

[GET /users/profile] - получить информацию о профиле студента.


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

 [\[GET /directories/countries\]](#) - получение справочника по странам - Справочник стран

 [\[GET /directories/genders\]](#) - Получение справочника полов (Гендеров) - справочник полов(гендеров)

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 - Пользователь не найден](#)

[Пример ответа - Превышен лимит запросов](#)

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

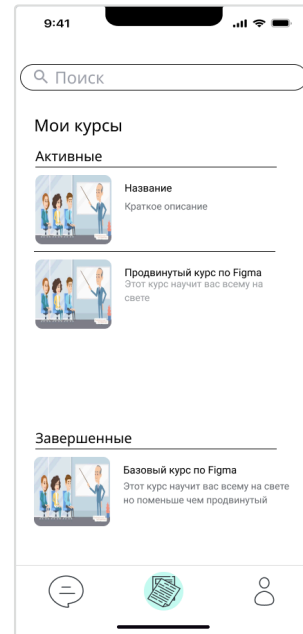
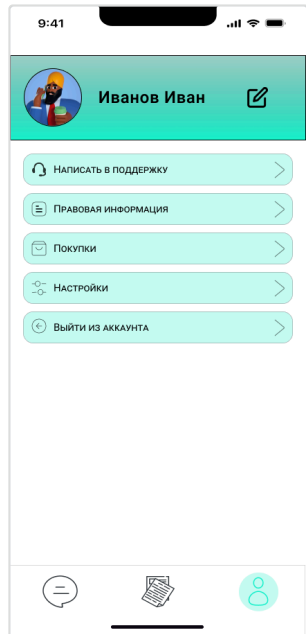
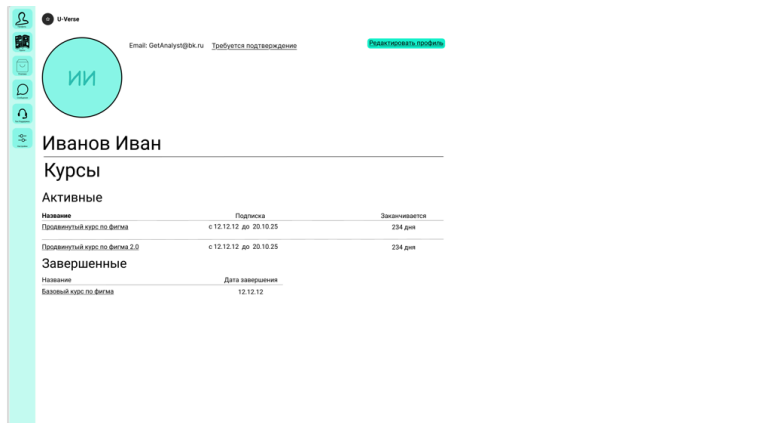
[Требования к реализации](#)

[Маппинг данных](#)

Общее описание

Метод GET /users/profile предназначен для получения полной информации о студенте: личных данных, страны проживания и подключенных курсов. Используется на клиентской стороне для отображения профиля пользователя.

Метод применяется в ЛК при переходе на страницу профиля в web-приложении или при запуске приложения ios/android.



Доступы и ограничения

- Требуется авторизация по Bearer-токену.
- Роль: Студент
- Ограничения по количеству запросов: не более 20 в минуту
- Пользователь может изменить только собственный профиль

Запрос

Method type: GET

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

API name: public

API version: v1

Method name: /users/profile

Authorization:

- Требуется авторизация по Bearer-токену.
- Роль: Студент

- Пользователь может просмотреть **только собственный профиль**.

Headers, Cookies:

- Authorization: Bearer <token>

Пример запроса:

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

query-parametrs:

- Нет

Тело запроса (Body)

Не поддерживается.

Пример ответа - Успех

Код HTTP: 200

Тело ответа (Body):

```
1 {
2   "id": "1d9008b7-9c1f-4d18-9635-c08653597f5a",
3   "first_name": "Иванов",
4   "last_name": "Иван",
5   "birthday": "2001-01-01",
6   "gender": 1,
7   "city": "Рязань",
8   "phone": "79271830303",
9   "email": "GetAnalyst@bk.ru",
10  "about": "Я люблю гулять",
11  "avatar_url": "https://uverse.com/public/defaults/avatar.png",
12  "country": "1",
13  "is_active": true,
14  "courses": [
15    {
16      "id": "5e341a7d-05de-4cef-9385-7b1b22a09dd6",
17      "title": "Продвинутый курс по фигма",
18      "term": 2,
19      "oum": "MONTH",
20      "is_active": true
21    },
22    {
23      "id": "5e341a7d-05de-4cef-9385-7b1b22a09dd2",
24      "title": "Продвинутый курс по фигма 2.0",
25      "term": 2,
26      "oum": "MONTH",
27      "is_active": true
28    },
29    {
30      "id": "5e341a7d-05de-4cef-9385-7b1b22a09dd3",
31      "title": "Базовый курс по фигма",
32      "end_at": "2012-12-12T10:10:10Z",
33      "is_active": false
34    }
35  ]
36 }
```

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 - 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 }
```

Требования к реализации

1. Получить запрос и проверить авторизацию пользователя - наличие заголовка Authorization в запросе:
 - а. Если авторизация не пройдена, то вернуть ошибку HTTP-401/1001 - "Пользователь не авторизован".
 - б. Иначе - далее.
2. Определить `user.id` по значению токена и проверить наличие соответствующего пользователя в БД:
 - а. Если такого пользователя нет, вернуть ошибку с кодом HTTP-404/3001 - "Пользователь не найден".
 - б. Иначе - далее.

3. Извлечь данные студента, страну проживания, курсы, к которым он подключён. **Вернуть успешный ответ HTTP-200.**

Обработка исключительных ситуаций:

1. Нет прав для просмотра.

Если пользователь пытается просмотреть чужой профиль, то вернуть ошибку с кодом HTTP-403/1002 - "Недостаточно прав для выполнения операции"

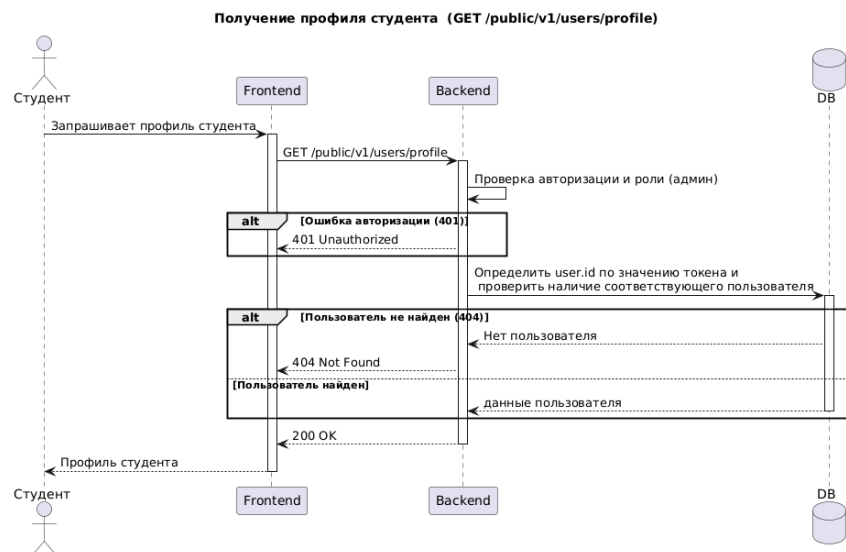
2. Ошибка при обращении к БД.

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".

3. Превышен лимит запросов.

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

UML Схема:



Код PlantUML

@startuml

title Получение профиля студента (GET /public/v1/users/profile)

actor "Студент" as Admin

participant "Frontend" as Frontend

participant "Backend" as API

database "DB" as DB

Admin → Frontend : Запрашивает профиль студента

activate Frontend

Frontend → API : GET /public/v1/users/profile

activate API

API → API : Проверка авторизации и роли (админ)

alt Ошибка авторизации (401)

API → Frontend : 401 Unauthorized

end

```

API → DB : Определить user.id по значению токена и \n проверить наличие соответствующего пользователя
activate DB
alt Пользователь не найден (404)
    DB → API : Нет пользователя
API → Frontend : 404 Not Found
else Пользователь найден
    DB → API : данные пользователя
    deactivate DB
end

API → Frontend : 200 OK
deactivate API
Frontend → Admin : Профиль студента
deactivate API

deactivate Frontend
@enduml

```

Маппинг данных

Для тела ответа:

Параметр в JSON	Описание / комментарий, требования к преобразованию после извлечения из БД или обработки	Обяз.	Тип данных JSON	В БД
id	UUID студента	+	string	user.id
last_name	Фамилия		string	user.last_name
first_name	Имя	+	string	user.first_name
birthday	Дата рождения (YYYY-MM-DD)		string	user.birthday
gender	Пол (0 - не указан, 1 - муж., 2 - жен.)	+	string	user.gender_id
city	Город		string	user.city
phone	Телефон	+	string	user.display_number
email	Email	+	string	user.email
about	Описание (О себе)		string	user.about

avatar_url	Ссылка на аватар	+	string (URL)	user.avatar_file_id
country	Страна студента		string	user.country_id
is_active	Флаг активности пользователя	+	boolean	user.is_active
courses	Список курсов студента		object	
courses.id	UUID курса		string	course.id
courses.title	Название курса		string	course.title
courses.term	Длительность доступа		int	student_course.access_period_term
courses.oum	Единицы срока доступа		string	student_course.access_period_oum
courses.is_active	Индикатор активности курса		boolean	course.is_active
courses.end_at	Дата окончания курса		string	course_flow.end_at

Автор: Даниэл Пянзин
Telegram: @DanielPyanzin