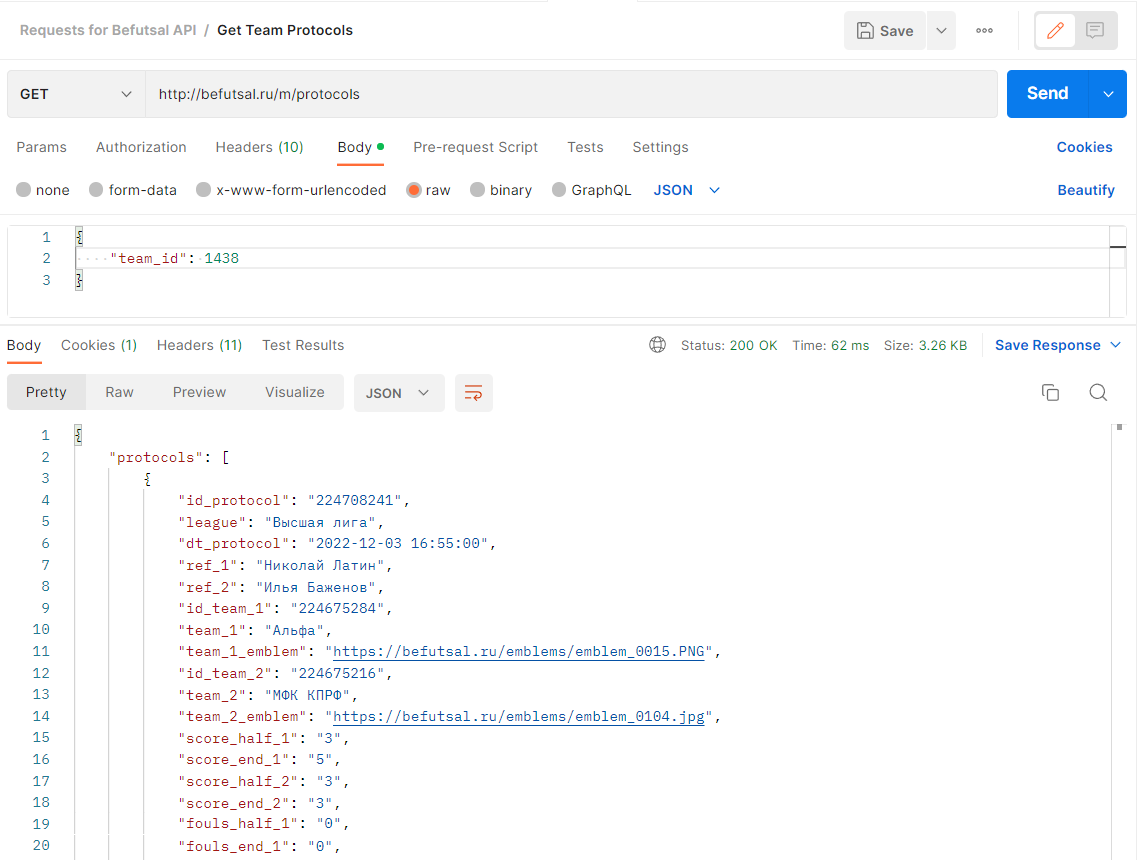


Рисунок 29. Фактический ответ на запрос (совпал с ожидаемым)

* Подробнее посмотреть конкретный протокол (/detailed\_protocol – get-request)
* Обязательные параметры для минимального запроса:

- protocol\_id - integer

Ниже представлен пример минимального запроса и ответа на него (рис.30):

Ожидаемый ответ от запроса: HTTP 200 Ok

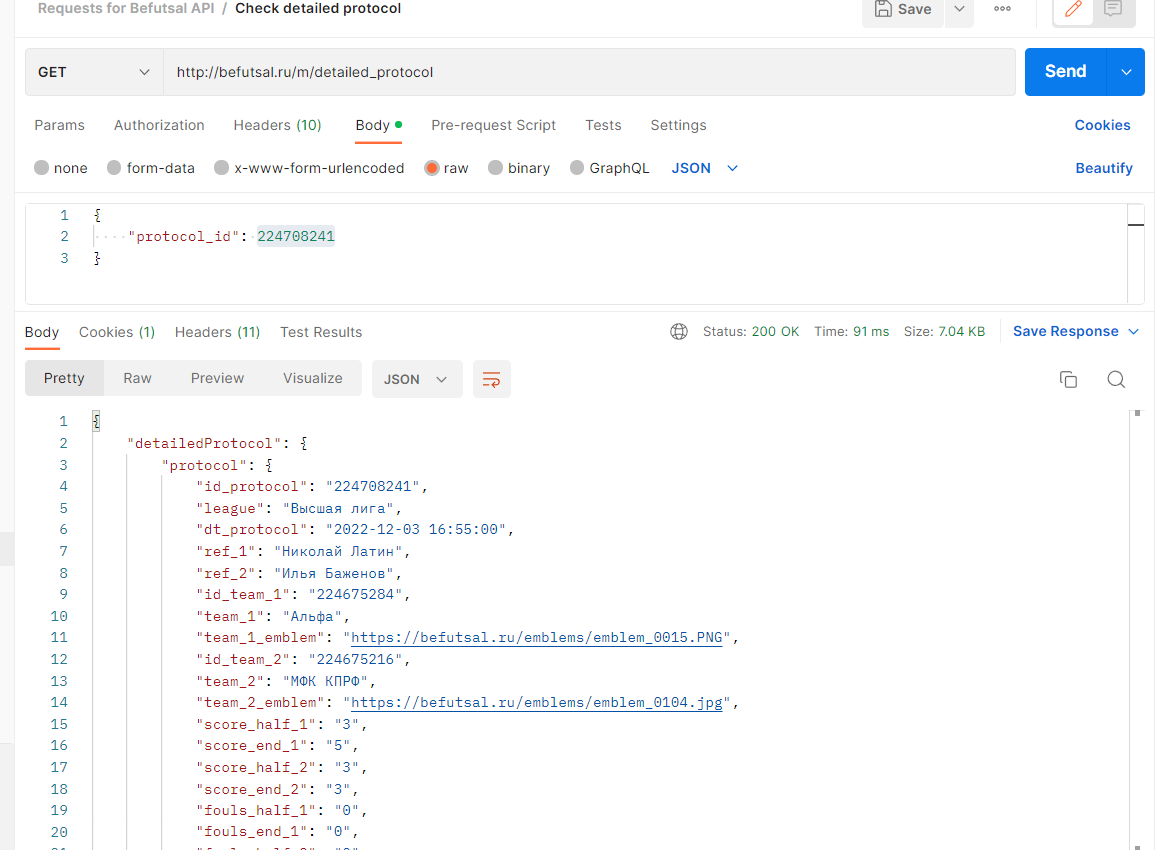


Рисунок 30. Фактический ответ на запрос (совпал с ожидаемым)

- Выйти с аккаунта (/logout – get-request)

Обязательные параметры отсутствуют – минимальный запрос не содержит параметров

Ниже представлен пример минимального Get-запроса с пустым телом и ответа на него (рис.31):

Ожидаемый ответ от запроса: HTTP 200 Ok

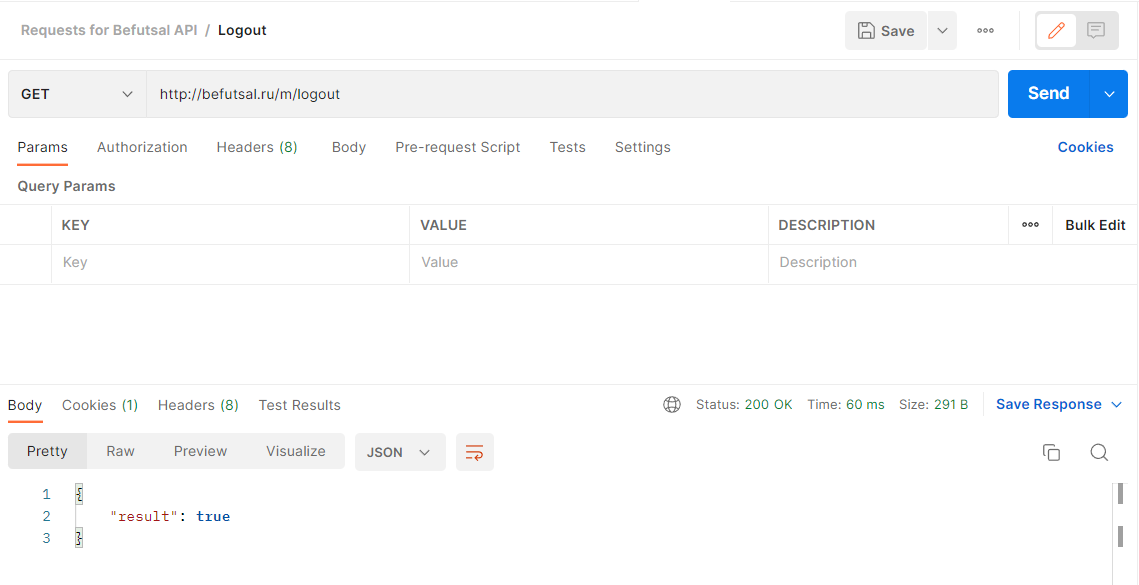


Рисунок 31. Фактический ответ на запрос (совпал с ожидаемым)

Вывод

В результате выполнения работы, я научился проектировать тестовые сценарии для тестирования REST API, а также познакомился с инструментом тестирования REST API - программой Postman.