

[PATCH /users/{user_id}/block] - метод для блокировки пользователя администратором

Автор: Даниэл Пянзин
Telegram: @DanielPyanzin

История изменений

[Ссылка на Jira](#)

Связанные документы (внутренние)

[A Free Database Designer for Developers and Analysts](#) - БД проекта

[U-Verse](#) - Figma проекта

[\[GET /users/{user_id}\] - Просмотр профиля пользователя администратором](#) - получение профиля ученика

Оглавление:

[Общее описание](#)

[Доступы и ограничения](#)

[Запрос](#)

[Пример запроса:](#)

[Тело запроса \(Body\)](#)

[Пример ответа - Успех](#)

[Пример ответа - Ошибка авторизации](#)

[Пример ответа - Ошибка прав доступа](#)

[Пример ответа - Некорректный формат](#)

[Пример ответа - Пользователь не найден](#)

[Пример ответа - Пользователь уже заблокирован](#)

[Пример ответа - Ошибка при работе с базой данных](#)

[Требования к реализации](#)

[Маппинг данных](#)

Общее описание

Метод PATCH /users/{user_id}/block предназначен для **блокировки профиля студента** со стороны администратора.

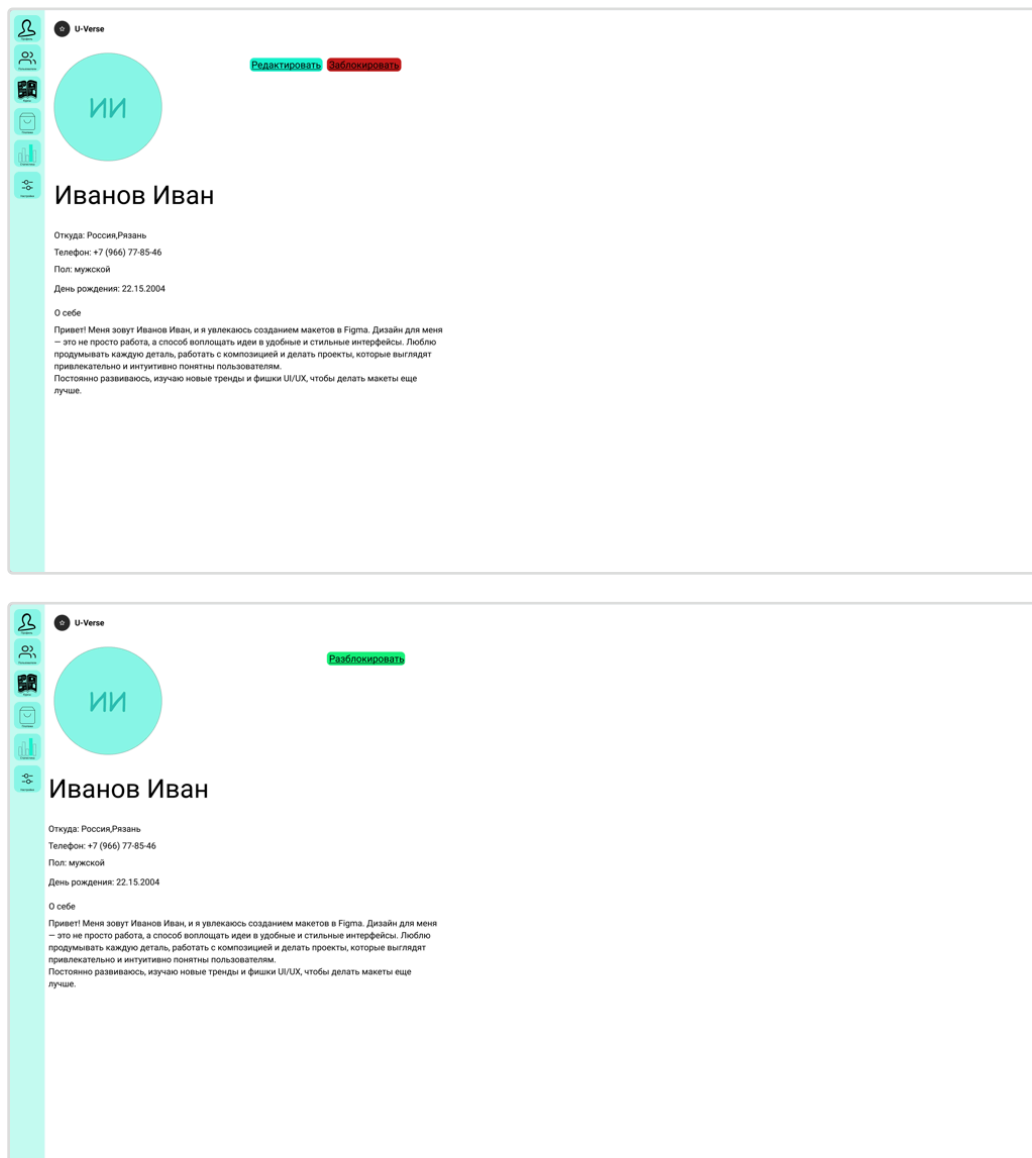
Блокировка может использоваться в случае нарушений, подозрений на мошеннические действия, просьбы пользователя или иных административных причин.

После успешной блокировки студент **теряет доступ к системе**:

- не может авторизоваться;
- не может выполнять действия от своего имени;
- отображается как «Заблокирован» в административных интерфейсах.

Метод вызывается с идентификатором пользователя (users `_id`) и требует подтверждения через Bearer-токен администратора.

Блокировка является **обратимой** — для снятия блокировки будет использоваться отдельный метод `[PATCH /users/{user_id}/un-block]` - метод для разблокировки пользователя администратором.



Доступы и ограничения

- Требуется авторизация по Bearer-токену.

- Роль: Администратор
- Администратор **не может заблокировать другого администратора**
- Ограничения по количеству запросов: не более 20 в минуту

Запрос

Method type: PATCH

BaseURL: <https://api.uverse.com>

API name: admin

API version: v1

Method name: users/{user_id}/block

Authorization:

- Тип: Token (Bearer Token)
- Роли: Администратор

Headers, Cookies:

- Authorization: Bearer <token>
- Content-Type: multipart/form-data

Пример запроса:

```
1 PATCH /admin/v1/users/1d9008b7-9c1f-4d18-9635-c08653597f5a/block
```

query-parameters:

- Нет

Тело запроса (Body)

Пример:

```
1 {  
2   "block_type": "temporary", // или "permanent"  
3   "block_until": "2025-07-01T00:00:00Z", // опционально для временной блокировки  
4   "reason": "Нарушение правил платформы"  
5 }
```

Пример ответа - Успех

Код HTTP: 204

Тело ответа (Body):

Отсутствует

Пример ответа - Ошибка авторизации

Код HTTP: 401

Тело ответа (Body):

```
1 {  
2   "code": "1001",  
3   "message": "Пользователь не авторизован"  
4 }
```

Пример ответа - Ошибка прав доступа

Код HTTP: 403

Тело ответа (Body):

```
1 {  
2   "code": "1002",  
3   "message": "Недостаточно прав для выполнения операции"  
4 }
```

Пример ответа - Некорректный формат

Код HTTP: 400

Тело ответа (Body):

```
1 {  
2   "code": "2003",  
3   "message": "Некорректный формат даты: 2025-31-07T00:00:00Z"  
4 }
```

Пример ответа - Пользователь не найден

Код HTTP: 404

Тело ответа (Body):

```
1 {  
2   "code": "3001",  
3   "message": "Пользователь не найден"  
4 }
```

Пример ответа - Пользователь уже заблокирован

Код HTTP: 409

Тело ответа (Body):

```
1 {  
2   "code": "3010",  
3   "message": "Невозможно применить действие: пользователь уже заблокирован"  
4 }
```

Пример ответа - Ошибка при работе с базой данных

Код HTTP: 500

Тело ответа (Body):

```
1 {  
2   "code": "5002",  
3   "message": "Ошибка при работе с базой данных"  
4 }
```

Требования к реализации

Основной сценарий:

1. Получить на вход {user_id}.
2. Проверить авторизацию пользователя - наличие заголовка Authorization в запросе:
 - a. **Если авторизация не пройдена, то вернуть ошибку HTTP-401 - "Пользователь не авторизован".**
 - b. Иначе - далее.
3. Проверить в БД, есть ли соответствующий пользователь с указанным {user_id}:
 - a. **Если такого пользователя нет, вернуть ошибку с кодом HTTP-404 - "Пользователь не найден"**
 - b. Иначе - далее.
4. Изменить параметры блокируемого пользователя и **вернуть ответ HTTP 204.**

Обработка исключительных ситуаций:

1. Нет прав для внесения изменений.
Если пользователь пытается заблокировать администратора, то вернуть ошибку с кодом HTTP-403/1002 - "Недостаточно прав для выполнения операции"
2. Пользователь уже заблокирован.
 - a. Если статус блокировки "permanent" и применяется блокировка "permanent", то вернуть ошибку с кодом HTTP-409/3010 - "Невозможно применить действие:

пользователь уже заблокирован".

- б. Если статус блокировки "temporary" и применяется блокировка permanent, то изменить статус блокировки на "permanent" и вернуть HTTP 204.
- с. Если статус блокировки "permanent" и применяется блокировка "temporary" , то вернуть ошибку с кодом HTTP-409/3010- "Невозможно применить действие: пользователь уже заблокирован".

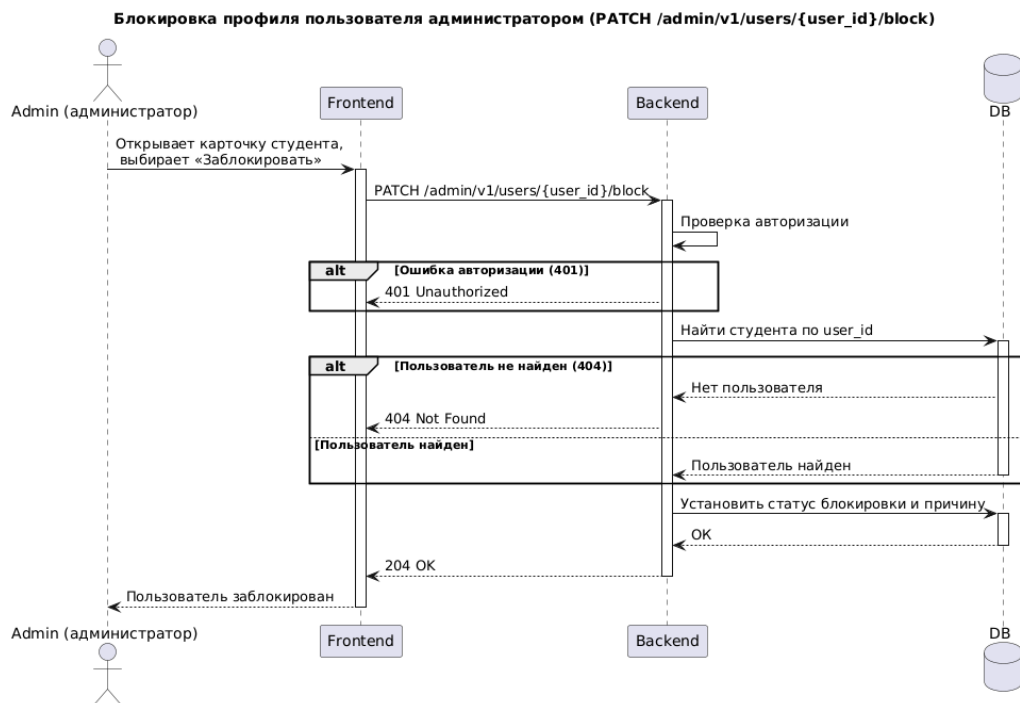
3. Ошибка при обращении к БД.

В процессе работы алгоритма не удастся подключиться к БД. Вернуть пользователю код ошибки HTTP-500/5002 с соответствующим текстом "Ошибка при работе с базой данных".

4. Неверный формат даты блокировки.

Если некорректный формат данных - то вернуть ошибку HTTP-400 - "Неверный формат даты блокировки".

UML Схема:



✓ Код PlantUML

@startuml

title Блокировка профиля пользователя администратором (PATCH /admin/v1/users/{user_id}/block)

actor "Admin (администратор)" as Admin

participant "Frontend" as Frontend

participant "Backend" as API

database "DB" as DB

Admin → Frontend : Открывает карточку студента,\n выбирает «Заблокировать»
activate Frontend

Frontend → API : PATCH /admin/v1/users/{user_id}/block
activate API

API → API : Проверка авторизации
alt Ошибка авторизации (401)

API → Frontend : 401 Unauthorized
end

API → DB : Найти студента по user_id
activate DB

alt Пользователь не найден (404)
DB → API : Нет пользователя

API → Frontend : 404 Not Found

else Пользователь найден
DB → API : Пользователь найден
deactivate DB
end

API → DB : Установить статус блокировки и причину
activate DB

DB → API : OK
deactivate DB

API → Frontend : 204 OK
deactivate API

Frontend → Admin : Пользователь заблокирован

deactivate Frontend

@enduml

Маппинг данных

Для тела запроса:

Параметр в JSON	Описание / комментарий, требования к преобразованию после извлечения из БД или обработки	Обяз.	Тип данны х JSON	В БД
block_type	Тип блокировки	+	string	user.block_ty pe
block_until	Дата окончания блокировки		string	user.block_u ntil
reason	Причина	+	string	user.block_re ason

Автор: Даниэл Пянзин
Telegram: @DanielPyanzin