# CrossPay MVP Documentation

# Оглавление


1. Введение . . . 1
2. Архитектура системы . . . 1
   2.1. Обзор . . . 1
   2.2. Диаграмма взаимодействия . . . 2
3. API Reference . . . 3
   3.1. Инициация перевода . . . 3
4. Установка и настройка . . . 4
   4.1. Предварительные требования . . . 4
   4.2. Конфигурация . . . 4
5. Деплой . . . 5
   5.1. Локальная разработка . . . 5
6. Тестирование . . . 5
   6.1. Сценарии тестирования . . . 5
7. Часто задаваемые вопросы . . . 5
   7.1. Какие комиссии? . . . 5
   7.2. Сколько времени занимает перевод? . . . 6
8. Поддержка . . . 6
9. История изменений . . . 6
10. 0.1.0 (2024-01-01) . . . 6


# 1. Введение

CrossPay — это решение для международных переводов заработной платы из России в другие страны через Telegram Bot.

- **Статус:** MVP
- **Версия:** 0.1.0
- **Целевая аудитория:** Экспаты, получающие зарплату в RUB

# 2. Архитектура системы

## 2.1. Обзор

Основные компоненты системы:

- **Telegram Bot** — интерфейс взаимодействия с пользователем

- **Backend Server** — ядро системы (Python + Flask)
- **PostgreSQL** — база данных
- **ЮKassa API** — прием платежей в RUB
- **BaaS Provider** — выпуск карт и управление счетами

## 2.2. Диаграмма взаимодействия

```
Failed to generate image: PlantUML preprocessing failed: [From <input> (line 17) ]

@startuml
title CrossPay: Схема взаимодействия клиента, бота, бэкенда и банков

actor Клиент
participant "Telegram Mini App" as Бот
participant "Backend (Flask)" as Бэкенд
participant "ЮKassa"
participant "BaaS Provider (Европа)"
participant "SimplChain"
participant "Банк-партнёр (Дубай)"
participant "Европейский банк"

== Авторизация и открытие карты ==
Клиент -> Бот : Открывает Mini App
Бот -> Бэкенд : Отправляет initDataUnsafe
Бэкенд -> BaaS Provider : Запрос на выпуск карты
^^^^^
 Syntax Error? (Assumed diagram type: sequence)


@startuml
title CrossPay: Схема взаимодействия клиента, бота, бэкенда и банков

actor Клиент
participant "Telegram Mini App" as Бот
participant "Backend (Flask)" as Бэкенд
participant "ЮKassa"
participant "BaaS Provider (Европа)"
participant "SimplChain"
participant "Банк-партнёр (Дубай)"
participant "Европейский банк"

== Авторизация и открытие карты ==
Клиент -> Бот : Открывает Mini App
Бот -> Бэкенд : Отправляет initDataUnsafe
Бэкенд -> BaaS Provider : Запрос на выпуск карты
BaaS Provider -> Бэкенд : Карта выпущена
Бэкенд -> Бот : Подтверждение выпуска карты
Бот -> Клиент : Карта готова
```

```
== Указание суммы и выбор карты ==
Клиент -> Бот : Вводит сумму и выбирает карту
Бот -> Бэкенд : Запрос на конвертацию RUB → EUR
Бэкенд -> Бот : Возвращает сумму в EUR

== Покупка токенов через ЮKassa ==
Клиент -> Бот : Подтверждает оплату
Бот -> Бэкенд : Запрос на создание платежа
Бэкенд -> ЮKassa : Создание платежа (RUB)
ЮKassa -> Клиент : Редирект на форму оплаты
ЮKassa -> Бэкенд : Уведомление об успешной оплате

== Выпуск токенов и передача клиенту ==
Бэкенд -> SimplChain : Выпуск simpl_coin и transport_coin
SimplChain -> Бэкенд : Токены выпущены
Бэкенд -> Бот : Подтверждение
Бот -> Клиент : Токены зачислены на кошелёк

== Подтверждение перевода за границу ==
Клиент -> Бот : Подтверждает перевод simpl_coin
Бот -> Бэкенд : Запрос на трансграничную транзакцию
Бэкенд -> SimplChain : Списание simpl_coin и transport_coin
SimplChain -> Бэкенд : Токены списаны

== Перевод средств через банк-партнёр ==
Бэкенд -> Банк-партнёр (Дубай) : Запрос на перевод EUR
Банк-партнёр -> Европейский банк : Зачисление на карту клиента
Европейский банк -> Банк-партнёр : Подтверждение зачисления
Банк-партнёр -> Бэкенд : Средства зачислены

== Финализация и сжигание токенов ==
Бэкенд -> SimplChain : Сжигание simpl_coin
SimplChain -> Бэкенд : Токены уничтожены
Бэкенд -> Бот : Завершение операции
Бот -> Клиент : Перевод завершён

@enduml
```

# 3. API Reference

## 3.1. Инициация перевода

```
POST /transfer HTTP/1.1
Content-Type: application/json

{
  "user_id": 123456789,
```

```
  "amount_rub": 50000,
  "currency": "EUR"
}
```

**Response:**

```
{
  "transaction_id": "txn_001",
  "amount_eur": 476.19,
  "payment_details": {
    "account": "40702810000000000001",
    "comment": "txn_001"
  },
  "exchange_rate": 105.0
}
```

# 4. Установка и настройка

## 4.1. Предварительные требования

- Python 3.9+
- PostgreSQL 12+
- Telegram Bot Token
- ЮKassa API keys
- BaaS Provider account

## 4.2. Конфигурация

*Create* `.env` *file:*

```
# Database
DATABASE_URL=postgresql://user:pass@localhost/crosspay

# Telegram
TELEGRAM_BOT_TOKEN=your_bot_token

# Payment Providers
YOO_MONEY_SHOP_ID=your_shop_id
YOO_MONEY_SECRET_KEY=your_secret_key

# BaaS
BAAS_API_KEY=your_baas_key
BAAS_BASE_URL=https://api.baas-provider.com

# Exchange Rates
```

```
DEFAULT_EXCHANGE_RATE=105.0
COMMISSION_RATE=0.02
```

# 5. Деплой

## 5.1. Локальная разработка

```
# Клонирование репозитория
git clone https://github.com/yourname/crosspay.git
cd crosspay

# Установка зависимостей
pip install -r requirements.txt

# Настройка базы данных
python scripts/init_db.py

# Запуск приложения
python app.py
```

# 6. Тестирование

## 6.1. Сценарии тестирования

*Успешный перевод:*

- Пользователь отправляет /transfer 50000
- Система генерирует transaction_id
- Пользователь оплачивает через ЮKassy
- Админ подтверждает платеж
- Система создает карту через BaaS API
- Пользователь получает данные карты

# 7. Часто задаваемые вопросы

## 7.1. Какие комиссии?

- Конвертация RUB/EUR: 2%
- Выпуск виртуальной карты: бесплатно
- Выпуск физической карты: 5 EUR

### 7.2. Сколько времени занимает перевод?

- Обработка платежа: 1-24 часа
- Выпуск карты: мгновенно

## 8. Поддержка

- **Telegram канал:** https://t.me/crosspay_support
- **Issues:** https://github.com/yourname/crosspay/issues
- **Email:** support@crosspay.com

## 9. История изменений

## 10. 0.1.0 (2024-01-01)

- Первый рабочий MVP
- Базовая интеграция с ЮKassa и A2PAY
- Telegram Bot интерфейс