

Предисловие

Эндпоинтов в API 4:

`/v1/users`

для регистрации новых пользователей

`/v1/users/<int:account_id>`

для просмотра информации о пользователе

`/v1/users/<int:account_id>/jokes`

для просмотра и изменения списка шуток

`/v1/users/<int:account_id>/jokes/<int:joke_id>`

для просмотра, изменения и удаления конкретных шуток

Модель аутентификации – HTTP Basic authentication, далеко не самая безопасная, но в моем варианте при регистрации пароль передается в json в plain text, так что Digest или OAuth не сделают меня безопасней :-)

При присвоении id шуткам нумерация сквозная, т. е. шутка с id 2 будет всего одна у одного пользователя, а не у каждого своя. С одной стороны, пришлось разбирать случаи, когда пользователь хочет посмотреть свою шутку, но, например, указывает чужой аккаунт, с другой, так у меня в системе у каждой шутки свой уникальный адрес.

Jokes API 1.0.0

[Base URL: localhost/v1]

/users

POST

регистрирует нового пользователя

body

```
{
  "username": "myUsername",
  "password": "myPassword"
}
```

response

```
{
  "id": "newUserId"
}
```

201 успешно
400 не указаны поля username или password
409 пользователь с таким именем уже существует

/users/{userId}

GET

возвращает пользователя по userId

* необходима авторизация

response

```
{
  "username": "myUsername"
}
```

200 успешно
401 при попытке посмотреть без авторизации
403 при успешной авторизации и попытке посмотреть другого/несуществующего пользователя

/user/{userId}/jokes

GET

возвращает список шуток пользователя по userId

* необходима авторизация

response

```
[
  {
    "joke": {
      "id": 1,
      "text": "joke text",
      "user_id": 1
    }
  }
]
```

200 успешно

401 при попытке посмотреть без авторизации

403 при успешной авторизации и попытке посмотреть другого/несуществующего пользователя

POST

генерирует новую уникальную шутку

* необходима авторизация

response

```
{
  "joke": {
    "id": 1,
    "text": "joke text",
    "user_id": 1
  }
}
```

201 успешно

401 при попытке посмотреть без авторизации

403 при успешной авторизации и попытке посмотреть другого/несуществующего пользователя

500 проблемы с подключением к ресурсу geek-jokes

/user/{userId}/jokes/{jokeId}

GET

возвращает шутку по jokeId

* необходима авторизация

response

```
{
  "joke": {
    "id": 1,
    "text": "joke text",
    "user_id": 1
  }
}
```

- 200 успешно
- 401 при попытке посмотреть без авторизации
- 403 при попытке залезть в чужой аккаунт (даже если по jokeId шутка Ваша)
- 404 при просмотре несуществующих/чужих шуток в своем аккаунте

PUT

обновляет шутку по jokeId

* необходима авторизация

body

```
{
  "text": "jokeNewText",
}
```

- 200 успешно
- 400 отсутствует или пустой параметр "text"
- 401 при попытке посмотреть без авторизации
- 403 при изменить какую-либо шутку в чужом аккаунте (даже если по jokeId шутка Ваша)
- 404 при просмотре несуществующих/чужих шуток в своем аккаунте

DELETE

удаляет шутку по jokeId

* необходима авторизация

- 200 успешно
- 401 при попытке посмотреть без авторизации
- 403 при попытке залезть в чужой аккаунт (даже если по jokeId шутка Ваша)
- 404 при просмотре несуществующих/чужих шуток в своем аккаунте