

Интеграция с платежной системой Raif Pay для U-Verse


Автор: Даниэл Пянзин
Telegram: @DanielPyanzin

История изменений:

Связанные документы:

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

 - БД проекта

 [U-Verse](#) - Figma проекта

Оглавление:

Общее описание

- API-документация Внешней системы

- Особенности и известные проблемы:

Сценарий работы (Use Case)

- Алгоритм работы:

- Альтернативные сценарии и обработка ошибок:

- UML-Sequence процесса оплаты

Интеграционные методы API Backend

- Создание платежа в Raif Pay

- Общее описание

- Запрос

- Успешный ответ

- Требования к обработке ошибок

- Пример ответа - Ошибка валидации входных данных

- Пример ответа - Ошибка авторизации

- Пример ответа - Ошибка прав доступа

- Пример ответа - Превышен лимит запросов

- Пример ответа - Ошибка запроса к Raif Pay (400/403/500) + Raif Pay недоступен (таймаут)

- Пример ответа - Ошибка базы данных

- Маппинг данных

- Получение статуса платежа в Raif Pay

- Общее описание

- Альтернативные сценарии и обработка ошибок:

- Запрос

- Успешный ответ

- Требования к обработке ошибок

- Пример ответа - Ошибка валидации входных данных

- Пример ответа - Ошибка авторизации

- Пример ответа - Ошибка прав доступа

- Пример ответа - Превышен лимит запросов

[Пример ответа - Raif Pay недоступен \(таймаут\)](#)

[Пример ответа - Ошибка базы данных](#)

[Маппинг данных](#)

[Обработка WebHook от Raif Pay](#)

[Маппинг данных UI - API нашей системы](#)

[1. Экран “Оплаты заказа”](#)

[2. Экран “Ожидания оплаты”](#)

[3. Экран “Результат оплаты”](#)

[4. Экран “Ошибка”](#)

[Логирование и мониторинг](#)

Общее описание

В этой статье описан процесс онлайн-оплаты курсов в приложении U-Verse с помощью платёжного шлюза Raif Pay.

Что это:

Интеграция между приложением U-Verse и системой интернет-эквайринга Raif Pay для безопасного приёма платежей.

Зачем:

Чтобы безопасно и предсказуемо принимать оплату за курсы без хранения платёжных данных и обеспечить корректную выдачу доступа к курсу сразу после подтверждения оплаты.

Для кого:

- Для студентов U-Verse, желающих быстро и удобно купить курсы онлайн.
- Для преподавателей, которые хотят продавать свои курсы.
- Для владельцев U-Verse, которые хотят получать доход от платформы.

API-документация Внешней системы

 [Raiffeisenbank e-commerce API](#)

Особенности и известные проблемы:

1. Перед работой над задачей должны быть настроены:
 - а. Конфигурация системы.
 - б. Общая авторизация на запросы к Raif Pay.
2. Возврат платежей (после оплаты) пока не будет поддержан - внесено в Backlog. Возврат будет делаться ответственными сотрудниками вручную, через ЛК Raif Pay.
3. Отмена заказов для которых создан платеж, но они не оплачены (нажатие кнопок “Назад” и “Отмена”) - будет реализовано в рамках отдельной задачи. Внесено в Backlog.

Сценарий работы (Use Case)

Роли пользователей:

- Студент

Системы:

- Frontend U-Verse (веб-приложение)
- Backend U-Verse
- БД U-Verse
- Интернет-эквайринг от Raif Pay

Предусловие:

- Пользователь авторизован в системе.
- Пользователь выбрал курс и указал тариф.
- Сформирован заказ со статусом Новый.
- Пользователь находится на экране просмотра данных заказа.
- Пользователь перед оплатой имеет доступ к публичной оферте, политике обработки персональных данных и правилами использования платформы, а также ссылки на инструкции к часто задаваемым вопросам:
 - Как оплатить курс?
 - Оплатить от компании

Алгоритм работы:

1. **Пользователь** оформляет заказ в U-Verse и нажимает «Оплатить».
2. **Frontend** отправляет запрос POST /orders/payments на Backend U-Verse для создания платежа к заказу.
3. **Backend U-Verse**
 - а. По идентификатору **"order"** проверяет в БД, что текущий статус заказа new (новый).
4. **Backend U-Verse** отправляет запрос в Raif Pay на регистрацию нового заказа: POST <https://pay.raif.ru/api/v1/merchants/{publicId}/orders>.
5. Raif Pay возвращает в ответ:

- a. `"payformUrl"` - ссылка на безопасную платежную форму Raif Pay, чтобы U-Verse перенаправил на не своего пользователя для совершения оплаты
Все данные банковской карты или другого средства платежа пользователь вводит на стороне системы Raif Pay.
Система U-Verse доступа к данным банковской карты пользователя иметь не будет.
6. **Backend U-Verse** записывает `"payformUrl"` в БД, а также обновляет статус платежа в БД на ожидание оплаты: `status = pending`.
7. **Backend** U-Verse возвращает на Frontend полную информацию о заказе и платеже, включая адрес платежной формы, полученный от Raif Pay, на который надо направить пользователя для оплаты.
8. **Frontend** делает редирект (перенаправление) пользователя на платежную форму Raif Pay по полученному в ответе url платежной формы - `paymentURL` в ответе от API Backend U-Verse
9. Пользователь вводит платёжные данные на стороне Банка и подтверждает платёж. Frontend и Backend U-Verse не имеют доступа к данным платежной формы.
10. После того, как пользователь завершил оплату, Frontend проверяет статус оплаты:
 - Raif Pay сообщает Backend о результате платежа (Webhook - [Raiffeisenbank e-commerce API](#))
 - валидируем подпись (Для проверки подлинности уведомления по оплате к данным добавляется подпись в заголовке `x-api-signature-sha256`, полученная на основе общего секретного ключа и контрольной строки
`amount|publicId|order|transaction.status.value|transaction.status.date` с помощью HMAC-SHA-256.)
 - сверяем сумму платежа с ожидаемой
 - Frontend проверяет статус платежа через API Backend U-Verse: GET `/orders/payments/status`.

В случае **успеха**: заказ в U-Verse переводится в `paid`, фиксируются атрибуты платежа, публикуются доменные события (уведомления/доступ к курсам).

В случае **неуспеха** — остаётся `pending / failed` с возможностью повтора.

11. Frontend получает актуальные статусы заказа и платежа от Backend U-Verse и отображает итоговое сообщение с результатами оплаты для пользователя:
 - a. Если успех, то экран с подтверждением оплаты: Курс доступен в личном кабинете.
 - b. Если ошибка, то предложения: повторить оплату или вернуться в режим просмотра и редактирования заказа.

Альтернативные сценарии и обработка ошибок:

1A. Backend U-Verse недоступен - нет ответа в течение 30 секунд.

Показать пользователю текст ошибки:

1 "Сервис временно недоступен. Попробуйте ещё раз"

2A. Нет доступа к сети Интернет.

Показать пользователю текст ошибки:

1 "Проверьте соединение с Интернет"

2Б. Превышен лимит запросов.

Если превышен лимит запросов по пользователю/IP, то вернуть ошибку HTTP-429/1005

3Б. Заказ в ином статусе, не new :

- **pending** : Вернуть ссылку на платежную форму ранее сохраненную для заказа, если заказ не был измен.
Удалить id внешней системы и ссылку на платежную форму из БД, если заказ был измен, чтобы создать новую , перейти к шагу 4 основного сценария.
- **paid** - если заказ уже оплачен или возвращен, вернуть HTTP-409/3012.

3.В Аккаунт студента заблокирован.

Если пользователь заблокирован (user.is_active=false) прекратить обработку и вернуть ошибку HTTP-403/1003.

4A. Получена ошибка валидации в Raif Pay - HTTP 400 Bad Request.

Backend U-Verse:

- Должен записать в лог информацию о неуспешном запросе с приоритетом "критическая" (т.к. это нестандартная ситуация для работы интеграции Raif Pay, в нем ключ API постоянный).
- Вернуть ошибку HTTP-502/4002 в ответ на запрос.

4Б. Превышено время ожидания ответа от Raif Pay - 20 секунд.

Backend U-Verse:

- Должен записать в лог информацию о неуспешном запросе с приоритетом "критическая".
- Backend U-Verse должен вернуть ошибку HTTP-502/4002 в ответ на запрос.

4В. Получена ошибка валидации в Raif Pay - HTTP 403 Forbidden.

Backend U-Verse:

- Должен записать в лог информацию о неуспешном запросе с приоритетом “критическая” (т.к. это нестандартная ситуация для работы интеграции Raif Pay, в нем ключ API постоянный).
- Вернуть ошибку HTTP-502/4002 в ответ на запрос.

4Г. Получена иная ошибка от Raif Pay.

Backend U-Verse:

- Должен записать в лог информацию о неуспешном запросе с приоритетом “критическая”.

6А. Ошибка при обращении к БД.

В процессе работы алгоритма не удается подключиться к БД. Вернуть пользователю код ошибки HTTP-500/5002.

10А. Неверная подпись webhook.

Статус платежа не меняется, фиксируем инцидент с приоритетом "предупреждение".

10А. Несовпадение сумм.

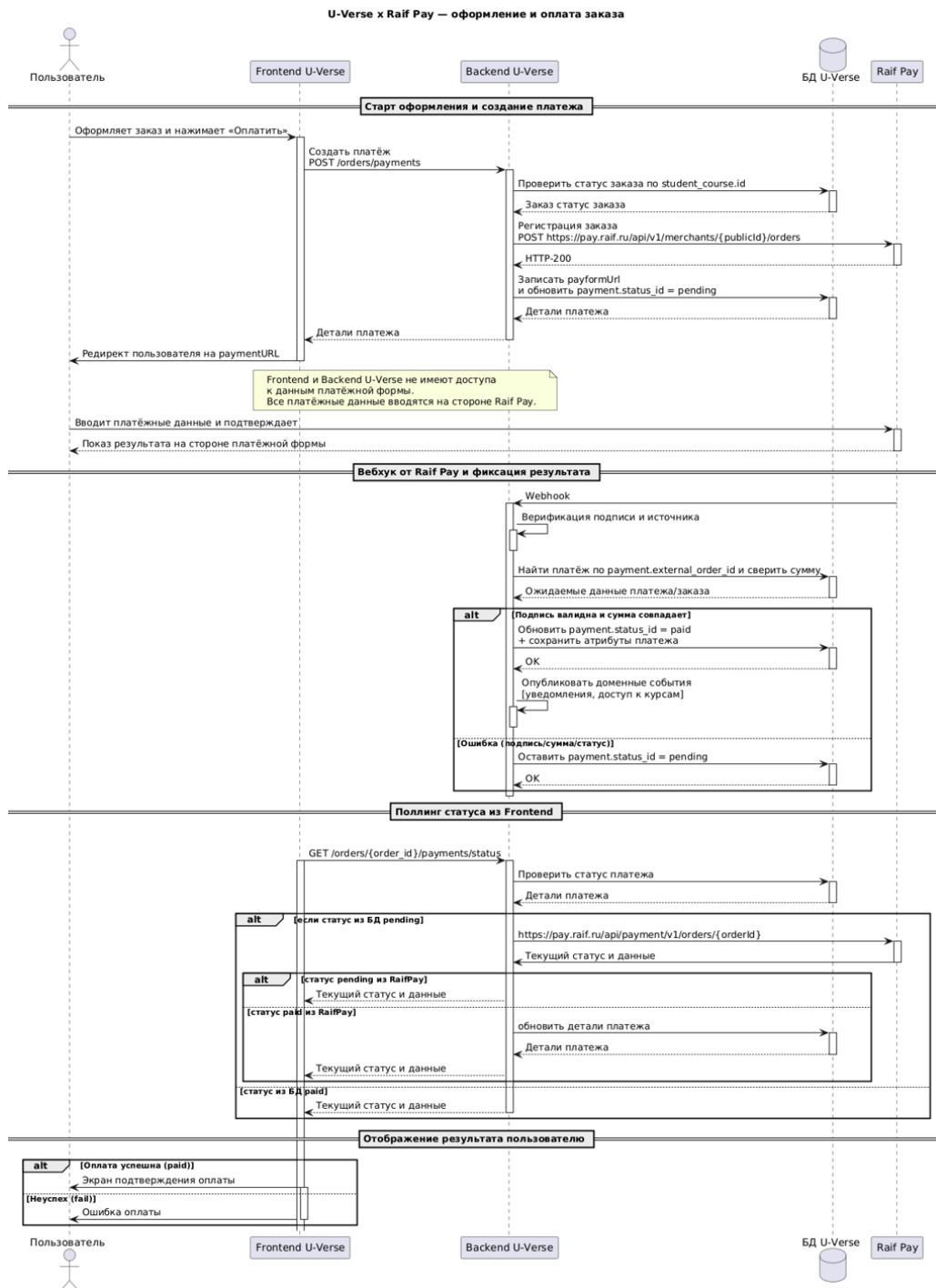
Статус платежа не меняется, фиксируем инцидент с приоритетом "предупреждение".

10Б. Отказ банка.

Пользователь прошёл форму, банк вернул финальный неуспех. Вернуть пользователя на экран покупки курса и показать текст ошибки:

1 Платеж не завершен, если у вас есть вопросы, обратитесь в саппорт help@U-Verse.org

UML-Sequence процесса оплаты



Код PlantUML

```

1 @startuml
2 title U-Verse x Raif Pay — оформление и оплата заказа
3
4 actor Пользователь
5 participant "Frontend U-Verse" as Frontend
6 participant "Backend U-Verse" as Backend
7 database "БД U-Verse" as DB
8 participant "Raif Pay" as RaifPay
9

```

```
10 == Старт оформления и создание платежа ==
11 Пользователь -> Frontend : Оформляет заказ и нажимает «Оплатить»
12 activate Frontend
13
14 Frontend -> Backend : Создать платёж \nPOST /orders/payments
15 activate Backend
16
17 Backend -> DB : Проверить статус заказа по student_course.id
18 activate DB
19 DB --> Backend : Заказ статус заказа
20 deactivate DB
21
22 Backend -> RaifPay : Регистрация заказа \nPOST https://pay.raif.ru/api/v1/merchants/{publ
23 activate RaifPay
24 RaifPay --> Backend : HTTP-200
25 deactivate RaifPay
26
27 Backend -> DB : Записать payformUrl\nи обновить payment.status_id = pending
28 activate DB
29 DB --> Backend : Детали платежа
30 deactivate DB
31
32 Backend --> Frontend : Детали платежа
33 deactivate Backend
34
35 Frontend -> Пользователь : Редирект пользователя на paymentURL
36 deactivate Frontend
37
38 note over Frontend,Backend
39 Frontend и Backend U-Verse не имеют доступа
40 к данным платёжной формы.
41 Все платёжные данные вводятся на стороне Raif Pay.
42 end note
43
44 Пользователь -> RaifPay : Вводит платёжные данные и подтверждает
45 activate RaifPay
46 RaifPay --> Пользователь : Показ результата на стороне платёжной формы
47 deactivate RaifPay
48
49 == Вебхук от Raif Pay и фиксация результата ==
50 RaifPay -> Backend : Webhook
51 activate Backend
52
53 Backend -> Backend : Верификация подписи и источника
54 activate Backend
55 deactivate Backend
56
57 Backend -> DB : Найти платёж по payment.external_order_id и сверить сумму
58 activate DB
59 DB --> Backend : Ожидаемые данные платежа/заказа
60 deactivate DB
61
62 alt Подпись валидна и сумма совпадает
63   Backend -> DB : Обновить payment.status_id = paid\n+ сохранить атрибуты платежа
64   activate DB
65   DB --> Backend : OK
66   deactivate DB
67
```



```
68 Backend -> Backend : Опубликовать доменные события\n[уведомления, доступ к курсам]
69 activate Backend
70 deactivate Backend
71 else Ошибка (подпись/сумма/статус)
72 Backend -> DB : Оставить payment.status_id = pending
73 activate DB
74 DB --> Backend : OK
75 deactivate DB
76 end
77 deactivate Backend
78 == Поллинг статуса из Frontend ==
79 loop Пока пользователь ожидает результат
80 Frontend -> Backend : GET /orders/{order_id}/payments/status
81 activate Backend
82 activate Frontend
83 Backend -> DB : Проверить статус платежа
84 activate DB
85 DB --> Backend : Детали платежа
86 deactivate DB
87 alt если статус из БД pending
88 Backend -> RaifPay : https://pay.raif.ru/api/payment/v1/orders/{orderId}
89 activate RaifPay
90 RaifPay -> Backend : Текущий статус и данные
91 deactivate RaifPay
92 alt статус pending из RaifPay
93 Backend --> Frontend : Текущий статус и данные
94
95 else статус paid из RaifPay
96 Backend -> DB : обновить детали платежа
97 activate DB
98 DB --> Backend : Детали платежа
99 deactivate DB
100 Backend --> Frontend : Текущий статус и данные
101 end
102 else статус из БД paid
103 Backend --> Frontend : Текущий статус и данные
104 deactivate Backend
105 end
106
107 == Отображение результата пользователю ==
108 alt Оплата успешна (paid)
109 Frontend -> Пользователь : Экран подтверждения оплаты
110 activate Frontend
111 else Неуспех (fail)
112 Frontend -> Пользователь : Ошибка оплаты
113 deactivate Frontend
114 end
115 @enduml
```

Интеграционные методы API Backend

Создание платежа в Raif Pay

Общее описание

Интеграционный API-метод предназначен для создания платежа на оплату курсов через платежную систему Raif Pay.

Доступы и ограничения:

- Для авторизованных пользователей приложения U-Verse
- Не более 5 запросов в минуту для одной пользовательской сессии.

Алгоритм работы метода:

Требования описаны в общем алгоритме работы интеграции, шаги основного сценария 2-7.

Запрос

Тип метода: POST

URL: /orders/payments

Headers:

- Authorization: Bearer Token (обязательный)
- Content-Type: application/json

Авторизация:

- Bearer Token

Query-параметры:

Отсутствуют

Тело запроса:

```
1 {  
2   "order": "d5d7b06a-6d68-4155-a8ab-395521a5bc0e"  
3 }
```

Успешный ответ

Код: HTTP-200

Тело ответа:

```
1 {
```

```
2  "order": "d5d7b06a-6d68-4155-a8ab-395521a5bc0e",
3  "created_at": "2025-01-10T20:10:00+03:00",
4  "course": {
5    "id": "d5d7b06a-6d68-4155-a8ab-395521a5bc0e",
6    "name": "d5d7b06a-6d68-4155-a8ab-395521a5bc0e"
7  },
8  "amount": 0.01,
9  "expiration_date": "2026-01-10T20:10:00+03:00",
10 "status": "pending",
11 "payform_url": "https://pay.raif.ru/pay?payformId=054f5023-85fb-4c0f-a892-cd7155657281"
12 }
```

Требования к обработке ошибок

Пример ответа - Ошибка валидации входных данных

Код HTTP: 400

Тело ответа (Body):

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

```
1 {
2   "code": "2002",
3   "message": "Значение поля {field} превышает допустимую длину"
4 }
```

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": "1003",
3   "message": "Аккаунт пользователя заблокирован"
4 }
```

Пример ответа - Превышен лимит запросов

Код HTTP: 429

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

Пример ответа - Ошибка запроса к Raif Pay (400/403/500) + Raif Pay недоступен (таймаут)

Код HTTP: 502

Тело ответа (Body):

```
1 {  
2   "code": "4002",  
3   "message": "Ошибка при обращении к платежному сервису",  
4 }  
5
```

Пример ответа - Ошибка базы данных

Код HTTP: 500

Тело ответа (Body):

```
1 {  
2   "code": "5002",  
3   "message": "Ошибка при работе с базой данных"  
4 }
```

Маппинг данных

Запрос:

Параметр	В API U-Verse	В БД U-Verse	В API Raif Pay в запросе на регистрацию платежа	Комментарий
	path-параметры:			

url для возврата в систему после оплаты - успех	—	—	returnUrls.successUrl	Всегда https://U-Verse urlsuccess
url для возврата в систему после оплаты - ошибка оплаты	—	—	returnUrls.failUrl	Всегда https://U-Verse urlfail
body(JSON):				
ID заказа	order	student_course.id	—	
Сумма к оплате	—	payment.sum	amount	Считается на бэке после скидки
Срок жизни заказа	—	payment.expires_at	expirationDate	
Локаль	—	user.country_id	locale	Страна указанная в профиле пользователя
ID платежа	—	payment.id	id	

Ответ:

Параметр	В API U-Verse	В БД U-Verse	В API Raif Pay	Комментарий
	body(JSON):			
ID заказа	order	student_course.id	–	Связанный заказ
Дата создания заказа	created_at	payment.created_at	–	Для отображения и аналитики
ID курса	course.id	student_course.course_id	–	Используется для связи с сущностью курс
Название курса	course.name	course.title	–	Для отображения на экране
Внешний ID заказа у банка	–	payment.external_order_id	id	
Сумма	amount	payment.sum	amount	Сравнивается запись в БД и ответ от Raif Pay
Дата истечения	expiration_date	payment.expires_at	expirationDate	Когда истечет срок на оплату
Статус платежа	status	payment.status_id	status.value	Актуальный статус на стороне U-Verse, при

				создании нового платежа на стороне Raif Pay всегда New
	—	—	status.date	Дата присвоения статуса в Raif Pay
Ссылка формы для редиректа	payform_Url	payment.paym ent_url	payformUrl	Ссылка переадреса ции на платёжную страницу Raif Pay

Получение статуса платежа в Raif Pay

Общее описание

Интеграционный API-метод предназначен для обновления статуса по платежу на покупку курса.

Доступы и ограничения:

- для авторизованных пользователей приложения U-Verse
- не более 5 запросов в минуту для одной пользовательской сессии.

Алгоритм работы метода:

Требования описаны в общем алгоритме работы интеграции, шаги основного сценария 10.

Здесь приводится расширенное описание работы метода по проверке статуса платежа методом GET /orders/payments/status с интеграцией к платежной системе Raif Pay.

1. Получен запрос от клиента API U-Verse.
2. Проверяется:
 - a. формат "order" - UUID
 - b. формат "payment.external_order_id" - UUID
 - c. соответствие "payment.external_order_id" для "order_id"

3. Проверить в БД статус платежа по полученному "payment.external_order_id" :
 - а. . Если платеж уже в статусе paid , значит был получен и обработан WebHook от Raif Pay об изменении статуса платежа.
 - б. Если платеж в статусе pending , то вызвать метод GET `https://pay.raif.ru/api/payment/v1/orders/{orderId}` для получения актуального статуса платежа в Raif Pay:
 - і. синхронизировать статус платежа с Raif Pay и обновить в БД, если был изменен
4. Вернуть актуальную информацию о платеже и заказе на клиента, который выполнил запрос.

Альтернативные сценарии и обработка ошибок:

См. ошибки на предыдущий метод.

Запрос

Тип метода: GET

URL: /orders/{order_id}/payments/status

Headers:

- Authorization: Bearer Token (обязательный)
- Content-Type: application/json

Авторизация:

- Bearer Token

Query-параметры:

- {order_id}

Тело запроса:

Отсутствует

Успешный ответ

Код: HTTP-201

Тело ответа:

```
1 {  
2   "order": "d5d7b06a-6d68-4155-a8ab-395521a5bc0e",  
3   "amount": 0.01,  
4   "status": "paid"  
5 }
```


Требования к обработке ошибок

Пример ответа - Ошибка валидации входных данных

Код HTTP: 400

Тело ответа (Body):

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

```
1 {  
2   "code": "2002",  
3   "message": "Значение поля {field} превышает допустимую длину"  
4 }
```

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": "1003",  
3   "message": "Аккаунт пользователя заблокирован"  
4 }
```

Пример ответа - Превышен лимит запросов

Код HTTP: 429

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

Пример ответа - Raif Pay недоступен (таймаут)

Код HTTP: 502

Тело ответа (Body):

```
1 {
2   "code": "4002",
3   "message": "Ошибка при обращении к платежному сервису",
4 }
```

Пример ответа - Ошибка базы данных

Код HTTP: 500

Тело ответа (Body):

```
1 {
2   "code": "5002",
3   "message": "Ошибка при работе с базой данных"
4 }
```

Маппинг данных

Запрос:

Параметр	В API U-Verse	В БД U-Verse	В API Raif Pay в запросе на получение статуса платежа	Комментарий
Идентификатор заказа	order_id	student_course.id	⊖	id заказа на стороне U-Verse
id платежа в Raif Pay	⊖	payment.id	{orderId}	id платежа по которому запрашивается статус

Ответ:

Параметр	В API U-Verse	В БД U-Verse	В API Raif Pay	Комментарий
ID заказа	order	student_course.id	—	Возвращаем наш UUID заказа.
Сумма	amount	payment.sum	amount	Сверяем с банком; при расхождении статус не обновляем.
Статус	status	payment.status_id	status.value	Сверяем с банком
	—	—	status.date	
	—	payment.expires_at	expirationDate	Когда истечет срок на оплату(не используется)
	—	—	qr.id	
	—	payment.external_order_id	id	Записываем в БД.

Обработка WebHook от Raif Pay



WebHook Raif Pay



Создано: GetAnalyst Student • Обновлено: 2 минуты назад

Автор: Даниэл Пянзин Telegram: @DanielPyanzin
Оглавление: Общее описание Что это:Обработчик входящих уведомлений от Raif Pay по событиям оплаты и возврата по Интернет-эквайрингу Для чего: Для реализации интеграция между приложением U-Verse и системой интернет-эквайринга Для кого: Для студентов U-Verse, желающих быстро и...

Confluence

Открыть предпросмотр

Маппинг данных UI - API нашей системы

1. Экран “Оплаты заказа”

Пользователь видит детали заказа и уходит на платёжную форму Raif Pay. Данные берём из ответа `GET /orders`


Ссылка на макет:  [U-Verse](#)

Параметр на UI	В ответе JSON Основной системы (U-Verse)	Описание и требования к ФЛК (форматно-логическому контролю)
Название курса	course.name	Обязателен
Номер заказа	order	Обязателен
Сумма к оплате	amount	Формат отображения: «₽»/ <code>RUB</code> <code>1 990,00 ₽</code>).
Валюта	—	Отображаем «₽»/ <code>RUB</code> константой.
Таймер до истечения	expiration_date	Если время истекло отобразить - “Срок ссылки истек”
Кнопка «Оплатить»	payform_url	Валидный <code>https</code> URL. Нажатие — редирект на Raif Pay. Блокировать двойной клик до навигации.
Ссылка на публичную оферту, политику обработки персональных данных и правилами	—	<ul style="list-style-type: none">• https://u-verse.org/legal/public-offer• https://u-verse.org/legal/privacy• https://u-verse.org/legal/terms

использования платформы		
Ссылка на часто задаваемые вопросы “Как оплатить курс?”	—	<ul style="list-style-type: none"> • https://u-verse.org/help/how-to-pay
Ссылка на часто задаваемые вопросы “Оплатить от компании”	—	<ul style="list-style-type: none"> • https://u-verse.org/help/company-payment

2. Экран “Ожидания оплаты”

После возвращения с платёжной формы UI опрашивает бэкенд и показывает прогресс.

Ссылка на макет:  U-Verse


Параметр на UI	В ответе JSON Основной системы (U-Verse)	Описание и требования к ФЛК (форматно-логическому контролю)
Номер заказа	order	Обязателен
Текущий статус оплаты	status	Допустимые значения: <code>pending</code> , <code>paid</code> , <code>faled</code> . Любое иное — считать ошибкой
Сообщение статуса	status	<code>pending</code> → «Платёж обрабатывается...», <code>paid</code> → «Оплата успешна», <code>faled</code> → «Оплата не завершена».

Кнопка «Обновить»		Доступна при <code>pending</code> . Повторно вызывает <code>GET /orders/payments/status</code>
----------------------	--	---

3. Экран “Результат оплаты”

Финальный экран после успешной/неуспешной оплаты.

Ссылка на макет: [U-Verse](#)

Параметр на UI	В ответе JSON Основной системы (U-Verse)	Описание и требования к ФЛК (форматно-логическому контролю)
Номер заказа	order	Обязателен
Заголовок и сообщение	status	<code>paid</code> → успешный экран; <code>FAILED</code> → экран ошибки; <code>pending</code> сюда не должен попадать (перехватывается на Экране 2).
Кнопка «Перейти к курсу»	Course.id (сохранено с Экран 1)	Линк на <code>/lk/courses</code>
Сумма оплаченная	amount	Формат как на Экране 1.
Кнопка «На главную»		Статическая навигация


Если оплата **FAILED**

Вернуть пользователя на экран 1 и показать текст ошибки:

1 Платеж не завершен, если у вас есть вопросы, обратитесь в саппорт help@U-Verse.org

4. Экран “Ошибка”

Вернуть пользователя на экран с заказами и показать текст ошибки:

Ссылка на макет:  U-Verse

Параметр на UI	В ответе JSON Основной системы (U- Verse)	Описание и требования к ФЛК (форматно-логическому контролю)
Текст ошибки	message	Печатать пользовательский текст из ответа: 400/2001–2002, 403/1003, 502/4002, 500/5002.

Логирование и мониторинг

Логирование (Kibana):

- Запросы от клиентов в API U-Verse:
 - тип метода
 - url
 - payload
 - код http ответа
 - тело ответа
- Запросы от клиентов в Backend U-Verse в API Raif Pay:
 - тип метода
 - url
 - payload
 - код http ответа
 - тело ответа

Мониторинг (Kibana):

Событие	Описание	Тип	Статус
Получена ошибка валидации в Raif Pay - HTTP-	При вызове любого из методов API Raif Pay получена	Ошибка	Критический

400 Bad Request.	ошибка HTTP-400 Bad Request.		
Получена ошибка валидации в Raif Pay - HTTP 403 Forbidden	При вызове любого из методов API Raif Pay получена ошибка HTTP 403 Forbidden	Ошибка	Критический
Шлюз к Raif Pay недоступен	При вызове любого из методов API Raif Pay нет ответа в течение 20 секунд - система Raif Pay недоступна	Ошибка	Критический
Получена иная ошибка от Raif Pay		Ошибка	Критический
Неверная подпись webhook	Статус платежа не меняется		предупреждение
Несовпадение сумм	Статус платежа не меняется		предупреждение

Автор: Даниэл Пянзин
Telegram: @DanielPyanzin