

# [PATCH /users/{user\_id}/un-block] - метод для разблокировки 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 прав доступа](#)
  - [Пример ответа - Некорректный формат](#)
  - [Пример ответа - Ошибка авторизации](#)
  - [Пример ответа - Пользователь не найден](#)
  - [Пример ответа - Сервис недоступен, ошибка на сервере](#)
  - [Пример ответа - Недостаточно места в хранилище](#)
- [Требования к реализации](#)
- [Маппинг данных](#)

## Общее описание

Метод PATCH /users/{user\_id}/un-block предназначен для **снятия блокировки с аккаунта пользователя** администратором и кроном.

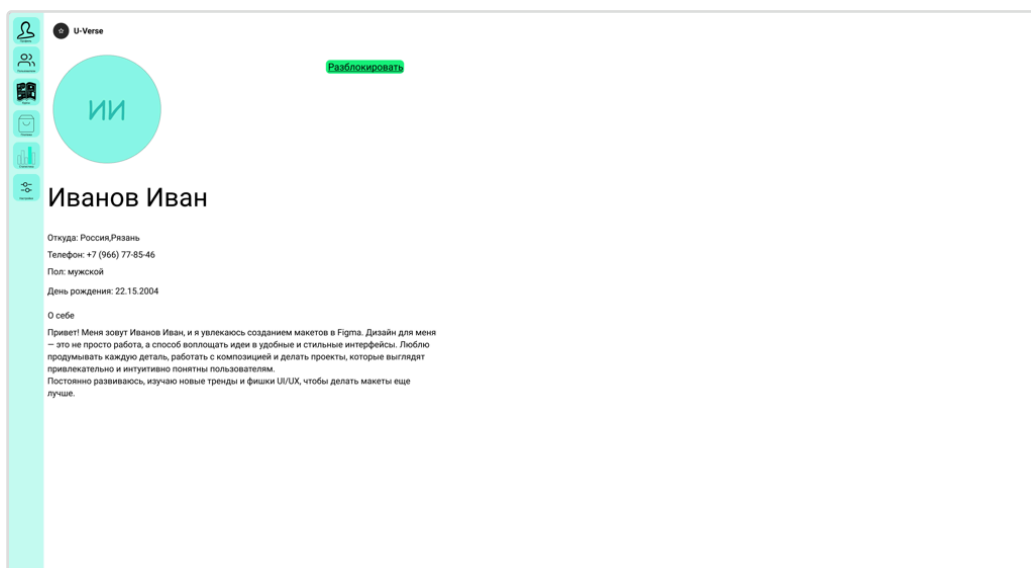
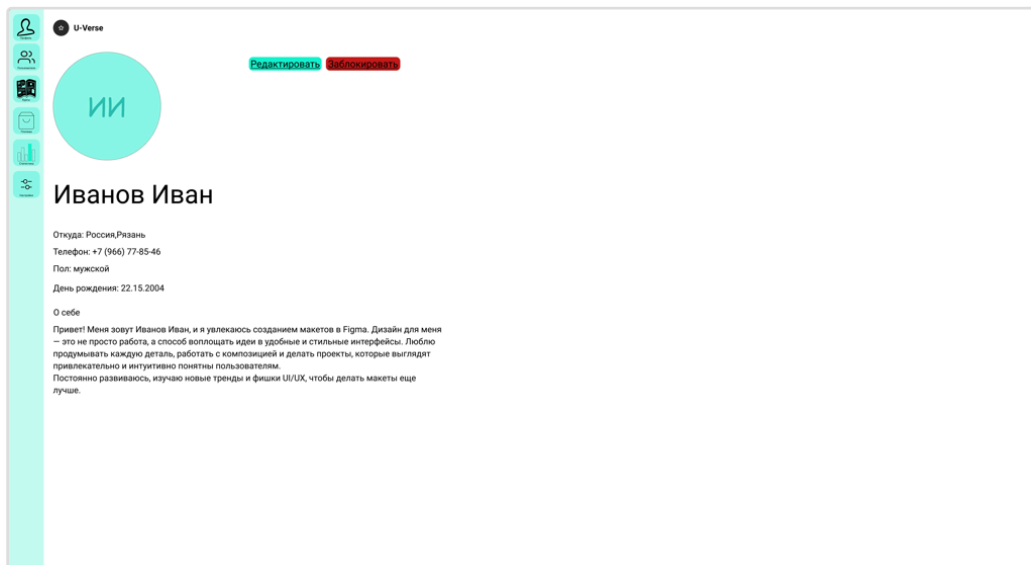
Разблокировка может применяться как после временной блокировки по истечении срока, так и в случае ручного восстановления доступа (по запросу, ошибочной блокировке и т.д.).

После успешной разблокировки пользователь:

- Снова получает доступ к авторизации в системе.
- Может выполнять действия от своего имени.
- Отображается как «Активен» в административных интерфейсах.

Метод требует авторизации и соответствующих прав доступа администратора.

Работает только в случае, если пользователь **заблокирован**.



## Доступы и ограничения

- Требуется авторизация по Bearer-токену.
- Роль: Администратор
- Администратор **не может разблокировать 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}/un-block

### Authorization:

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

### Headers, Cookies:

- Authorization: Bearer <token>

### Пример запроса:

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

### query-parameters:

- Нет

### Тело запроса (Body)

Пример:

```
1 {  
2   "reason": "Причина"  
3 }
```

### 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 - Пользователь не найден

**Код HTTP:** 404

**Тело ответа (Body):**

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

Пример ответа - Пользователь не заблокирован

**Код HTTP:** 409

**Тело ответа (Body):**

```
1 {  
2   "code": "3014",  
3   "message": "Невозможно применить действие: пользователь не заблокирован"  
4 }
```

Пример ответа - Превышен лимит запросов

Код HTTP: 429

Тело ответа (Body):

```
1 {  
2   "code": "1005",  
3   "message": "Превышено количество запросов. Попробуйте позже"  
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:
  - а. **Если авторизация не пройдена, то вернуть ошибку HTTP-401 - "Пользователь не авторизован".**
  - б. Иначе - далее.
3. Проверить в БД, есть ли соответствующий пользователь с указанным {user\_id}:
  - а. **Если такого пользователя нет, вернуть ошибку с кодом HTTP-404 - "Пользователь не найден"**
  - б. Иначе - далее.
4. Изменить параметры разблокированного пользователя и **вернуть ответ HTTP 204.**

### Обработка исключительных ситуаций:

1. Нет прав для внесения изменений.  
Если пользователь пытается разблокировать администратора, то вернуть ошибку с кодом HTTP-403/1002 - "Недостаточно прав для разблокировки"

2. Пользователь не находится в статусе блокировки.

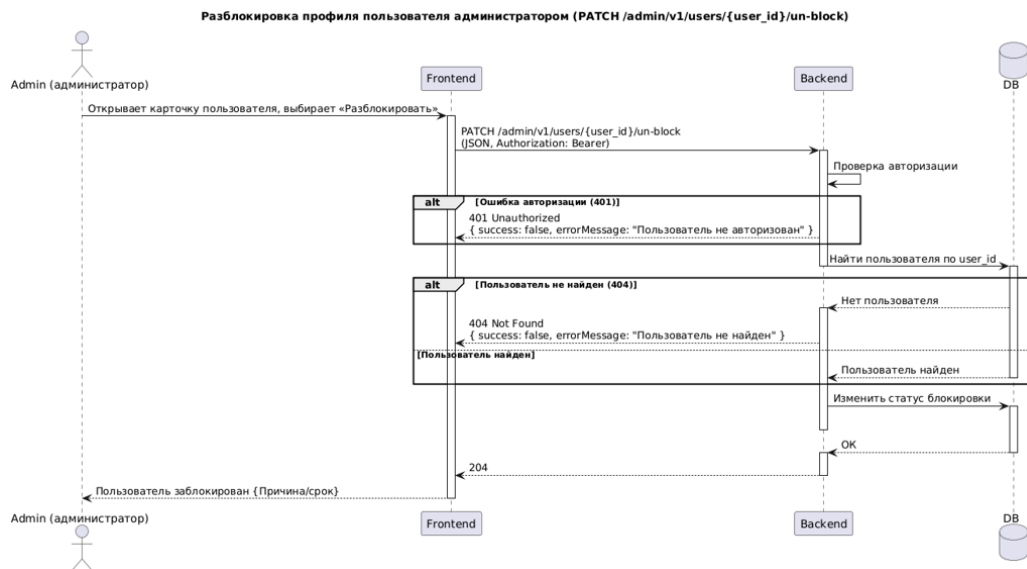
Если пользователь не заблокирован, то вернуть ошибку с кодом HTTP-409/3014 — "Невозможно применить действие: пользователь не заблокирован".

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".

## UML Схема:

### Пример ответа - Пользователь не заблокирован



Код HTTP: 429

### Тело ответа (Body):

```
1 {
2   "code": "1005",
3   "message": "Превышено количество запросов. Попробуйте позже"
4 }
```

✓ Код PlantUML

@startuml

title Разблокировка профиля пользователя администратором (PATCH /admin/v1/users/{user\_id}/un-block)

actor "Admin (администратор)" as Admin

participant "Frontend" as Frontend

participant "Backend" as API

database "DB" as DB

Admin → Frontend : Открывает карточку пользователя, выбирает «Разблокировать»  
activate Frontend

Frontend → API : PATCH /admin/v1/users/{user\_id}/un-block\n(JSON, Authorization: Bearer)  
activate API

API → API : Проверка авторизации

alt Ошибка авторизации (401)

API → Frontend : 401 Unauthorized\n{ success: false, errorMessage: "Пользователь не авторизован" }

end

API → DB : Найти пользователя по user\_id

deactivate API

activate DB

alt Пользователь не найден (404)

DB → API : Нет пользователя

activate API

API → Frontend : 404 Not Found\n{ success: false, errorMessage: "Пользователь не найден" }

else Пользователь найден

DB → API : Пользователь найден

deactivate DB

end

API → DB : Изменить статус блокировки

activate DB

deactivate API

DB → API : OK

deactivate DB

activate API

API → Frontend : 204

deactivate API

Frontend → Admin : Пользователь заблокирован {Причина/срок}

deactivate API

deactivate Frontend

@enduml

## Маппинг данных

Для тела запроса:

Параметр в JSON	Описание / комментарий, требования к преобразованию после извлечения из БД или обработки	Обяз.	Тип данных в JSON	В БД
reason	Причина разблокировки		string	user.unblock _reason

---

Автор: Даниэл Пянзин

Telegram: @DanielPyanzin