## ГАЙД ПО API

# **Введение**

API новостного сервиса МГТУ предоставляет доступ к различным ресурсам, таким как новости, статьи, события и объявления, публикуемые университетом. С его помощью вы можете получать актуальные данные, фильтровать их по различным параметрам, а также создавать и обновлять контент.

# **Аутентификация**

Bearer аутентификация — это метод аутентификации, используемый в API, при котором клиентское приложение отправляет токен доступа в заголовке HTTP-запроса для аутентификации и авторизации. Этот токен, известный как "Bearer token", передается в заголовке Authorization и подтверждает, что клиент имеет право доступа к запрашиваемому ресурсу.

В таблице запросов будет указано необходим ли Access Token

# **Сортировка**

В Laravel сортировка данных через API может быть реализована с использованием возможностей Eloquent и Query Builder.

В таблице запросов будет указано существует ли возможность сортировки данных и какие параметры можно передавать для сортировки.

# **Эндпоинты**

### AUTH

1. URL: '/auth/register'

Метод – POST

Параметры запроса:

* 'name'(Обязяательное поле) – Имя пользователя (Минимум 2 символа. Максимум 100 символов)
* 'surname'(Обязательное поле) – Фамилия пользователя (Минимум 2 символа. Максимум 100 символов)
* 'lastname' = Отчество пользователя (Максимум 100 символов)
* 'username'(Обязательное поле) – Юзер нейм (Английские символы и числа. Должно быть уникальным)
* 'email'(Обязательное поле) – Почта пользователя (Должна быть написана в формате почты. Должна быть уникальной
* 'specialization\_id'(Обязательное поле) – id специальности пользователя
* 'position\_id'(Обязательное поле) – Должность пользователя в университете
* 'password'(Обязательное поле) – Пароль пользователя
* 'password\_confirmation'(Обязательное поле) – Подтверждение пароля. (Должнен совпадать с параметром ‘Password’)

1. URL: '/auth/login'

Метод – POST

Параметры запроса:

* 'email' (Обязательное поле) – Эл. почта пользователя (Должна быть написана в формате почты. Должна быть уникальной)
* 'password' (Обязательное поле) – Пароль пользователя

1. URL: '/auth/logout'

Метод: POST

Параметры запроса: Нет

### USER

1. URL: '/profile/updateUserData'

Метод – PATCH

Параметры запроса:

* 'name'(Обязяательное поле) – Имя пользователя (Минимум 2 символа. Максимум 100 символов)
* 'surname'(Обязательное поле) – Фамилия пользователя (Минимум 2 символа. Максимум 100 символов)
* 'lastname' = Отчество пользователя (Максимум 100 символов)
* 'username'(Обязательное поле) – Юзер нейм(Английские символы и числа. Должно быть уникальным)
* 'email'(Обязательное поле) – Почта пользователя (Должна быть написана в формате почты. Должна быть уникальной
* 'specialization\_id'(Обязательное поле) – id специальности пользователя
* 'position\_id'(Обязательное поле) – Должность пользователя в университете

1. URL: '/profile/getProfileData'

Метод – GET

Параметры запроса:

* 'user\_id'(Обязяательное поле) – id пользователя (только целое число)

1. URL: ' /profile/getUserPosts'

Метод – GET

Параметры запроса:

* 'user\_id'(Обязяательное поле) – id пользователя (только целое число)

1. URL: '/profile/getUserLikedPosts'

Метод – GET

Параметры запроса:

* 'user\_id'(Обязяательное поле) – id пользователя (только целое число)

1. URL: /profile/updateUserPassword

Метод – PATCH

Параметры запроса:

* 'old\_password' (Обязательное поле) – Старый пароль пользователя
* 'password'(Обязательное поле) – Пароль пользователя
* 'password\_confirmation'(Обязательное поле) – Подтверждение пароля. (Должнен совпадать с параметром ‘Password’)

### POSTS

1. URL: '/posts/createPost'

Метод – POST

Параметры запроса:

* 'post\_title' (Обязательное поле) – Заголовок поста (Минимум 2 символа. Максимум 300 символов)
* 'post\_description' – Краткое описание поста (Минимум 10 символов. Максимум 500 символов)
* 'post\_text' (Обязательное поле) – Текст поста (неограниченное число символов)
* 'type\_id' (Обязательное поле) – Тип создаваемого поста

1. URL: '/posts/updatePost'

Метод – PATCH

Параметры запроса:

* 'post\_id' (Обязательное поле) – id изменяемого поста
* 'post\_title' (Обязательное поле) – Заголовок поста (Минимум 2 символа. Максимум 300 символов)
* 'post\_description' – Краткое описание поста (Минимум 10 символов. Максимум 500 символов)
* 'post\_text' (Обязательное поле) – Текст поста (неограниченное число символов)
* 'type\_id' (Обязательное поле) – Тип создаваемого поста

1. URL: '/posts/deletePost'

Метод – DELETE

Параметры запроса:

* 'post\_id' (Обязательное поле) – id удаляемого поста

1. URL: '/posts/getPost'

Метод – GET

Параметры запроса:

* 'post\_id' (Обязательное поле) – id искомого поста

1. URL: '/posts/getAllPosts'

Метод – GET

Параметры запроса: Нет

### COMMENT

1. URL: '/comments/createComment'

Метод – POST

Параметры запроса:

* 'comment\_text' (Обязательное поле) – Текст комментария (неограниченное число символов)
* 'post\_id' (Обязательное поле) – id поста, к которому пишется комментарий

1. URL: '/comments/updateComment'

Метод – PATCH

Параметры запроса:

* 'comment\_id' (Обязательное поле) – id изменяемого комментария
* 'comment\_text' (Обязательное поле) – Текст комментария (неограниченное число символов)

1. URL: '/comments/deleteComment'

Метод – DELETE

Параметры запроса:

* 'comment\_id' (Обязательное поле) – id удаляемого комментария

1. URL: '/comments/getComment'

Метод – GET

Параметры запроса:

* 'comment\_id' (Обязательное поле) – id искомого комментария

1. URL: '/comments/getComments'

Метод – GET

Параметры запроса:

* 'post\_id' (Обязательное поле) – id поста, у которого нужны комментарии

### LIKES

1. URL: '/likes/addLikeToPost'

Метод – POST

Параметры запроса:

* 'post\_id' (Обязательное поле) – id поста, на который ставится лайк

1. URL: '/likes/getPostLikes'

Метод – GET

Параметры запроса:

* 'post\_id' (Обязательное поле) – id поста, у которого нужны лайки

1. URL: '/likes/deletePostLike'

Метод – DELETE

Параметры запроса:

* 'post\_id' (Обязательное поле) – id поста, у которого нужно удалить лайк