# Руководство по использованию

# «Взаимодействие программ с WEB API Gwent»

Оглавление

[Общие правила работы с системой 3](#_Toc176173179)

[Схема работы интеграции 4](#_Toc176173180)

[Методы запросов 5](#_Toc176173181)

[Регистрация пользователя 6](#_Toc176173182)

[Логин 8](#_Toc176173190)

[Смена пароля 10](#_Toc176173198)

[Удаление пользователя 12](#_Toc176173206)

[Получение статистки пользователя 15](#_Toc176173214)

[Изменение/получение рейтинга пользователя 17](#_Toc176173222)

[Получение игрового профиля пользователя 19](#_Toc176173231)

[Присоединение к гильдии 21](#_Toc176173239)

[Выход из гильдии 24](#_Toc176173247)

[Проверка является ли пользователь администратором гильдии 27](#_Toc176173255)

[Получение персональной информации о пользователе 30](#_Toc176173263)

[Изменение страны проживания пользователя 32](#_Toc176173271)

[Изменение раздела “О себе” 34](#_Toc176173279)

[Изменение ссылки на Twitch пользователя 36](#_Toc176173287)

[Изменение возраста пользователя 38](#_Toc176173295)

[Создание гильдии 40](#_Toc176173303)

[Изменить описание гильдии 42](#_Toc176173311)

[Исключить члена гильдии 44](#_Toc176173319)

[Удаление гильдии 47](#_Toc176173327)

[Получение данных гильдии 49](#_Toc176173335)

# Общие правила работы с системой

1. Все запросы к системе должны осуществляться методами POST, GET, PATCH, DELETE.
2. Тип каждого запроса прописан в текущей документации.
3. Поля в поле запроса передаются либо в теле запроса(“[FromBody]”), либо в строке запроса(“[FromQuery]”). Тип для каждого запроса также указан в документации.
4. Ответ ожидается всегда в формате json с кодом HTTP 200.
5. Для каждого запроса в ответе передается внутренний код состояния запроса и сообщение к нему. Все коды состояний описаны для каждого запроса.
6. Если вы устанавливаете систему у себя, то перед началом работы необходимо установить PostgreSQL и в программе выполнить миграцию базы данных.

# Схема работы интеграции

Тут надо картинку

**Рис.1.** Схема работы интеграции

ПОМЕНЯТЬ!!!

Поскольку обмен информацией двухсторонний то на стороне «Внешней системы» должен иметься интерфейс API для передачи изменяемой информации от системы «Буфер».

**REST API Шаблон построения на стороне внешней системы**

* Все параметры могут передаваться в теле запросов POST, PATCH, DELETE, либо в строке запроса GET.
* Названия параметров на английском языке с маленькой буквы. (пример: name, password, email, и т.д.).
* API должен передавать ответы на запросы в JSON формате.
* Все ошибки должны быть обработаны и ответ должен формироваться также в JSON.

Пример:

**{**

**"code": 401,**

**"message": "User not found"**

**}**

* Наличие ошибки проверяется по наличию параметра code c соответствующими значениями, хорошим правилом будет наличие code с значением 200 в случае успешного выполнения запроса.

# Методы запросов

## Регистрация пользователя

### Описание метода

Данный запрос реализует регистрацию пользователя в системе и добавление в базу данный. Все последующие методы возможны только для зарегистрированных пользователей.

### Request URL

/api/User/register

### Тип запроса

POST

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| email | string | Email пользователя |
| password | string | Пароль пользователя |
| confirmPassword | string | Подтверждение пароля |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Registration is successful | Регистрация прошла успешно |

**{**

**"code": 200,**

**"message": "Registration is successful"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Invalid credentails: this Username is already taken | Пользователь с таким Username'ом уже зарегистрирован |

**{**

**"code": 401,**

**"message": "Invalid credentails: this Username is already taken"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Invalid credentails: this Email is already taken | Пользователь с таким Email'ом уже зарегистрирован |

**{**

**"code": 402,**

**"message": "Invalid credentails: this Email is already taken"**

**}**

## Логин

### Описание метода

Данный запрос реализует логин пользователя в систему.

### Request URL

/api/User/login

### Тип запроса

POST

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| password | string | Пароль пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| token | string | Токен доступа для передачи данных для аутентификации пользователя |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Login successeful | Логин пользователя прошел успешно |

**{**

**"code": 200,**

**"message": "Login successeful"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Invalid credentails: incorrect login or password | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "Invalid credentails: incorrect login or password"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Invalid credentails: incorrect login or password | Неверный пароль |

**{**

**"code": 402,**

**"message": "Invalid credentails: incorrect login or password"**

**}**

## Смена пароля

### Описание метода

Данный запрос позволяет сменить пароль пользователя.

### Request URL

/api/User/changePassword

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| oldPassword | string | Старый пароль пользователя |
| newPassword | string | Новый пароль пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Password changed successfully | Пароль сменён на новый |

**{**

**"code": 200,**

**"message": "Password changed successfully"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User not found | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User not found"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Invalid credentails: incorrect password | Введён неправильный старый пароль |

**{**

**"code": 402,**

**"message": "Invalid credentails: incorrect password"**

**}**

## Удаление пользователя

### Описание метода

Данный запрос позволяет удалить пользователя из базы. Пользователь удаляется из всех таблиц базы данных, а также из списка членов гильдии, если он в таковой находится.

### Request URL

/api/User/deleteUser

### Тип запроса

DELETE

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| password | string | Старый пароль пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | User has been successfully deleted | Пользователь успешно удалён из базы |

**{**

**"code": 200,**

**"message": "User has been successfully deleted"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User not found | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User not found"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Invalid credentails: incorrect password | Введён неправильный пароль |

**{**

**"code": 402,**

**"message": "Invalid credentails: incorrect password"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 403 | User is admin of guild | Пользователь является администратором гильдии, из-за чего удаление невозможно |

**{**

**"code": 403,**

**"message": "User is admin of guild"**

**}**

## Получение статистки пользователя

### Описание метода

Данный метод позволяет получить игровую статистику пользователя.

### Request URL

/api/Statistics/getStatistics?UserName=...

### Тип запроса

GET

### Атрибут запроса

Из строки – [FromQuery]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| totalGames | Int | Общее количество игр пользователя |
| wins | Int | Количество побед пользователя |
| defeats | Int | Количество поражений пользователя |
| draws | int | Количество ничьих игр пользователя |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь найден и его статистика успешно получена |

**{**

**"code": 200,**

**"message": "Success",**

**"totalGames": 4,**

**"wins": 2,**

**"defeats": 1,**

**"draws": 1**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User not found | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User not found",**

**"totalGames": 0,**

**"wins": 0,**

**"defeats": 0,**

**"draws": 0**

**}**

## Изменение/получение рейтинга пользователя

### Описание метода

Данный метод позволяет получить игровой рейтинг пользователя, например для вывода в игровой профиль, либо для внесения изменений после завершения игры, в таком случае изменения автоматически вносятся в статистику игр пользователя и в общий рейтинг гильдии, если пользователь принадлежит к таковой. Всем пользователям сразу после регистрации автоматически присваивается рейтинг 1000 условных единиц.

### Request URL

/api/Rating/rating

### Тип запроса

POST

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| status | GameStatus | Определяет взаимодействие с рейтингом |

### GameStatus

* 0 (Victory) – победа в игре, увеличение игрового рейтинга
* 1 (Defeat) – поражение в игре, уменьшение игрового рейтинга
* 2 (Draw) – ничья в игре
* 3 (Show) – получение игрового рейтинга

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| rating | string | Игровой рейтинг пользователя |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь найден и его рейтинг успешно получен или изменён |

**{**

**"code": 200,**

**"message": "Success",**

**"rating": "1049"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | No user with this username was found | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "No user with this username was found",**

**"rating": null**

**}**

## Получение игрового профиля пользователя

### Описание метода

Данный метод позволяет получить информацию об игровом профиле пользователя.

### Request URL

/api/Statistics/getStatistics?UserName=...

### Тип запроса

GET

### Атрибут запроса

Из строки – [FromQuery]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| email | string | Email пользователя |
| rating | string | Игровой рейтинг пользователя |
| wins | string | Количество побед пользователя |
| guildName | string | Название гильдии, к которой принадлежит пользователь. Или “-“, если таковой нет. |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь найден, и информация о нём успешно получена |

**{**

**"code": 200,**

**"message": "Success",**

**"email": "user@example.com",**

**"rating": "1049",**

**"wins": "2"**

**“guildName”: "name"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist",**

**"email": "false",**

**"rating": "false",**

**"wins": "false"**

**“guildName”: "false"**

**}**

## Присоединение к гильдии

### Описание метода

Данный метод позволяет пользователю присоединиться к существующей гильдии. Данные гильдии автоматически обновляются.

### Request URL

/api/Profile/JoinGuild

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| userName | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь и гильдия найдены, пользователь успешно присоединился к гильдии |

**{**

**"code": 200,**

**"message": "Success",**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | No such user | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "No such user",**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | No such guild | Гильдия с таким именем не найдена |

**{**

**"code": 200,**

**"message": "No such guild",**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 403 | Guild is already maxed out | В гильдии уже максимальное количество участников |

**{**

**"code": 403,**

**"message": "Guild is already maxed out",**

**}**

|  |  |  |
| --- | --- | --- |
| Code | message | Описание |
| 404 | User already has a guild | Пользователь уже находится в гильдии |

**{**

**"code": 404,**

**"message": "User already has a guild",**

**}**

## Выход из гильдии

### Описание метода

Данный метод позволяет пользователю выйти из текущей гильдии. Данные гильдии автоматически обновляются.

### Request URL

/api/Profile/ResignGuild

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| userName | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь и гильдия найдены, пользователь успешно вышел из гильдии |

**{**

**"code": 200,**

**"message": "Success",**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | No such user | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "No such user",**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | No such guild | Гильдия с таким именем не найдена |

**{**

**"code": 200,**

**"message": "No such guild",**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 403 | Single member user | Пользователь является единственным участником (администратором). В этом случае необходимо удалить гильдию |

**{**

**"code": 403,**

**"message": "Single member user",**

**}**

|  |  |  |
| --- | --- | --- |
| Code | message | Описание |
| 404 | User isn't member of this guild | Пользователь не принадлежит к указанной гильдии |

**{**

**"code": 404,**

**"message": "User isn't member of this guild",**

**}**

## Проверка является ли пользователь администратором гильдии

### Описание метода

Данный метод позволяет узнать является ли пользователь администратором гильдии или нет. Может использоваться, например для отображения “Изменения описания гильдии” или “Удаления гильдии”.

### Request URL

/api/Profile/IsAdminOfGuild?GuildName=...&UserName=...

### Тип запроса

GET

### Атрибут запроса

Из строки – [FromQuery]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| userName | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 201 | Admin | Пользователь является администратором указанной гильдии |

**{**

**"code": 201,**

**"message": "Admin"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 202 | Not admin | Пользователь не является администратором указанной гильдии |

**{**

**"code": 202,**

**"message": "Not admin"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | No such user | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "No such user"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | No such guild | Гильдия с таким именем не существует |

**{**

**"code": 402,**

**"message": "No such guild"**

**}**

## Получение персональной информации о пользователе

### Описание метода

Данный метод позволяет получить персональную информацию о пользователе. Все данные пользователь предоставляет сам по своему желанию.

### Request URL

/api/Info/getInfo?UserName=...

### Тип запроса

GET

### Атрибут запроса

Из строки – [FromQuery]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| country | string | Страна проживания пользователя |
| twitchLink | string | Ссылка на ресурс “Twitch” пользователя |
| age | Int? | Возраст пользователя |
| about | string | Раздел “О себе” |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Пользователь найден, и информация о нём успешно получена |

**{**

**"code": 200,**

**"message": "Success",**

**"country": "-",**

**"twitchLink": "-",**

**"age": null,**

**"about": ""**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist",**

**"country": "false",**

**"twitchLink": "false",**

**"age": -1,**

**"about": "false"**

**}**

## Изменение страны проживания пользователя

### Описание метода

Данный метод позволяет изменить страну проживания пользователя. Данные обновляются в разделе с персональной информацией о пользователе.

### Request URL

/api/Country/country

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| country | string | Страна проживания |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Страна успешно обновлена на новую |

**{**

**"code": 200,**

**"message": "Success"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist"**

**}**

## Изменение раздела “О себе”

### Описание метода

Данный метод позволяет изменить данные в разделе “О себе”. Данные обновляются в разделе с персональной информацией о пользователе.

### Request URL

/api/About/setAbout

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| about | string | Информация о пользователе |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Информация успешно обновлена на новую |

**{**

**"code": 200,**

**"message": "Success"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist"**

**}**

## Изменение ссылки на Twitch пользователя

### Описание метода

Данный метод позволяет изменить ссылку на Twitch канал пользователя. Данные обновляются в разделе с персональной информацией о пользователе.

### Request URL

/api/Twitch/setTwitch

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| twitchLink | string | Ссылка на Twitch канал пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Twitch ссылка успешно обновлена на новую |

**{**

**"code": 200,**

**"message": "Success"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist"**

**}**

## Изменение возраста пользователя

### Описание метода

Данный метод позволяет изменить возраст пользователя. Данные обновляются в разделе с персональной информацией о пользователе.

### Request URL

/api/Age/setAge

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| userName | string | Username пользователя |
| age | int | Возраст пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 200 | Success | Возраст успешно обновлен |

**{**

**"code": 200,**

**"message": "Success"**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | User doesn't exist | Пользователь с таким Username'ом не существует |

**{**

**"code": 401,**

**"message": "User doesn't exist"**

**}**

## Создание гильдии

### Описание метода

Данный метод позволяет создать гильдию. Пользователь автоматически становится её администратором. Все данные гильдии автоматически получают актуальные значения.

### Request URL

/api/Guild/RegisterGuild

### Тип запроса

POST

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название создаваемой гильдии |
| userName | string | Username пользователя |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| сode | message | Описание |
| 200 | Guild created successful | Гильдия успешно создана |

**{**

**"code": 200,**

**"message": "Guild created successful”**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Guild by that name already exists | Гильдия с таким названием уже существует |

**{**

**"code": 401,**

**"message": "Guild by that name already exists"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | User is already a member of a guild | Этот пользователь уже находится в гильдии |

**{**

**"code": 402,**

**"message": "User is already a member of a guild"**

**}**

## Изменить описание гильдии

### Описание метода

Данный метод позволяет изменить описание гильдии.

### Request URL

/api/Guild/EditDescription

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| description | string | Строка с описанием, на которое будет изменено старое |
| userName | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| сode | message | Описание |
| 200 | Success | Описание успешно изменено на новое |

**{**

**"code": 200,**

**"message": "Success”**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Such guild doesn't exist | Гильдия с таким названием не существует |

**{**

**"code": 401,**

**"message": "Such guild doesn't exist"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Not enough rights | Пользователь не является администратором гильдии и у него недостаточно прав для внесения изменений |

**{**

**"code": 402,**

**"message": "Not enough rights"**

**}**

## Исключить члена гильдии

### Описание метода

Данный метод позволяет администратору исключить члена из состава гильдии. Данные гильдии и профиля исключаемого пользователя автоматически обновляются.

### Request URL

/api/Guild/RemoveMember

### Тип запроса

PATCH

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| userToDelete | string | Username исключаемого пользователя |
| sender | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| сode | message | Описание |
| 200 | Success | Пользователь исключен |

**{**

**"code": 200,**

**"message": "Success”**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Such guild doesn't exist | Гильдия с таким названием не существует |

**{**

**"code": 401,**

**"message": "Such guild doesn't exist"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Not enough rights | Пользователь не является администратором гильдии и у него недостаточно прав для внесения изменений |

**{**

**"code": 402,**

**"message": "Not enough rights"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 403 | No such user | Пользователь с таким Username’ом не существует |

**{**

**"code": 403,**

**"message": "No such user"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 404 | No such member in guild | В указанной гильдии такого пользователя не найдено |

**{**

**"code": 404,**

**"message": "No such member in guild"**

**}**

## Удаление гильдии

### Описание метода

Данный метод позволяет администратору удалить гильдию. У всех участников автоматически обновляются данные профиля.

### Request URL

/api/Guild/DeleteGuild

### Тип запроса

DELETE

### Атрибут запроса

Из тела – [FromBody]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |
| sender | string | Username пользователя, отправляющего запрос |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| сode | message | Описание |
| 200 | Success | Гильдия успешно удалена |

**{**

**"code": 200,**

**"message": "Success”**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Such guild doesn't exist | Гильдия с таким названием не существует |

**{**

**"code": 401,**

**"message": "Such guild doesn't exist"**

**}**

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 402 | Not enough rights | Пользователь не является администратором гильдии и у него недостаточно прав для внесения изменений |

**{**

**"code": 402,**

**"message": "Not enough rights"**

**}**

## Получение данных гильдии

### Описание метода

Данный метод позволяет администратору удалить гильдию. У всех участников автоматически обновляются данные профиля.

### Request URL

/api/Guild/GetGuild?GuildName=...

### Тип запроса

GET

### Атрибут запроса

Из строки – [FromQuery]

### Передаваемые параметры

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| guildName | string | Название гильдии |

### Параметры ответа

|  |  |  |
| --- | --- | --- |
| Имя | Тип данных | Описание |
| code | Int | Код состояния обработки запроса |
| message | string | Поясняющее сообщение к коду |
| guildName | string | Название гильдии |
| countOfMembers | int | Текущее количество участников в гильдии |
| totalRating | int | Суммарный рейтинг участников гильдии |
| leaderName | string | Username администратора гильдии |
| members | List<string> | Список Username'ов участников гильдии |

### Типы ответа

#### В случае успеха

|  |  |  |
| --- | --- | --- |
| сode | message | Описание |
| 200 | Success | Гильдия успешно удалена |

**{**

**"code": 200,**

**"message": "Success",**

**"guildName": "guildName",**

**"countOfMembers": 3,**

**"totalRating": 3000,**

**"leaderName": "user02",**

**"members": [**

**"user02",**

**"user01",**

**"user03"**

**]**

**}**

#### В случае неудачи

|  |  |  |
| --- | --- | --- |
| code | message | Описание |
| 401 | Guild doesn't exist | Гильдия с таким названием не существует |

**{**

**"code": 401,**

**"message": "Guild doesn't exist",**

**"guildName": "false",**

**"countOfMembers": 0,**

**"totalRating": 0,**

**"leaderName": "false",**

**"members": []**

**}**