**Оглавление**

[Позитивные тесты 2](#__RefHeading___Toc2495_2532478993)

[Управление аккаунтом 2](#__RefHeading___Toc2603_2532478993)

[Управление контентом 2](#__RefHeading___Toc2605_2532478993)

[Негативные тесты 4](#__RefHeading___Toc2497_2532478993)

[Управление аккаунтом 4](#__RefHeading___Toc2607_2532478993)

[Тестовый набор «Создание пользователя» 4](#__RefHeading___Toc2640_2532478993)

[Тестовый набор «Логин» 5](#__RefHeading___Toc2642_2532478993)

[Тестовый набор «Обновление пароля пользователя» 6](#__RefHeading___Toc2644_2532478993)

[Управление контентом 8](#__RefHeading___Toc2609_2532478993)

[Тестовый набор «Создать памятку» 8](#__RefHeading___Toc2646_2532478993)

[Тестовый набор «Получить памятку» 9](#__RefHeading___Toc2648_2532478993)

[Тестовый набор «Редактировать памятку» 10](#__RefHeading___Toc2650_2532478993)

[Тестовый набор «Удалить памятку» 11](#__RefHeading___Toc2652_2532478993)

[Тестовый набор «Получить все памятки пользователя» 13](#__RefHeading___Toc2654_2532478993)

[Повторное выполнение негативных тест-кейсов из разделов «Управление аккаунтом», «Управление контентом». 14](#__RefHeading___Toc2660_2532478993)

# Позитивные тесты

## Управление аккаунтом

1. Создание пользователя:

* Отправить запрос на сервер: POST /users {«username»: «Alex», «password1»: «Race2021», «password2»: «Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 200(OK).

2. Логин (Аутентификация):

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password1»: Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 200 и JWT-токен.

3. Обновление пароля пользователя:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
* Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «Race2021», «password1»: «Amusement20», «password2»: «Amusement20»}.
* Ожидаемый результат: сервер присылает ответ со статусом 200(OK).
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
* Ожидаемый результат: сервер присылает ответ со статусом 200 и JWT-токен.

## Управление контентом

4. Создать памятку:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
* Отправить запрос на сервер: POST /todos/< «username»: «Alex»>/<todo\_id: 1> {«text»: «Buy some fruits» , «status»: «TODO»}.
* Ожидаемый результат: сервер присылает ответ со статусом 200(OK).

5. Получить памятку:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
* Отправить запрос на сервер: GET /todos/<username: «Alex»>/<todo\_id: 1>
* Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и передает запрошенную памятку в теле ответа.

6. Редактировать памятку:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> «Buy some fruits».
* Отправить запрос на сервер: PUT /todos/<username: «Alex»>/<todo\_id: 1> {«text»: «Buy some vegetables», «TODO»: str}
* Ожидаемый результат: сервер присылает ответ со статусом 201(Created)
* Проверить, что памятка изменилась: GET /todos/<username: «Alex»>/<todo\_id: 1>
* Ожидаемый результат: сервер присылает ответ со статусом 200 и памятку в теле ответа. Значение ключа «text» в заметке должно измениться.

7. Удалить памятку:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> «Buy some fruits».
* Отправить запрос на сервер: DELETE /todos/<Alex: str>/<todo\_id: 1>.
* Ожидаемый результат: сервер присылает ответ со статусом 204 (NoContent).
* Проверить, что памятка удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
* Ожидаемый результат: сервер присылает ответ со статусом 404 (NotFound)

8. Получить памятки пользователя:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должны быть созданы памятки cо значениями <todo\_id: 1> и <todo\_id: 2> и текстом «Buy some fruits», «Buy some vegetables».
* Отправить запрос на сервер: GET /todos/<username: «Alex»>
* Ожидаемый результат: сервер присылает ответ со статусом 200(OK) и двумя памятками пользователя в теле сообщения.

# Негативные тесты

## Управление аккаунтом

### Тестовый набор «Создание пользователя»

9. Создание пользователя с недопустимой длиной имени пользователя(>50 симовлов):

* Отправить запрос на сервер: POST /users {«username»: «too\_long\_username > 50 characters», «password1»: «Race2021», «password2»: «Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

10. Попытка создания пользователя без имени:

* Отправить запрос на сервер: POST /users {«username»: «», «password1»: «Race2021», «password2»: «Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

11. Создание пользователя с недопустимой длиной пароля(>20 символов):

* Отправить запрос на сервер: POST /users {«username»: «Alex», «password1»: «too\_long\_password > 20 characters», «password2»: «too\_long\_password > 20 characters»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

12. Создание пользователя без пароля :

* Отправить запрос на сервер: POST /users {«username»: «Alex», «password1»: «», «password2»: «»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

13. Создание пользователя с пустым полем «password1»:

* Отправить запрос на сервер: POST /users {«username»: «Alex», «password1»: «», «password2»: «Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

14. Создание пользователя с пустым полем «password2»:

* Отправить запрос на сервер: POST /users {«username»: «Alex», «password1»: «Race2021», «password2»: «»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

### Тестовый набор «Логин»

15. Логин (Аутентификация). Попытка аутентификации с пустым именем пользователя:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «», «password1»: Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

16. Логин (Аутентификация). Попытка аутентификации с пустым паролем:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password1»: «»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

17. Логин (Аутентификация). Попытка аутентификации с невалидным именем пользователя:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «alex», «password1»: Race2021»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

18. Логин (Аутентификация). Попытка аутентификации с невалидным паролем:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password1»: Race1111»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

19. Логин (Аутентификация). Попытка аутентификации с пустыми именем пользователя и паролем:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021».
* Отправить запрос на сервер: POST /login {«username»: «», «password1»: «»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

### Тестовый набор «Обновление пароля пользователя»

20. Обновление пароля пользователя. Попытка обновления пароля без ввода старого пароля:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «», «password1»: «Amusement20», «password2»: «Amusement20»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
  + Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

21. Обновление пароля пользователя. Попытка обновления пароля без ввода нового пароля:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «Race2021», «password1»: «», «password2»: «»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
  + Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

22. Обновление пароля пользователя. Попытка обновления пароля с невалидным старым паролем:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
* Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «Race», «password1»: «Amusement20», «password2»: «Amusement20»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

23. Обновление пароля пользователя. Попытка обновления пароля с пустым полем «password1»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «Race2021», «password1»: «», «password2»: «Amusement20»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
  + Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

24. Обновление пароля пользователя. Попытка обновления пароля с пустым полем «password2»:

* Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
* Отправить запрос на сервер: PUT /users {«username»: «Alex», «old\_password»: «Race2021», «password1»: «Amusement20», «password2»: «»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
* Отправить запрос на сервер: POST /login {«username»: «Alex», «password»: «Amusement20»}.
* Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).

## Управление контентом

### Тестовый набор «Создать памятку»

25. Создать памятку. Попытка создать памятку без имени пользователя:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: POST /todos/< «username»: «»>/<todo\_id: 1> {«text»: «Buy some fruits» , «status»: «TODO»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

26. Создать памятку. Попытка создать памятку без id :

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: POST /todos/< «username»: «Alex»>/<todo\_id: > {«text»: «Buy some fruits» , «status»: «TODO»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

27. Создать памятку. Попытка создать «дубликат» памятки (вторую памятку с идентичным id):

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации. Должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: POST /todos/< «username»: «Alex»>/<todo\_id: 1> {«text»: «Buy some fruits» , «status»: «TODO»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 412 (Precondition Failed).

28. Создать памятку. Попытка создать памятку с пустым полем «status»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: POST /todos/< «username»: «Alex»>/<todo\_id: 1> {«text»: «Buy some fruits» , «status»: «»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

29. Создать памятку. Попытка создать памятку с пустыми полями «username», «id», «status»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации.
  + Отправить запрос на сервер: POST /todos/< «username»: «»>/<todo\_id: > {«text»: «Buy some fruits» , «status»: «»}.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad request).

### Тестовый набор «Получить памятку»

30. Получить памятку. Попытка получить памятку без имени пользователя:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: GET /todos/<username: «»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

31. Получить памятку. Попытка получить памятку без id:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: GET /todos/<username: «Alex»>/<todo\_id: >
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

31.1 Получить памятку. Попытка получить несуществующую памятку:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: GET /todos/<username: «Alex»>/<todo\_id: 2>
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).

32. Получить памятку. Попытка получения памятки пользователя до прохождения процедуры аутентификации:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 401 (Unauthorized).

### Тестовый набор «Редактировать памятку»

33. Редактировать памятку. Отправка запроса с пустым значением поля «username»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос для изменения памятки: PUT /todos/<username: «»>/<todo\_id: 1> {«text»: «Buy some vegetables», «TODO»: str}
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).  
    Проверить, что памятка у зарегистрированного пользователя не изменилась: GET /todos/<username: «Alex»>/<todo\_id: 1>.
  + Ожидаемый результат: сервер присылает ответ со статусом 200 и памятку в теле ответа. Значения всех ключей в должны остаться прежними.

34. Редактировать памятку. Отправка запроса с пустым значением поля «id»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос для изменения памятки: PUT /todos/<username: «Alex»>/<todo\_id: > {«text»: «Buy some vegetables», «TODO»: str}
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
  + Проверить, что памятка у зарегистрированного пользователя не изменилась: GET /todos/<username: «Alex»>/<todo\_id: 1>.
  + Ожидаемый результат: сервер присылает ответ со статусом 200 и памятку в теле ответа. Значения всех ключей должны остаться прежними.

35. Редактировать памятку. Отправка запроса с пустым полем «status»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: PUT /todos/<username: «Alex»>/<todo\_id: 1> {«text»: «Buy some vegetables», «»: str}
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad request)
  + Проверить, что памятка не изменилась: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 и памятку в теле ответа. Значения всех ключей должны остаться прежними.

36. Редактировать памятку. Отправка запроса на изменение памятки с пустыми полями «username», «id», «status»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: PUT /todos/<username: «»>/<todo\_id: 1> {«text»: «», «»: str}
  + Ожидаемый результат: сервер присылает ответ со статусом 400(Bad Request).
  + Проверить, что памятка не изменилась: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 и памятку в теле ответа. Значения всех ключей должны остаться прежними.

### Тестовый набор «Удалить памятку»

37. Удалить памятку. Попытка удалить памятку без указания «id»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: DELETE /todos/<Alex: str>/<todo\_id: >
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request)
  + Проверить, что памятка не удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и памяткой в теле ответа.

38. Удалить памятку. Попытка удалить памятку без указания имени пользователя:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: DELETE /todos/<: str>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).
  + Проверить, что памятка не удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и памяткой в теле ответа.

39. Удалить памятку. Попытка удалить памятку без указания имени пользователя и «id»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: DELETE /todos/<: str>/<todo\_id: >.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).
  + Проверить, что памятка не удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и памяткой в теле ответа.

40. Удалить памятку. Попытка удалить памятку без указания имени пользователя:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: DELETE /todos/<: str>/<todo\_id: 1>.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).
  + Проверить, что памятка не удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и памяткой в теле ответа.

41. Удалить памятку. Попытка удалить памятку без «id»:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должна быть создана памятка cо значением <todo\_id: 1> и текстом «Buy some fruits».
  + Отправить запрос на сервер: DELETE /todos/<Alex: str>/<todo\_id: >.
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Bad Request).
  + Проверить, что памятка не удалена: GET /todos/<username: «Alex»>/<todo\_id: 1>
  + Ожидаемый результат: сервер присылает ответ со статусом 200 (OK) и памяткой в теле ответа.

### Тестовый набор «Получить все памятки пользователя»

42. Получить памятки пользователя. Попытка получить памятки пользователя без прохождения процедуры аутентификации:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, должны быть созданы памятки cо значениями <todo\_id: 1> и <todo\_id: 2>, «Buy some fruits» и «Buy some vegetables» соответственно.
  + Отправить запрос на сервер: GET /todos/<username: «Alex»>
  + Ожидаемый результат: сервер присылает ответ со статусом 400 (Unauthorized).

43. Получить памятки пользователя. Попытка получить памятки пользователя из пустой базы:

* + Предварительные условия: должен быть создан пользователь «Alex» с паролем «Race2021», должна быть пройдена процедура аутентификации, памятки в базе данных должны отсутствовать.
  + Отправить запрос на сервер: GET /todos/<username: «Alex»>.
  + Ожидаемый результат: сервер присылает ответ со статусом 204(No Content).

## Повторное выполнение негативных тест-кейсов из разделов «Управление аккаунтом», «Управление контентом».

Повторить негативные тест-кейсы из тестовых наборов «Обновление пароля пользователя», «Создать памятку», «Получить памятку», «Редактировать памятку», «Удалить памятку» без прохождения процедуры аутентификации.  
Ожидаемый результат всех тестов: сервер возвращает ответ со статусом 401(Unauthorized).

Повторить негативные тест-кейсы из тестовых наборов «Обновление пароля пользователя», «Создать памятку», «Получить памятку», «Редактировать памятку», «Удалить памятку» следующим образом:

* Создать пользователя «Alex»;
* Создать пользователя «Olga»;
* Пройти аутентификацию под пользователем «Alex»;
* Повторить вышеуказанные тест-кейсы из под пользователя «Alex» для аккаунта пользователя «Olga»;
* Ожидаемый результат всех тестов: сервер возвращает ответ со статусом 401(Unauthorized).