

Backend documentation

Тихонов Кирилл

February 2024

1 Описание API

1.1 Home page (/)

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает HTML-страницу, содержащую главную страницу

1.2 /favicon.ico

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает иконку сайта, отображающуюся в вкладке браузера

1.3 /signup

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает HTML-страницу для регистрации

1.4 /quiz

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает HTML-страницу для регистрации

1.5 /signin

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает HTML-страницу для прохождения теста

1.6 /signin_error

- Тип запроса - GET
- Заголовки запроса: нет
- Тело запроса: нет
- Описание: возвращает HTML-страницу с сообщением об ошибке о неза-
регистрированности пользователя

1.7 /signin/check

- Тип запроса - POST
- Заголовки запроса: нет
- Тело запроса: name - имя пользователя, password - пароль пользовате-
ля и score - очки пользователя (нужно для корректной работы других
функций, настоятельно рекомендуется поставить значение 0)
- Описание: возвращает JSON, содержащий информацию о входе в ак-
каунт. Проверяет на наличие пользователя в базе, если его нет, то
в JSON'е будет поле present, равное нулю. Если пользователь есть, но
неверный пароль, то будет поле password, равное нулю. Если же все
верхнее верно, то будет поле token, содержащее новосгенерированный
для этого пользователя токен по алгоритму uuid4. Всегда присутству-
ет поле info, содержащее информацию об ошибке либо сообщение об
успехе входа в аккаунт.

1.8 /signup/check

- Тип запроса - POST
- Аргументы запроса: name - имя пользователя, password - пароль пользователя и score - очки пользователя (нужно для корректной работы других функций, настоятельно рекомендуется поставить значение 0)
- Описание: возвращает JSON, содержащий информацию о регистрации на сайте. Проверяет на наличие пользователя в базе, если он есть (уже зарегистрирован), то в JSON'е будет поле created, равное нулю. Если пользователь еще не зарегистрирован, то добавляет его в базу и в ответе будет поле created, равное единице. Также в JSON'е всегда будет поле info с информацией об успехе или ошибке при запросе.

1.9 /quiz/check

- Тип запроса - POST
- Заголовки запроса: token - токен пользователя, сгенерированный при входе в аккаунт (1.7)
- Тело запроса: список из 10 ответов, второй и шестой из которых состоят из списков
- Описание: возвращает JSON, содержащий информацию о результатах теста. Проверяет пользователя на наличие токена, если токена нет (пользователь не вошел в аккаунт), то в JSON'е будет поле signin, равное единице. Если пользователь отправил валидный токен, то будет поле score, равное набранным баллам (по десятибалльной системе).

1.10 /results

- Тип запроса - GET
- Заголовки запроса: token - токен пользователя, сгенерированный при входе в аккаунт (1.7)
- Тело запроса: нет
- Возвращает HTML страницу. Если токен невалиден (пользователь незарегистрирован), то возвращает HTML-страницу с сообщением о незарегистрированности (1.6). Если токен валиден, то возвращает HTML-страницу с таблицей результатов.

1.11 Примечания

- Если пользователь зарегистрирован, но не проходил тест, то его результат **будет** отображаться в табличке и будет равен нулю
- Порт сайта указывается в конце `main.ru`, но не изменяется автоматически в JavaScript (`/static/js/script.js`) файле в `fetch`-запросах