

# [PATCH /users/profile] - Изменение профиля ученика

Автор: Даниэл Пянзин  
Telegram: @DanielPyanzin

---

## История изменений

20.08.2025 — постановка задачи,  
первая версия

## Связанные документы (внутренние)

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

- БД проекта

 [U-Verse](#) - Figma проекта

 [\[GET /users/profile\]](#) - получить информацию о профиле студента.

-  
Получение профиля

## 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 позволяет авторизованному студенту в одном запросе частично изменять персональные данные профиля и при необходимости **загружать/обновлять** либо **удалять** аватар. Применяется в интерфейсе личного кабинета, когда пользователь вносит изменения в свой профиль. Это позволяет пользователю персонализировать свой профиль, установив актуальную фотографию и информацию о себе.

Изображение передается как base64-строка в JSON. Допустимые форматы: .jpg, .jpeg, .png. Максимальный размер файла — 2 МБ.

Доступ к вызову имеют только студенты, и только в отношении собственного профиля. Внесение изменений в чужой профиль запрещено и приводит к ошибке доступа. После успешной загрузки сервер возвращает прямую ссылку на новое изображение.

## Доступы и ограничения

- Требуется авторизация по Bearer-токену.
- Роль: Студент
- Ограничения по количеству запросов: не более 10 в минуту
- Пользователь может изменить только собственный профиль
- Если студент заблокирован — изменения запрещены

## Запрос

**Method type:** PATCH

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

**API name:** public

**API version:** v1

**Method name:** /profile

**Authorization:**

- Тип: Bearer Token
- Роли: Студент

**Headers, Cookies:**

- Authorization: Bearer <token>

**Пример запроса:**

```
1 PATCH https://api.uverse.com/public/v1/users/profile
```

## query-parameters:

- Нет

## Тело запроса (Body)

Пример:

```
1 {
2   "last_name": "Иванов",
3   "first_name": "Иван",
4   "birthday": "2001-01-01",
5   "gender": 1,
6   "city": "Рязань",
7   "phone": "79271830303",
8   "about": "Люблю аналитические курсы",
9   "avatar": {
10    "mime": "image/jpeg",
11    "data": "iVBORw0KGgoAAAANSUhEUgAA..."
12  }
13 }
```

## Варианты поля "avatar":

### 1. Загрузка/Обновление

```
1 "avatar": {
2   "mime": "image/png",
3   "data": "<base64>"
4 }
```

### 2. Удаление аватара:

```
1 "avatar": {
2   "delete": true
3 }
```

Пример ответа - Успех

Код HTTP: 200

## Тело ответа (Body):

```
1 {
2   "last_name": "Иванов",
3   "first_name": "Иван",
4   "birthday": "2001-01-01",
5   "gender": 1,
6   "city": "Рязань",
7   "phone": "79271830303",
```

```
8  "about": "Люблю аналитические курсы",
9  "avatar_url": "https://uverse.com/public/uploads/avatars/550e8400-...png"
10 }
11
```

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 }
```

Другие варианты: 1003

Пример ответа - Некорректный формат

**Код HTTP:** 400

**Тело ответа (Body):**

```
1 {
2   "code": "2001",
3   "message": "Некорректный формат данных: поле email"
4 }
```

Другие варианты:

1. 2002

2. 2003

3. 2004

4. 2005

5. 2006

6. 2007

7. 2008

8. 2009

См. [📘 Единый справочник ошибок API U-Verse](#)

Пример ответа - Пользователь не найден

**Код HTTP:** 404

**Тело ответа (Body):**

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

Пример ответа - Превышен лимит запросов

**Код HTTP:** 429

```
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 }
```

Пример ответа - Ошибка при обращении к файловому хранилищу

**Код HTTP:** 502

**Тело ответа (Body):**

```
1 {  
2   "code": "4001",  
3   "message": "Ошибка при обращении к файловому хранилищу"  
4 }
```

Пример ответа - Недостаточно места в хранилище

Код HTTP: 507

Тело ответа (Body):

```
1 {  
2   "code": "4006",  
3   "message": "Недостаточно места для сохранения изображения. Попробуйте позже."  
4 }
```

## Требования к реализации

### Основной сценарий:

1. Получить запрос и проверить авторизацию пользователя - наличие заголовка Authorization в запросе:
  - a. **Если авторизация не пройдена, то вернуть ошибку HTTP-401/1001 - "Пользователь не авторизован".**
  - b. Иначе - далее.
2. Проверить, что введенные данные корректны:
  - a. **Если некорректный формат данных - то вернуть ошибку HTTP-400/(Опционально) - Некорректный формат данных: поле {название поля}.**
  - b. Иначе - далее.
3. Определить `user.id` по значению токена и проверить наличие соответствующего пользователя в БД:
  - a. **Если такого пользователя нет, вернуть ошибку с кодом HTTP-404/3001 - "Пользователь не найден"**
  - b. Иначе - далее.
4. Если есть аватар:
  - a. Заменить старый аватар на новый в ФХ. Старый аватар и ссылка на него удаляются.
  - b. Иначе - далее.
5. Сохранить данные профиля и аватар в БД.
6. Вернуть успешный ответ.

### Обработка исключительных ситуаций:

1. Нет прав для внесения изменений.  
Если пользователь пытается изменить чужой профиль, то вернуть ошибку с кодом HTTP-403/1002 - "Недостаточно прав для выполнения операции"

2. Аккаунт студента заблокирован.

Если пользователь заблокирован (user.is\_active=false) прекратить обработку без изменений профиля и вернуть ошибку HTTP-403/1003.

3. Превышен лимит запросов.

Если превышен лимит запросов по пользователю/IP, то вернуть ошибку HTTP-429/1005 - "Превышено количество запросов. Попробуйте позже"

4. Ошибка при обращении к БД.

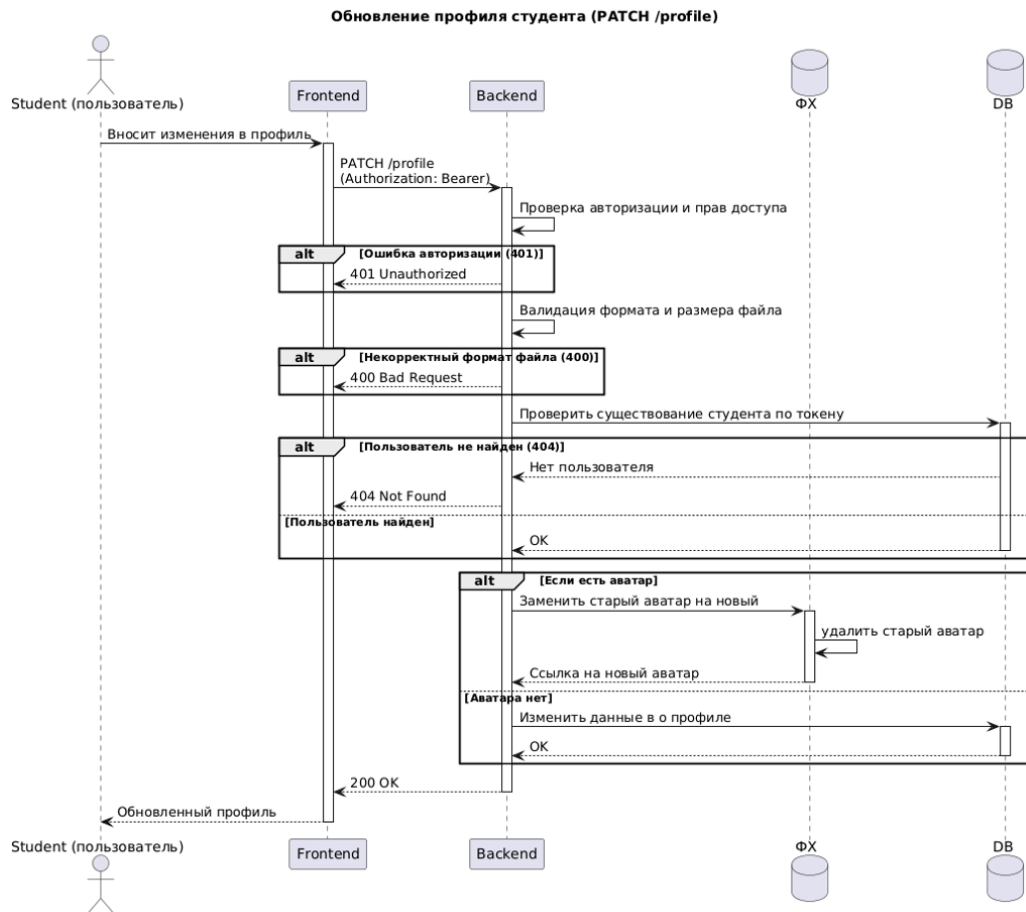
В процессе работы алгоритма не удастся подключиться к БД. Вернуть пользователю код ошибки HTTP-500/5002 с соответствующим текстом "Ошибка при работе с базой данных".

5. Ошибка при обращении к ФХ.

В процессе работы алгоритма не удастся подключиться к ФХ. Вернуть пользователю код ошибки HTTP-502/4001 с соответствующим текстом "Ошибка при обращении к файловому хранилищу".

6. Если хранилище недоступно или превышен лимит дискового пространства. Вернуть пользователю код ошибки HTTP-507/4006 с соответствующим текстом "Недостаточно места для сохранения изображения. Попробуйте позже."

**UML Схема:**



✓ Код PlantUML

@startuml

title Обновление профиля студента (PATCH /profile)

actor "Student (пользователь)" as Student

participant "Frontend" as Frontend

participant "Backend" as API

database "ФХ" as FileService

database "DB" as DB

Student → Frontend : Вносит изменения в профиль

activate Frontend

Frontend → API : PATCH /profile\n(Authorization: Bearer)

activate API

API → API : Проверка авторизации и прав доступа

alt Ошибка авторизации (401)

API → Frontend : 401 Unauthorized

end

API → API : Валидация формата и размера файла



```
alt Некорректный формат файла (400)
  API → Frontend : 400 Bad Request
end

API → DB : Проверить существование студента по токену
activate DB
alt Пользователь не найден (404)
  DB → API : Нет пользователя
  API → Frontend : 404 Not Found
else Пользователь найден
  DB → API : OK
  deactivate DB
end

alt Если есть аватар
API → FileService : Заменить старый аватар на новый
activate FileService
FileService → FileService : удалить старый аватар
FileService → API : Ссылка на новый аватар
deactivate FileService

else Аватара нет
API → DB : Изменить данные в профиле
activate DB
DB → API : OK
deactivate DB
end

API → Frontend : 200 OK
deactivate API
Frontend → Student : Обновленный профиль
deactivate Frontend
@enduml
```

## Маппинг данных

Для тела запроса:

Параметр в JSON	Описание / комментарий,	Обяз.	Тип данны	В БД
--------------------	----------------------------	-------	--------------	------

	требования к преобразованию после извлечения из БД или обработки		x JSON	
last_name	Фамилия пользователя		string	user.last_name
first_name	Имя пользователя	+	string	user.first_name
birthday	Дата рождения в формате YYYY-MM-DD		string (date)	user.birthday
gender	Пол пользователя		integer	user.gender_id
city	Город проживания		string	user.city
phone	Номер телефона в международном формате, без пробелов и спецсимволов		string	display_number
about	Краткая информация о пользователе		string	user.about
avatar.mime	MIME-тип файла		string	file.file_format
avatar.data	base64-строка файла		string	<b>Не хранится в БД;</b> по результату создаётся запись в <b>file</b> и проставляется <b>user.avatar</b>

				<code>tar_file_id</code> .
--	--	--	--	----------------------------

Для тела ответа:

Параметр в JSON	Описание / комментарий, требования к преобразованию после извлечения из БД или обработки	Обяз.	Тип данных JSON	В БД
last_name	Фамилия пользователя		string	user.last_name
first_name	Имя пользователя	+	string	user.first_name
birthday	Дата рождения в формате <code>YYYY-MM-DD</code>		string (date)	user.birthday
gender	Пол пользователя		integer	user.gender_id
city	Город проживания		string	user.city
phone	Номер телефона в международном формате, без пробелов и спецсимволов		string	display_number
about	Краткая информация о пользователе		string	user.about
avatar_url	Ссылка на актуальный аватар. Если аватар удалён — ссылка на дефолтный.		string (url)	<b>Вычисляется:</b> <code>user.avatar_file</code>

			<code>_id →</code> <code>file.sto</code> <code>ring_ur</code> <code>l</code> . Если <code>NULL</code> — подставляет ся дефолтный URL из конфигураци и.
--	--	--	--

---

Автор: Даниэл Пянзин  
Telegram: @DanielPyanzin