

Оплата мойки через Forte

Кошелёк приложения → банковская карта → шлюз Forte

01

Ссылка на оплату

Пополнение кошелька
на странице банка

02

Списание с карты

Отдельный запрос,
если кошелька не хватает

03

Статус и callback

Проверка оплаты
и уведомление мойки

- В приложении мойки есть внутренний кошелёк: с него оплачивают мойку и на него кладут деньги.
- Пополнение кошелька идёт через этот сервер — платёжный шлюз Forte.
- Если в кошельке не хватает средств — нужно списать недостачу с привязанной карты, отдельным запросом.
- Если на карте тоже пусто — долг живёт на мойке; когда деньги появятся, списание повторяется.

Требование	Как выглядит для клиента	Что делает система
Пополнить кошелёк	Открывает ссылку и платит на странице банка	CreateOrder → ссылка → callback мойке
Мойка при пустом кошельке	Не вводит карту повторно	execTran Recurring по сохранённому токenu
Нет денег на карте	Мойка / долг по правилам сервиса	Отказ банка → долг + ретрай на мойке
Карты клиентов	Видит маску карты в приложении	Токен Forte хранит сервер моек, не шлюз

Приложение мойки	Сервер моек	Этот сервер — шлюз	Forte Bank
● Экран мойки и кошелька ● Согласие на автосписание ● Показ маски карты	● Баланс кошелька ● Долги и ретраи charge ● Картотека: токен Forte, маска ● Решение: кошелёк или карта	● Создать заказ и ссылку ● exesTran без клиента ● Опрос getOrder (cron) ● Callback на мойку	● Платёжная страница, 3DS ● Списание с карты ● CoF — сохранённая карта ● Отказ, если нет денег

Шлюз не знает баланс кошелька. Мойка говорит: «дай ссылку» или «спиши N тенге с карты».

Есть в API Forte	Нет в API Forte
CreateOrder — создать заказ и ссылку на оплату	Проверить баланс карты заранее
CoF (Card on File) — сохранить карту при первой оплате	Узнать, что клиент пополнил банковскую карту
execTran Recurring — списать без клиента и без CVV	Поставить списание «в очередь до пополнения карты»
getOrder — статус заказа (FullyPaid, Declined...)	Входящий webhook от банка (только опрос)
HTTP Basic Auth, статусы ордера, коды ошибок	Холд «на будущее», если денег сейчас нет

Ключевой вывод

«Списать, когда на карту положат» — это не функция банка. Это ретраи charge на сервере моек плюс погашение долга после успешного пополнения кошелька.



Банк не ждёт пополнения карты. Ждёт мойка: повторяет charge или закрывает долг после FullyPaid по ссылке.

A

Ссылка · **POST /v1/wash/forte/link**

- Клиент в приложении, готов платить сам
- `save_card=true` → заказ `Order_COF` (привязка карты)
- `save_card=false` → обычный `Order_RID`
- Ответ: `link`, `order_id`, `payment_id`
- Клиент платит на странице банка
- `Cron getOrder` → `FullyPaid` → `callback + card{}`

B

Списание · **POST /v1/wash/forte/charge**

- Кошелька не хватает, карта уже привязана
- Мойка шлёт `client_rid` и сумму — не PAN
- `CreateOrder(Order_COF) + execTran Recurring`
- Успех → `callback success`, кошелёк / долг
- `PmoDecline` → `callback declined`, долг на мойке
- Ретраи долга запускает мойка, не шлюз

C

Статус · **getOrder + callback**

- У банка нет `webhook` — только опрос
- `Cron` раз в минуту по `pending`
- Или `GET /v1/wash/forte/status/{order_id}`
- Терминальный статус → один `callback`
- `notified_at` — без повторных уведомлений
- Мойка зачисляет баланс и закрывает долг

Сервер МОЕК

wallet	Баланс внутреннего кошелька
debts	Долг, если charge не прошёл
client_cards	src_token_id, маска, brand, consumer_id
user	Стабильный client_rid = consumer.rid
запрет	PAN, CVV, полный номер карты — не хранить

Этот сервер — шлюз Forte

payments	Заказ банка: order_id, password, сумма, статус, ссылка
client_payment_info	Хвост мойки 1:1: rid, callback, ошибки
card_payload	Снимок токена только чтобы отдать мойке
cron	getOrder по pending, один callback
запрет	Не картотека клиентов и не баланс кошелька

payments — как сейчас, без новых столбцов

Поле	Значение для мойки	Роль
module_type	wash	Отделить от Bitrix / amo / AIOKK
payment_type	wash_forte_link wash_forte_charge	Тип операции шлюза
order_id / password	из ответа Forte	Идентификатор заказа банка
amount / currency	сумма операции	Пополнение или charge
payment_status	pending / success / declined...	Наш статус
generated_link	ссылка HPP (y charge — null)	Открыть в приложении

client_payment_info — новая таблица, связь 1:1 (payment_id UNIQUE → payments.id)

Поле	Тип	Зачем
payment_id	FK unique	Один платёж — одна строка info
client_rid	varchar 64	ID клиента мойки = consumer.rid в Forte
wash_payment_id	varchar 64	ID операции на сервере моек
callback_url	varchar 500	Куда слать статус
operation_type	link charge	Какой сценарий
save_card / forte_status	bool / varchar	CoF и сырой статус банка
error_* / notified_at	varchar / ts	Отказ банка и защита от двойного callback
card_payload	json null	Снимок токена для мойки, не картотека

POST /api/v1/wash/forte/link и POST /api/v1/wash/forte/charge

Поле	link	charge	Смысл
amount, currency	да	да	Сумма пополнения или недостачи
description	да	да	Текст в заказе банка
client_rid	да	да	Стабильный id клиента, consumer.rid
wash_payment_id	да	да	Ид операции мойки
callback_url	да	да	URL для уведомления
hpp_redirect_url	да	нет	Куда вернуть клиента со страницы банка
save_card	да (bool)	нет	true = Order_COF, привязка карты
domain	да	да	Идентификатор инстанса мойки
данные карты	нет	нет	PAN не передаём; Forte ищет CoF по rid

Ответ link: { payment_id, order_id, link, status: pending } · Ответ charge: { payment_id, order_id, status: success | declined | pending }

JSON callback

```
{
  "wash_payment_id": "wp-456",
  "order_id": 1000000000475,
  "status": "success",
  "amount": "1000.00", "currency": "KZT",
  "client_rid": "user-123",
  "card": { "src_token_id": 1681, "mask": "444499*****7777", ... }
}
```

Поле	Что делает мойка
status=success	Зачислить кошелёк / закрыть долг
status=declined	Записать долг, запланировать ретрай
card{ }	Сохранить в client_cards (после CoF)
error_code	Показать причину / лимит ретраев
notified_at на шлюзе	Повторно не шлём, если 2xx

GET /api/v1/wash/forte/status/{order_id} — страховка, если callback не дошёл

- Cron шлюза каждую минуту опрашивает pending (wash_forte_link и wash_forte_charge).
- Блок card заполняется после первой CoF-оплаты — мойка кладёт токен в свою таблицу client_cards.
- Ретрай долга = новый POST /charge = новая пара payments + client_payment_info. Старую строку не переписываем.

1. Первое пополнение и привязка карты

- Клиент: «Пополнить 1000»
- Мойка → POST /link (save_card=true)
- Клиент платит на странице Forte
- Cron → FullyPaid → callback + card{}
- Мойка: кошелёк +1000, сохранить карту

2. Мойка, кошелька хватает

- Кошелёк 2000, мойка 800
- Мойка списывает 800 у себя
- Forte не вызывается
- Шлюз и банк не участвуют
- Самый частый и дешёвый путь

3. Не хватает кошелька, карта с деньгами

- Кошелёк 200, мойка 800, нестача 600
- Мойка → POST /charge 600
- Forte execTran OK
- Callback success → кошелёк +600
- Списать 800, открыть мойку

4. На карте нет денег

- charge → PmoDecline
- Мойка: долг 600 ₸
- Позже cron мойки снова /charge
- Или клиент сам /link
- После success долг гасится из кошелька

Шаг	HTTP	Тело (суть)	Когда
1. Ссылка	POST /order	typeRid: Order_COF Order_RID, consumer.rid, hppRedirectUrl	Клиент пополняет кошелёк
CoF-поля	в том же POST	hppCofCapturePurposes: Recurring, initiationEnvKind: Browser	save_card=true
2. Списание	POST /order	Order_COF, consumer.rid, initiationEnvKind: Server	Недостача / ретрай долга
2b. execTran	POST /order/{id}/exec-tran	phase: Single, conditions.cofUsage: Recurring	Сразу после создания заказа
3. Статус	GET /order/{id}	password, tranDetailLevel=2	Срон и ручная проверка
Auth	заголовок всех запросов	Authorization: Basic TerminalSys/login:password	Всегда

Статус Forte	Наш статус	Действие
Preparing	pending	Ждать, клиент ещё на странице банка
FullyPaid	success	Callback мойке, зачислить / закрыть долг
Declined	declined	Callback, долг и ретрай на мойке
Expired / Cancelled	expired / cancelled	Callback, ссылка протухла или отменена

Этап	Где	Работы
1	Шлюз	Миграция client_payment_info, модель, hasOne с payments
2	Шлюз	ForteWashPaymentController: link, charge, check, autoCheckWashOrders
3	Шлюз	Маршруты /v1/wash/forte/*, .env (FORTE_*), cron каждую минуту
4	Мойка	Таблицы кошелька, долгов, client_cards (токен, маска, rid)
5	Мойка	Вызовы link/charge, приём callback, зачисление баланса
6	Мойка	Логика: кошелёк → charge неостачи → долг → ретрай с лимитом
7	Совместно	Тест: ссылка, CoF, успешный charge, PmoDecline, повторный callback
8	Юр. / продукт	Оферта автосписания, галочка согласия в приложении

Параллельность

Этапы 1–3 (шлюз) и 4 (мойка) можно вести параллельно. Стыковка — этап 5. Коннекторы Bitrix/amo и AIOKK продолжают ходить в старый контроллер.

№	Правило
1	client_rid — стабильный id клиента мойки. Смена id = банк не найдёт сохранённую карту.
2	PAN и CVV не храним ни на шлюзе, ни на мойке. Только токен Forte и маска.
3	Баланс карты заранее не проверяем. Хватает / нет = ответ execTran.
4	Ретраи долга — на мойке, с паузой и лимитом попыток (антифрод банка).
5	Одна операция мойки (wash_payment_id) — одна попытка на шлюзе. Повтор = новый платёж.
6	Карта появляется только после успешного /link с save_card=true. Пока карты нет — только ссылка.
7	Предавторизация (Auth) не решает «подождать пополнения карты»: холд встанет, только если деньги уже есть.

Итоговая схема

