

[ГХ] 7.1 Возврат платежа в панели администратора

Оглавление

Общее описание
Известные ошибки, особенности и ограничения
Макеты UI
Сценарий работы
Роли пользователей
Системы
Внешние системы
Входные данные
Предусловия
Алгоритм работы
Результат
Диаграмма UML
Альтернативные сценарии и ошибки
Маппинг данных
Логирование

Общее описание

Для администраторов TWM необходимо реализовать функциональность просмотра и возврата платежей пользователей соц. сети TWM в веб-приложении “Панель администратора”.

В разделе “Платежи” отображается список всех произведённых платежей с основной информацией: номер платежа, сумма, валюта, статус и описание.

Администратор может перейти на страницу платежа, где отображается расширенная информация.

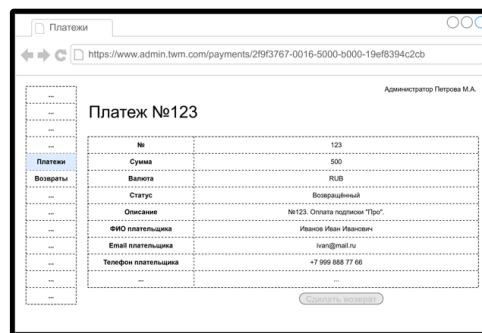
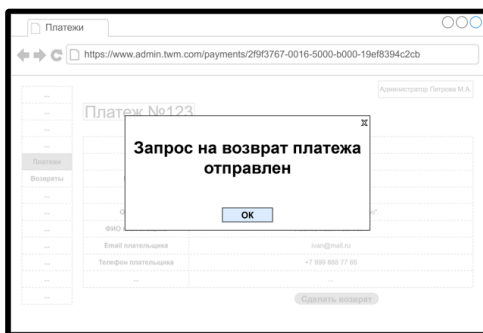
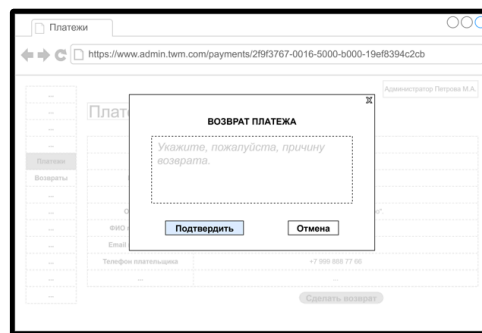
На странице платежа доступна функция возврата. После нажатия кнопки “Сделать возврат” выводится модальное окно для указания причины возврата и подтверждения действия. В случае подтверждения, система инициирует возврат платежа через ЮКасса. После успешного возврата активная подписка “Про” пользователя деактивируется, и пользователь теряет доступ к платным функциям.

Известные ошибки, особенности и ограничения

- Возврат возможен только по платежам со статусом “Успешный” (“succeeded”)
- Возможность частичного возврата не реализуется (в рамках этой задачи)

- Инициировать возврат можно только вручную через “Панель администратора”
- Возможность отмены возврата не предусмотрена (отменить возврат нельзя)

Макеты UI



Сценарий работы

Роли пользователей

- Администратор

Системы

- Веб-приложение “Панель администратора”
- Сервер-приложение TWM (Backend)

Внешние системы

- ЮКасса

Входные данные

- Уникальный идентификатор платежа в TWM
- Причина возврата

Предусловия

- Пользователь соц. сети TWM ранее оплатил подписку “Про”
- Платёж имеет статус “Успешный” (succeeded)
- Администратор авторизован в “Панели администратора”
- Администратор перешёл в раздел “Платежи” и открыл страницу платежа

Алгоритм работы

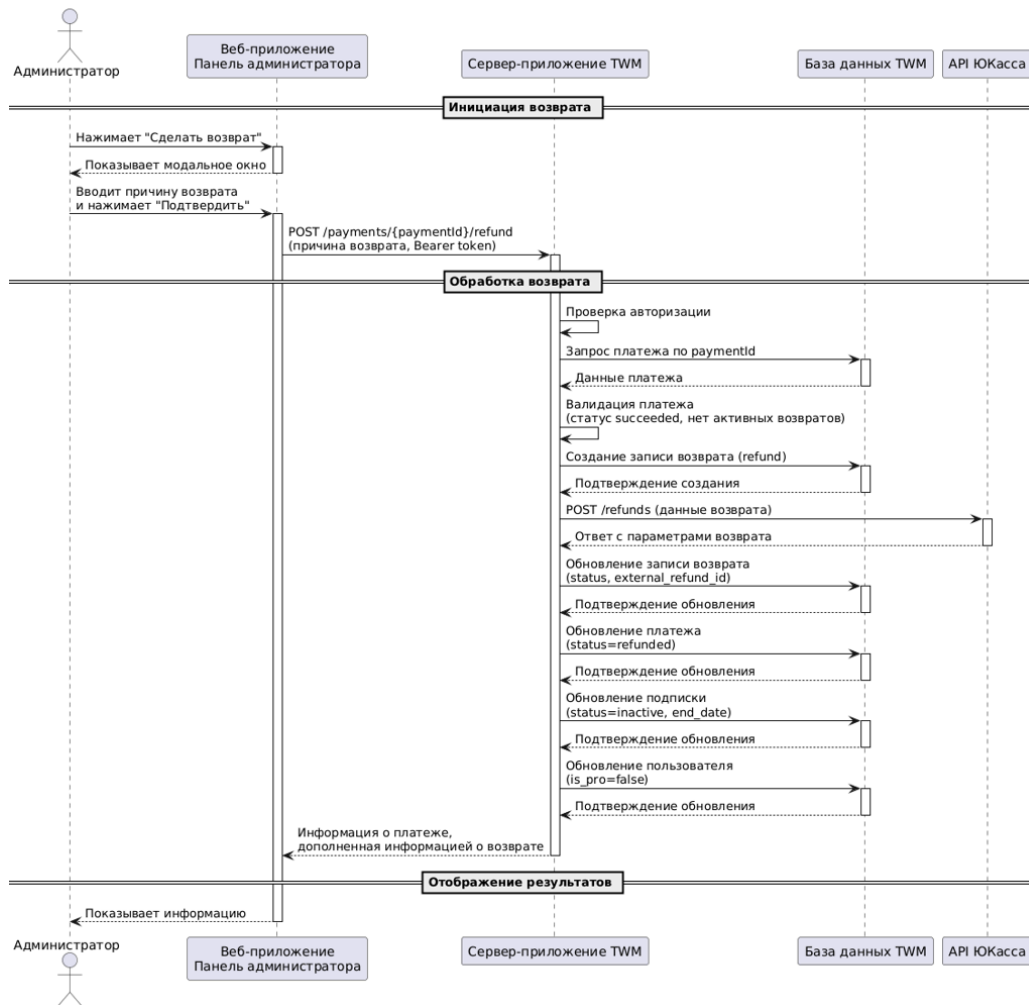
1. **Администратор** нажимает кнопку “Сделать возврат”
2. В открывшемся модальном окне **администратор** указывает причину возврата и нажимает “Подтвердить”
3. **Frontend TWM** отправляет запрос на создание возврата POST /payments/{paymentId}/refund на **Backend TWM**

4. **Frontend TWM** получает информацию о платеже, дополненной информацией о возврате, от метода POST /payments/{paymentId}/refund
5. **Frontend TWM** отображает информацию на экране

Результат

- Возврат платежа пользователю выполнен успешно
- Подписка “Про” пользователя деактивирована

Диаграмма UML



Альтернативные сценарии и ошибки

1А. Если ошибка соединения с интернет:

- Отобразить текст предупреждения наверху экрана “Нет соединения. Проверьте подключение к интернету и попробуйте снова.”
- Модальное окно не открывается

2А. Если ошибка соединения с интернет:

- Отобразить текст предупреждения наверху экрана “Нет соединения. Проверьте подключение к интернету и попробуйте снова.”
- Запрос на создание возврата на Backend TWM не отправляется

3А. Если ошибка подключения к серверу:

- Отобразить текст предупреждения наверху экрана “Сервер временно недоступен. Повторите попытку позже.”
- Запрос на создание возврата на Backend TWM не отправляется

Маппинг данных

Экран с расширенной информацией о платеже

На UI	В API TWM (URL/JSON)	Комментарий
-	payment.id	Идентификатор платежа в TWM
Значение поля в модальном окне “Возврат платежа”	reason	Причина возврата

Статусы платежа

На UI	В БД	Комментарий
В ожидании	pending	
Успешный	succeeded	
Неуспешный	failed	
Отменённый	canceled	
Возвращённый	refunded	

Статусы возврата

На UI	В БД	Комментарий
pending	В ожидании	
succeeded	Успешный	
failed	Неуспешный	
canceled	Отменённый	

Логирование

Отдельные требования не предъявляются. Использовать общие по системе.

[ГХ] 7.1.1 POST /payments/{paymentId}/refund - Создание возврата в ЮКасса

Оглавление

Общее описание

Метод создаёт возврат успешного платежа в ЮКасса и возвращает обновленную информацию о платеже, дополненной информацией о возврате.

Документация внешней системы

[🔗 Справочник API ЮKassa](#)

Особенности и ограничения

Для обеспечения идемпотентности в заголовке Idempotence-Key передается ключ идемпотентности:

- если повторить запрос с теми же данными и тем же ключом, ЮКасса обрабатывает его как повторный;
- если повторить запрос с теми же данными, а ключ идемпотентности отличается, ЮКасса обрабатывает его как новый.

Ограничение на кол-во запросов не установлено по документации ЮКасса.

Сценарий работы (Use Case)

Системы

- Backend TWM
- API ЮКасса
- БД

Входные данные

- Уникальный идентификатор платежа в TWM
- Причина возврата

Предусловия

- Администратор указывает причину и подтверждает возврат платежа

- С “Панели администратора” отправляется запрос на создание возврата POST /payments/{paymentId}/refund на Backend TWM

Алгоритм работы

1. **Backend TWM** получает запрос на создание возврата POST /payments/{paymentId}/refund
2. **Backend TWM** проверяет авторизацию пользователя (Bearer)
3. **Backend TWM** проверяет по полученному идентификатору платежа paymentId (совпадает со значением payment.id БД), что:
 - a. платёж в таблице БД payment есть
 - b. платёж в таблице БД payment имеет статус “succeeded”
 - c. отсутствует возврат по данному платежу в таблице БД refund в статусах “pending” или “succeeded”
4. **Backend TWM** создаёт запись о возврате в таблице БД refund с параметрами:
 - a. id - Идентификатор возврата, генерируется уникальное значение, используется в качестве idempotence_key для запроса в ЮКасса
 - b. number - Номер возврата, генерируется уникальное значение, max(number)+1
 - c. payment_id - Идентификатор платежа, устанавливается значение paymentId из запроса (совпадает со значением payment.id БД)
 - d. status - Статус возврата, сначала присваивается “pending”
 - e. reason - Причина возврата, устанавливается значение reason из запроса
 - f. external_refund_id - Идентификатор платежа в ЮКасса, сначала не заполняется
 - g. created_at - Время создания записи, устанавливается текущее время создания
 - h. updated_at - Время изменения записи, сначала совпадает со значением created_at
 - i. refund_at - Время успешного возврата, сначала не заполняется
5. **Backend TWM** вызывает метод **ЮКасса** для создания возврата POST /refunds

Пример запроса:

```
1 curl 'https://api.yookassa.ru/v3/refunds' \  
2 -H 'Idempotence-Key: 045196a8-b702-4755-a588-cf9d6fd8a1b4' \  
3 -H 'Content-Type: application/json' \  
4 -H 'Authorization: Basic XXXXX' \  
5 -d '{  
6   "payment_id": "2fec8be1-000f-5000-8000-15819b3d5329",  
7   "amount": {  
8     "value": "628.27",  
9     "currency": "RUB"  
10  }  
}
```



```
11 }'
```

6. **ЮКасса** возвращает ответ с параметрами возврата

Пример ответа:

```
1 {
2   "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
3   "payment_id": "2fec8be1-000f-5000-8000-15819b3d5329",
4   "status": "succeeded",
5   "created_at": "2025-06-30T18:21:46.002Z",
6   "amount": {
7     "value": "628.27",
8     "currency": "RUB"
9   }
10 }
```

7. **Backend TWM** обновляет запись о возврате в таблице БД refund по полученным параметрам:

- a. status - меняется с “pending” на значение status из ответа ЮКасса “succeeded”
- b. external_refund_id - устанавливается значение id из ответа ЮКасса
- c. updated_at - меняется на текущее время изменения
- d. refund_at - устанавливается значение created_at из ответа ЮКасса

8. **Backend TWM** обновляет запись о платеже в таблице БД refund

- a. status - Статус платежа, меняется с “succeeded” на “refunded”
- b. updated_at - Время изменения записи, меняется на текущее время изменения

9. **Backend TWM** обновляет запись о подписке в таблице БД subscription

- a. status - Статус подписки, меняется с “active” на “inactive”
- b. end_date - Дата окончания подписки, устанавливается значение refund.refund_at БД
- c. updated_at - Время изменения записи, меняется на текущее время изменения

10. **Backend TWM** обновляет запись о пользователе в таблице БД user

- a. is_pro - Признак “Подписка Про”, меняется с “true” на “false”
- b. updated_at - Время изменения записи, меняется на текущее время изменения

11. **Backend TWM** возвращает информацию о платеже, дополненной информацией о возврате, на **Frontend TWM**

Результат

- Есть запись о возврате, статус возврата “succeeded”
- Статус платежа “refunded”
- Статус подписки “inactive”

- Признак “Подписка Про” у пользователя “false”

Альтернативные сценарии и обработка ошибок

2А. Если отсутствует авторизация:

- Вернуть ошибку “Пользователь не авторизован.” (HTTP 401)
- Зафиксировать событие в журнале ошибок

2В. Если нет прав доступа:

- Вернуть ошибку “Доступ запрещён.” (HTTP 403)
- Зафиксировать событие в журнале ошибок

3А. Если платёж в БД не найден:

- Вернуть ошибку “Платеж не найден.” (HTTP 404)
- Зафиксировать событие в журнале ошибок

3В. Если платёж в БД имеет статус, отличный от “succeeded”:

- Вернуть ошибку “Возврат невозможен для платежа со статусом: [текущий статус платежа].” (HTTP 400)
- Зафиксировать событие в журнале ошибок

3С. Если существует возврат по данному платежу в БД в статусах “pending” или “succeeded”:

- Вернуть ошибку “Возврат по данному платежу уже существует.” (HTTP 400)
- Зафиксировать событие в журнале ошибок

4А. Если не удалось создать запись о возврате в БД:

- Повторить попытку создания записи (макс. 3 попытки)
- При повторной ошибке:
 - Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
 - Зафиксировать событие в журнале ошибок
 - Отправить уведомление в систему мониторинга

5А. Если не удалось вызвать метод ЮКасса:

- Повторить попытку вызова метода (макс. 3 попытки)
- При повторной ошибке:
 - Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)

- Зафиксировать событие в журнале ошибок
- Отправить уведомление в систему мониторинга

6A. Если ЮКасса возвращает ошибку:

- Сделать повторный запрос (макс. 3 попытки)
- При повторной ошибке:
 - Обновить статус возврата в таблице БД refund на “failed”
 - Вернуть пользователю: “Извините, произошла непредвиденная ошибка. Повторите попытку позже.” (HTTP 502)
 - Зафиксировать событие в журнале ошибок с деталями запроса/ответа
 - Отправить уведомление в систему мониторинга

6B. Если ЮКасса возвращает статус возврата “pending”:

- Значение status менять не требуется (т.к. в БД уже такой статус)
- Установить в external_refund_id значение id из ответа ЮКасса
- Изменить значение updated_at на текущее время изменения
- Вернуть пользователю текущий статус возврата
- Отправить уведомление в систему мониторинга

6C. Если ЮКасса возвращает статус возврата “canceled”:

- Значение status изменить на значение status из ответа ЮКасса “canceled”
- Установить в external_refund_id значение id из ответа ЮКасса
- Изменить значение updated_at на текущее время изменения
- Вернуть пользователю текущий статус
- Отправить уведомление в систему мониторинга

7A. Если не удалось обновить запись о возврате в БД:

- Повторить попытку обновления (макс. 3 попытки)
- При повторной ошибке:
 - Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
 - Зафиксировать расхождение данных в журнале ошибок
 - Отправить уведомление в систему мониторинга

7B. Если запись о возврате в БД имеет статус, отличный от “pending”:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)

- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

7С. Если запись о возврате в БД имеет заполненные значения external_refund_id и/или refund_at:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

8А. Если не удалось обновить запись о платеже в БД:

- Повторить попытку обновления (макс. 3 попытки)
- При повторной ошибке:
 - Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
 - Зафиксировать расхождение данных в журнале ошибок
 - Отправить уведомление в систему мониторинга

8В. Если запись о платеже в БД имеет статус, отличный от “succeeded”:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

9А. Если не удалось обновить запись о подписке в БД:

- Повторить попытку обновления (макс. 3 попытки)
- При повторной ошибке:
 - Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
 - Зафиксировать расхождение данных в журнале ошибок
 - Отправить уведомление в систему мониторинга

9В. Если запись о подписке в БД имеет статус “inactive”:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

10А. Если не удалось обновить запись о пользователе в БД:

- Повторить попытку обновления (макс. 3 попытки)
- При повторной ошибке:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

10B. Если значение “is_pro” записи о пользователе в БД уже соответствует “false”:

- Вернуть ошибку “Внутренняя ошибка сервера.” (HTTP 500)
- Зафиксировать расхождение данных в журнале ошибок
- Отправить уведомление в систему мониторинга

Техническая реализация метода

Запрос

Тип метода: POST

URL: <https://api.tripwithme.com/public/v1.0/payments/{paymentId}/refund>

Авторизация: Basic Auth

Headers:

- Content-Type
- Authorization

Query params: -

Body:

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

Пример запроса:

```
1 curl 'https://api.tripwithme.com/public/v1.0/payments/{paymentId}/refund' \  
2 -H 'Content-Type: application/json' \  
3 -H 'Authorization: Basic XXXXX' \  
4 -d '{  
5   "reason": "Причина возврата"  
6 }'
```

Ответ

Успешный ответ

HTTP 201

Пример тела ответа:

```
1 {
2   "id": "66d5fc4c-e3b0-4c0b-b63c-a59b971c3aff",
3   "number": 1,
4   "status": "succeeded",
5   "reason": "Причина возврата",
6   "refundAt": "2025-06-30T18:21:46.002Z"
7 }
```

Обработка ошибок

Причина возникновения ошибки указана в теле ответа, в параметре error.description.

HTTP 400

Возвращается, если:

- в запросе не переданы обязательные параметры данные
- в запросе переданы некорректные данные (неверный формат, недопустимое значение)
- возврат по данному платежу уже существует
- возврат запрашивается для платежа со статусом, отличным от “succeeded”

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
4     "description": "Возврат по данному платежу уже существует."
5   }
6 }
```

HTTP 401

Возвращается, если отсутствует авторизация

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
4     "description": "Пользователь не авторизован."
5   }
6 }
```

HTTP 403

Возвращается, если нет прав доступа

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
```

```
4     "description": "Доступ запрещён."
5   }
6 }
```

HTTP 404

Возвращается, если платежа, для которого запрашивается возврат, не существует

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
4     "description": "Платёж не найден."
5   }
6 }
```

HTTP 500

Возвращается, если возникла ошибка в работе Backend TWM

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
4     "description": "Внутренняя ошибка сервера."
5   }
6 }
```

HTTP 502

Возвращается, если ЮКасса возвращает ошибку

Пример тела ответа:

```
1 {
2   "error": {
3     "id": "2f9f3767-0016-5000-b000-17ef8394c2cb",
4     "description": "Извините, произошла непредвиденная ошибка. Повторите попытку позже."
5   }
6 }
```

Маппинг данных

Для запроса:

В URL/JSON TWM	В БД TWM	Тип данных	В URL/JSON ЮКасса	Комментарий

paymentId	payment.id refund.payment_id	uuid	-	Идентификатор платежа в TWM
reason	refund.reason	text	-	Причина возврата
-	refund.idempotence_key	uuid	Idempotence-Key	Ключ идемпотентности
-	payment.external_payment_id	uuid	payment_id	Идентификатор платежа в ЮКасса
-	payment.amount	decimal	amount.value	Сумма в выбранной валюте
-	payment.currency	varchar	amount.currency	Трехбуквенный код валюты

Для ответа:

В JSON ЮКасса	В БД TWM	Тип данных	В JSON TWM	Комментарий
id	refund.external_refund_id	uuid	-	Идентификатор возврата в ЮКасса
payment_id	payment.external_payment_id	uuid	-	Идентификатор платежа в ЮКасса
status	refund.status	varchar	refund.status	Статус возврата

created_at	refund.refund_at	timestamp	refund.refundAt	Время успешного возврата
amount.value	payment.amount	decimal	-	Сумма в выбранной валюте
amount.currency	payment.currency	varchar	-	Трехбуквенный код валюты
-	refund.id	uuid	refund.id	Идентификатор возврата в TWM
-	refund.number	integer	refund.number	Номер возврата
-	refund.reason	text	refund.reason	Причина возврата

Логирование

Требования к логированию (Kibana)

- Запросы на создание возврата в TWM
 - a. paymentId
 - b. reason
- Создание возвратов в TWM
 - a. id - идентификатор возврата
 - b. number - номер возврата
 - c. payment_id - идентификатор платежа
 - d. idempotence_key - ключ идемпотентности
 - e. reason - причина возврата
 - f. created_at - время создания
- Вызовы метода ЮКасса для создания возврата

- a. URL запроса
 - b. Idempotence-Key
 - c. тело запроса
 - d. HTTP-статус и тело ответа
- Ошибки
 - a. Ошибки на уровне TWM (ошибки валидации, БД, сервера)
 - b. Ошибки взаимодействия с ЮКасса
 - c. Ошибки от ЮКасса (HTTP-статус и тело ответа)

Метрики для мониторинга (Grafana)

- Количество успешно созданных возвратов в TWM в сутки
- Количество успешно созданных возвратов в ЮКасса в сутки
- Количество ошибок на уровне TWM в сутки
- Количество ошибок от ЮКасса в сутки (по группам 400, 401, 403, 404, 405, 415, 429, 500)
- Количество попыток вызова метода ЮКасса для создания возврата с повторным Idempotence-Key в сутки
- Среднее время отклика от ЮКасса посуточно

Уведомления в систему мониторинга

- При повторных попытках подряд более 3 раз
- Более 5% ошибок при вызове ЮКасса за 10 минут