

[AO] Задача на REST API: Создание заявки на отказ от прохождения курса [POST /v1/refusals]




Проект платформы для дополнительного образования от GetAnalyst


Автор: Антонина Охтина

antoninaohtina@gmail.com


+7 (909) 950-15-96

 Ссылка на Jira

 Связанные документы
(внутренние)

 A Free Database Designer
for Developers and Analysts

- БД U-Verse

 [AO] Таблица БД - Заявка
на возврат - описание
таблицы БД refund

Оглавление:

- Общее описание
 - Доступы и ограничения
- Запрос
 - Пример запроса:
 - Тело запроса (Body)
 - Пример ответа - Успех
 - Требования к обработке ошибок
- Требования к реализации
- Маппинг данных
- Логирование и мониторинг

Общее описание

Метод предназначен для создания новой заявки на отказ от прохождения курса на платформе U-Verse пользователем-Студентом.

В результате использования метода в БД создается новая заявка на отказ, которая далее направляется на согласование Администратору.

Доступы и ограничения

Метод доступен только авторизованным пользователям-Студентам.

Запрос

Method type: POST

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

API name: public

API version: v1

Method name: /refusals

Authorization:

- Тип: Bearer Token
- Требуется всегда
- Роль: Студент

Headers:

- Используются общие для системы

Query-parameters:

- Нет

Пример запроса:

```
1 POST https://api.uverse.com/public/v1/refusals
```

Тело запроса (Body)

Пример:

```
1 {
2   "orderId": "60fba13a-fd7e-4f14-a056-c4618b04ba4d",
3   "reason": "Неблагоприятные финансовые обстоятельства",
4   "reasonComment": null,
5   "studentComment": "Очень жаль... но как-то так...",
6   "amount": 105340.00
7 }
```

Пример ответа - Успех

Код HTTP: 201 Created

Тело ответа (Body):

```
1 {
2   "id": "fe908133-1e43-485f-b45d-aba92d476760",
3   "number": 12,
4   "createdAt": "2025-10-15T17:35:04.046Z",
5   "status": "На согласовании",
6   "courseTitle": "Системный анализ",
7   "flowTitle": "2025-09",
8   "accessTerminationDate": "2027-06-15"
9 }
```

Требования к обработке ошибок

Каждый ответ - ошибка возвращает:

```
1 {
2   "code": "2001",
3   "message": "Пользователь не авторизован"
4 }
```

Тело ответа (Body):

Описание возможных ошибок - см. [\[АО\] Коды ошибок_В РАБОТЕ](#)

Требования к реализации

1. Аутентификация и авторизация.

Проверить заголовок `Authorization: Bearer <Token>`.

- Если заголовок отсутствует: вернуть ошибку **HTTP-401** + код 2001.
- Если токен имеет неверный формат: вернуть ошибку **HTTP-401** + код 2002.
- Если токен не соответствует ни одному из пользователей или устарел: вернуть ошибку **HTTP-401** + код 2003.
- Если токен верный и роль пользователя `role <> 'student'`, то вернуть ошибку **HTTP-403** + код 2004.
- Иначе - далее.

2. Проверить формат JSON-тела запроса:

- `order_id` - обязателен, uuid.

- проверить, что в БД нет ранее созданных заявок на отказ (кроме находящихся в статусе “Отозвана” или “Отклонена”) к заказу.

Для `orderId`: искать `order_id` в таблице `refund`.

Если найден, проверить статус: `refund.status_id = '8aa04d59-a48e-480b-b4a3-06d45a0d842e'` (“Отозвана”) или `refund.status_id = '7e92dcca-61aa-4cef-add7-282e7be30679'` (“Отклонена”).

Если статус найденной заявки отличный от указанных: вернуть ошибку **HTTP-409** + код 1392.

- Возникновение любой ошибки при обращении к БД - вернуть ошибку **HTTP-500** + код 5012.
- `reason` - обязателен, строка, значение должно соответствовать одному из:
 - Курс не соответствует ожиданиям;
 - Возникли технические проблемы;
 - Неблагоприятные финансовые обстоятельства;
 - Иное

Если значение отличается, то вернуть ошибку **HTTP-400** + код 1331.

- `reasonComment` - обязателен, если `reason = "Иное"`, строка от 3 до 512 символов.

Если при `reason = "Иное"` значение отсутствует или составляет менее 3 символов, вернуть ошибку **HTTP-400** + код 1332.

- `studentComment` - необязателен, строка от 0 до 1024 символов.

- `amount` - обязателен, числовое значение.

- проверить корректность полученной суммы - сравнить с рассчитанным на основе данных связанных таблиц (`student`, `flow`, `course`, `terms`, `payment`, `module`, `lesson`, `student_lesson`) значением.

Полученное расчетное значение перед сравнением перевести из копеек в рубли (в случае введения новой валюты, при переводе учитывать самое маленькое ее значение). Округлять по стандартным правилам.

Если расчетное значение отличается от полученного, вернуть ошибку **HTTP-400** + код 1333.

Формула для расчета:

$$R = M - E - (P / N1) \times N2$$

где:

Обозначение	Поле(я) в БД	Описание	Связь между таблицами
M	Сумма <code>payment.amount</code>	Сумма всех успешных платежей по заказу (т.е. <code>payment.status_id = '235ed000-9ef8-4fd8-9ae7-4cbd8142d840'</code>)	{orderId} = <code>payment.order_id</code>
E	Сумма (<code>payment.amount</code> - <code>payment.income_amount</code>) + <code>payment.receive_fee</code>	Сумма комиссий по всем платежам по заказу: комиссия за платеж (= сумма платежа - сумма фактически поступившая) + комиссия за чек	{orderId} = <code>payment.order_id</code>
P	<code>terms.price</code>	Базовая цена курса	{orderId} = <code>order.id</code> → <code>order.terms_id</code> = <code>terms.id</code>
N1	Количество <code>lesson.id</code>	Количество всех уроков в курсе	{orderId} = <code>order.id</code> → <code>order.flow_id</code> = <code>flow.id</code> → <code>flow.course_id</code> = <code>course.id</code> → <code>course.id</code> = <code>module.course_id</code> → <code>module.id</code> = <code>lesson.module_id</code>
N2	Количество <code>student_less</code>	Количество всех уроков в курсе, пройденных Студентом	{orderId} = <code>order.id</code> → <code>order.flow_id</code> =

	on.id	(student_lesson.status = 'Просмотрен')	flow.id → flow.course_id = course.id → course.id = module.course_id → module.id = lesson.module_id → lesson.id = student_lesson.lesson_id и {orderId} = order.id → order.student_id = student_id → student.id = student_lesson.student_id
--	-------	--	---

3. Создать запись о новой заявке на отказ в таблице БД **refund** со статусом “На согласовании”

(**status.id** = 'd037f439-45ca-4b0b-8321-5c389edabdbc'):

- Присвоить уникальный идентификатор **id** - случайно сгенерированный UUID.
- Присвоить заявке порядковый номер **number** - целое число, присвоить значение, следующее за номером предшествующей записи (автоинкремент).
- Указать дату и время создания заявки **created_at** в момент записи в БД.

Любая ошибка при сохранении в БД - вернуть ошибку **HTTP-500** + код 5011.

4. Сформировать ответ с данными о созданной заявке и вернуть с **HTTP-201 Created** JSON.

Требования к соответствию JSON и БД приведены в разделе Маппинг данных.

5. Отправить уведомление Администратору о создании новой заявки на отказ - см. <Отправка уведомлений>.

6. Ограничить доступ Студента к курсу

⚠ Требуется дополнительной проработки



Проект платформы для дополнительного
образования от GetAnalyst

Автор: Антонина Охтина

antoninaohtina@gmail.com

+7 (909) 950-15-96

Маппинг данных

Для тела запроса:

Параметр в JSON	Тип данных в JSON	Поле(я) в БД	Описание / комментарий, требования к валидации при приеме на обработку
⊖		id	Уникальный идентификатор заявки на возврат - случайно сгенерированный UUID Обязательное uuid
orderId	string (uuid)	refund.order_id	Идентификатор заказа, к которому создается заявка на возврат Обязательное uuid
⊖		number	Номер заявки для отображения на UI при просмотре заявки и журнала заявок Обязательное integer, автоинкремент
⊖		created_at	Дата и время создания заявки на отказ Обязательное timestamp
⊖		status_id	Идентификатор статуса, соответствующий значению “На согласовании” для типа сущности entity_type.name = 'refund' (status.id = 'd037f439-45ca-4b0b-8321-5c389edabdbc') Обязательное uuid
reason	string	refund.reason	Причина отказа от прохождения курса Обязательное varchar[64], значение из предустановленного списка
reasonComment	string	refund.reason_comment	Комментарий к причине отказа, если выбран вариант “Иное” Опционально. Обязательное, если reason = "Иное" varchar[512]

<code>studentComment</code>	string	<code>refund.comment</code>	Свободный комментарий студента Опционально varchar[1024]
<code>amount</code>	number	<code>refund.amount</code>	Предварительная сумма возврата, рассчитанная при создании заявки Обязательное integer, сумма переводится в копейки (в случае введения новой валюты, будет использоваться самое маленькое ее значение)

Для тела ответа:

Параметр в JSON	Тип данных в JSON	Обязательность	Описание / комментарий, требования к преобразованию после извлечения из БД или обработки	Поле(я) в БД
<code>id</code>	string (uuid)	Да	Идентификатор созданной заявки на отказ от прохождения курса	<code>refund.id</code>
<code>number</code>	integer	Да	Номер заявки на отказ	<code>refund.number</code>
<code>createdAt</code>	string (timestamp)	Да	Дата создания заявки	<code>refund.created_at</code>
<code>status</code>	string	Да	Статус заявки Определять через связку <code>refund.status_id</code> → <code>status.id</code>	<code>status.name</code>
<code>flowTitle</code>	string	Да	Название потока Определять через связку <code>refund.order_id</code> → <code>order.id</code> , <code>order.flow_id</code> → <code>flow.id</code>	<code>flow.title</code>
<code>courseTitle</code>	string	Да	Название курса Определять через связку <code>refund.order_id</code> → <code>order.id</code> ,	<code>course.title</code>

			<code>order.flow_id → flow.id ,</code> <code>flow.course_id →</code> <code>course.id</code>	
<code>accessTerminationDate</code>	string	Да	<p>Дата прекращения доступа к пройденным урокам</p> <p>Определять через связку</p> <code>refund.order_id →</code> <code>order.id ,</code> <code>order.flow_id → flow.id ,</code> <code>flow.course_id →</code> <code>course.id</code> <p>Если поток проводится по расписанию (<code>flow.type = 'По расписанию'</code>), то дату окончания потока увеличивать на период доступа к материалам курса;</p> <p>иначе (т.е. <code>flow.type = 'Свободный'</code>), дату предоставления доступа к материалам курса увеличивать на срок обучения и период доступа к материалам курса</p>	<code>flow.endDate +</code> <code>course.access_period</code> или <code>order.access_date</code> <code>+</code> <code>course.duration +</code> <code>course.access_period</code>

Логирование и мониторинг

Общие требования к логированию запросов REST API системы. Особенности нет.



Проект платформы для дополнительного образования от GetAnalyst

Автор: Антонина Охтина

antoninaohtina@gmail.com

+7 (909) 950-15-96