API – документация

1. /user

* POST /registration – регистрация нового пользователя

Значение поля is\_superuser ставится false по умолчанию. Пароль хэшируется. Если в запросе нет одного из полей или пользователь с таким email уже существует, то возвращаются соответствующие ошибки с кодом 404. При успешном запросе возвращается токен, в котором находится информация о user\_id, email и is\_superuser. Он валиден 24 часа и используется для проверки аутентификации пользователя.

Все следующие запросы, в которых требуется авторизация, передают в header токен. Если токен не валиден, то возвращается ошибка 403 с сообщением «Не авторизован».

Body:

|  |
| --- |
| username |
| password |
| email |

Выходные данные:

|  |
| --- |
| token |

Ошибки:

|  |  |
| --- | --- |
| 404 | Заполните все поля |
| 404 | Пользователь с таким email не существует |

* POST /login – вход пользователя.

Body:

|  |
| --- |
| email |
| password |

Выходные данные:

|  |
| --- |
| token |

Ошибки:

|  |  |
| --- | --- |
| 404 | Заполните все поля |
| 500 | Неверный пароль |
| 404 | Пользователь с таким email не существует |

* GET /auth – проверяет авторизирован ли пользователь. В header передается токен. Если он валиден, генерируется новый токен.

Выходные данные:

|  |
| --- |
| token |

Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |

* PUT / - назначает обычного пользователя администратором.

Доступна только администратору. Если проверка не прошла, возвращается одна из ошибок 401 или 403 с соответствующим сообщением. Это справедливо для всех следующих запросов, в которых требуются права администратора. На выходе получает обновленную информацию о пользователе.

Body:

|  |
| --- |
| username |
| password |
| email |

Выходные данные:

|  |
| --- |
| user\_id |
| username |
| password |
| email |
| is\_superuser |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

* GET / - возвращает список emailов обычных пользователей (is\_superuser=false). Доступна только администраторам.

Пример выходных данных:

[

{ “email”: “email1@mail.com” },

{ “email”: “email2@mail.com” },

…

]

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

2. /serial

* GET / - возвращает список всех сериалов.

На вход подаются необязательные параметры: жанр, страна, страница, количество сериалов на странице. В зависимости от запроса выдается результат: если не введены жанр и/или страна, сериалы по этим параметрам не фильтруются. В качестве дефолтных значений для страницы и количества сериалов на странице устанавливается 1 и 9.

Входные данные:

|  |
| --- |
| genre\_id |
| country\_id |
| limit |
| page |

* GET /:id – возвращает информацию о сериале по id

Params:

|  |
| --- |
| id |

* POST / - добавляет сериал в список. Доступна только администраторам

Body:

|  |
| --- |
| name |
| description |
| content |
| genre\_id |
| status |
| number\_of\_seasons |
| number\_of\_episodes |
| duration\_of\_episode |
| country\_id |
| rating\_imdb |
| release\_year |
| main\_actors |
| resource\_url |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

* PUT /:id – обновляет информацию о сериале по id. Доступна только администраторам

Params:

|  |
| --- |
| id |

Body:

|  |
| --- |
| number\_of\_seasons |
| number\_of\_episodes |
| rating\_imdb |
| release\_year |
| main\_actors |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

3. /reviews

* GET /:id - возвращает список всех отзывов к сериалу по его id.

Params:

|  |
| --- |
| id |

* POST /:id – добавляет отзыв к сериалу с serial\_id=id. Доступна только авторизированным пользователям.

Params:

|  |
| --- |
| id |

Body:

|  |
| --- |
| text\_rew |
| rating |

Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |

* PUT /:id – редактирует текст отзыва. Доступна только администраторам. В header – id отзыва.

Params:

|  |
| --- |
| id |

Body:

|  |
| --- |
| text\_rew |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

* DELETE /:id – удаляет отзыв по его id. Доступна только администраторам.

Params:

|  |
| --- |
| id |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

4. /genre

* GET / - возвращает список всех жанров
* POST / - создает новый жанр. Доступна только администраторам

Body:

|  |
| --- |
| name |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

5. /country

* GET / - возвращает список всех стран
* POST / - создает новую страну. Доступна только администраторам

Body:

|  |
| --- |
| name |

Ошибки:

|  |  |
| --- | --- |
| 401 | Не авторизован |
| 403 | Нет доступа |

6. /userlists

* GET / - возвращает список сериалов, добавленных пользователем в конкретный список: «Посмотрел», если is\_viewed=1, «Хочу посмотреть», если is\_viewed=0. Доступна только авторизированным пользователям

Входные данные:

|  |
| --- |
| is\_viewed |

* Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |

* GET /check – проверяет, находится ли сериал с данным id в одном из списков пользователя. Если да, то возвращает массив с полем in\_list с номером списка, иначе в поле ставится -1. Доступна только авторизированным пользователям

Входные данные:

|  |
| --- |
| id |

Выходные данные:

|  |
| --- |
| in\_list |

Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |

* POST /:id – добавляет сериал с данным id в один из списков пользователя. Возвращает массив с полем in\_list с номером списка. Доступна только авторизированным пользователям

Params:

|  |
| --- |
| id |

Body:

|  |
| --- |
| is\_viewed |

Выходные данные:

|  |
| --- |
| in\_list |

Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |

* DELETE /:id – удаляет сериал с данным id из списка. Доступна только авторизированным пользователям

Params:

|  |
| --- |
| id |

Ошибки:

|  |  |
| --- | --- |
| 403 | Не авторизован |