



СЕРВИС ДЛЯ УПРАВЛЕНИЯ ЛИМИТАМИ ПЛАТЕЖЕЙ

Инструкция по развертыванию экземпляра ПО



2023 ГОД

ООО ТЕСТ

115088, г.Москва, ул.Симоновский Вал, 26А

ОГЛАВЛЕНИЕ

1. Введение	2
1.1 Описание документа	2
1.2 Область применения	2
1.3 Термины и сокращения	2
1.4 Краткое описание возможностей	2
1.5 Уровень подготовки пользователей.....	3
2. Требования к эксплуатации серверной части.....	3
3. Установка.....	3
3.1 Описание компонентов	3
3.2 Порядок установки	3

1. ВВЕДЕНИЕ

Наименование программного обеспечения “Сервис для управления лимитами платежей”.

1.1 ОПИСАНИЕ ДОКУМЕНТА

Документ содержит сведения о порядке установки программного обеспечения “Сервис для управления лимитами платежей”. Также в документе приводятся требования к программному и аппаратному обеспечению.

1.2 ОБЛАСТЬ ПРИМЕНЕНИЯ

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) “Сервис для управления лимитами платежей”.

“Сервис для управления лимитами платежей” должен представлять собой техническое решение, самостоятельный программный продукт, который интегрируется с любыми IT-решениями. Продукт предназначен для автоматизации деятельности по контролю лимитов, позволяет получать и проверять установленные лимиты платежей по валюте и типу операции (пополнение, выплата).

1.3 ТЕРМИНЫ И СОКРАЩЕНИЯ

ПО	Программное обеспечение – совокупность компьютерных программ и связанных с ними данных, которая содержит инструкции по указанию компьютеру, что и как делать.
Сервис	Независимо компилируемый программный модуль, динамически подключаемый к основной программе и предназначенный для расширения и/или использования её возможностей. Сервисы обычно выполняются в виде библиотек общего пользования.
Платежный лимит	Установленное ограничение на максимальную сумму денежных операций, которые можно выполнить за определенный период времени.
Пополнение	Процесс внесения дополнительных средств на электронный счет пользователя.
Выплата	Операция выплаты средств на счет пользователя по его реквизитам.
API	Программный интерфейс приложения — это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными.
Очередь	Совокупность объектов, которые поддерживаются в последовательности и могут быть изменены путем добавления объектов на одном конце последовательности и удаления объектов с другого конца последовательности. Операции очереди делают ее структурой данных, которая обеспечивает хранение и передачу двоичных данных между различными участниками системы.
Инстанс	Дубликат объекта, сохраняющий неразрывную связь с оригиналом и полную зависимость от него. Модификация любого образца равносильна модификации оригинала – результаты сказываются как на оригинале объекта, так и на всех образцах.
Мерчант	Партнер, пользователь программного обеспечения.

1.4 КРАТКОЕ ОПИСАНИЕ ВОЗМОЖНОСТЕЙ

- Масштабирование благодаря наличию ресурсов для запуска нескольких экземпляров сервиса;
- Штатный функционал позволяет подписать **N инстансов** на одну и ту же очередь, сообщения из которой будут случайным образом приходит в тот или иной инстанс;
- Передача данных в виде запросов в Базу данных и получение этих данных;
- Проверка лимита платежа по валюте (можно ли производить оплату) и типу операции (пополнение, выплата);

- Получение лимита платежа по валюте и направлению (пополнение, выплата).

1.5 УРОВЕНЬ ПОДГОТОВКИ ПОЛЬЗОВАТЕЛЕЙ

Документ предназначен для администратора АС, выполняющего установку программного обеспечения “Сервис для управления лимитами платежей”.

2. ТРЕБОВАНИЯ К ЭКСПЛУАТАЦИИ СЕРВЕРНОЙ ЧАСТИ

Обеспечение функционирования ПО серверной части “Сервис для управления лимитами платежей” должно быть реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей должны являться:

- Современная ОС: Linux;
- Оперативная память: 32 Гб;
- Свободное дисковое пространство: не менее 50 Gb;
- Количество логических ядер процессора: 4;
- Частота процессора: 3.50 GHz.

Необходимо реализовать возможность разворачивания экземпляра ПО и на других ОС, поддерживающих платформу для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации Docker, например Windows 10 (Профессиональная или Корпоративная).

3. УСТАНОВКА

3.1 ОПИСАНИЕ КОМПОНЕНТОВ

Состав компонентов для установки:

- `gateway-api` – сервис для передачи информации от мерчантов и первичной обработки информации.
- `rabbitmq-service` – программный брокер сообщений на основе стандарта AMQP;
- `db-pay-service` – свободная объектно-реляционная система управления базами данных.
- `limits-service` – сервис для управления лимитами платежей.

3.2 ПОРЯДОК УСТАНОВКИ

Для разворачивания экземпляра ПО необходим `docker-compose` (инструментальное средство, входящее в состав Docker. Оно предназначено для решения задач, связанных с развёртыванием проектов). Его можно установить, следуя инструкции на официальном сайте <https://docs.docker.com/compose/install/>.

Необходимо создать файл `docker-compose.yml` с содержимым:

```
version: '2.4'
```

```
services:
```

```
limits-service:
```

```
image: 'docker.testllc.ru/paymentboom_limits-service:1.2.0.25'
```

```
restart: always
```

```
environment:
```

- DB_READ_DBNAME=db_pay
- DB_READ_HOST=db-pay-service
- DB_READ_DB_PORT=5432
- DB_WRITE_DBNAME=db_pay
- DB_WRITE_HOST=db-pay-service
- DB_WRITE_DB_PORT=5432
- DB_CONN_TIMEOUT_MILLIS=1000
- DB_IDLE_TIMEOUT_MILLIS=10000
- DB_MAX_CLIENTS=10
- DB_LOGGING=false
- DB_CACHE=false
- DB_MAX_QUERY_EXECUTION_TIME_MILLIS=5000
- DB_AUTO_RE_INIT=true
- DB_AUTO_RE_INIT_COUNT=10000
- DB_AUTO_RE_INIT_TIMEOUT_MILLIS=1000
- SERVER_CONNECTION_PROTOCOL=amqp
- SERVER_CONNECTION_HOSTNAME=rabbitmq-service
- SERVER_CONNECTION_PORT=5672
- SERVER_CONNECTION_VHOST=/
- SERVER_CONNECTION_FRAME_MAX=0
- SERVER_CONNECTION_CHANNEL_MAX=0
- SERVER_CONNECTION_HEARTBEAT=0
- SERVER_CONNECTION_LOCALE=en_US

- SERVER_AUTO_RECONNECT=true
- SERVER_AUTO_RECONNECT_TIMEOUT=5000
- SERVER_SHARED_SERVICE_QUEUE=true
- SERVER_SHARED_SERVICE_QUEUE_ACK=false
- SERVER_SERVICE_NAME=ph_limits-service
- LOG_LEVEL=0
- DB_READ_PASSWORD=STRONG_password
- DB_READ_USERNAME=postgres
- DB_WRITE_PASSWORD=STRONG_password
- DB_WRITE_USERNAME=postgres
- SERVER_CONNECTION_PASSWORD=rmpassword
- SERVER_CONNECTION_USERNAME=ruser
- K8S_POD_NAME=payment-status-pod-name
- K8S_POD_NAME=limits-service-pod-name

ports:

- "3000:3000"

depends_on:

- rabbitmq-service
- db-pay-service

gateway-api-service:

image: 'docker.testllc.ru/paymentboom_gateway-api:feat-import.195'

restart: always

environment:

- NGINX_WORKERS=2

- API_NAME=gateway-api
- API_VERSION=v1
- API_PORT=8080
- TRANSPORT_INIT_TIMEOUT=5000
- TRANSPORT_CONNECTION_HOSTNAME=rabbitmq-service
- TRANSPORT_CONNECTION_PORT=5672
- TRANSPORT_CONNECTION_VHOST=/
- TRANSPORT_AUTO_RECONNECT=true
- TRANSPORT_AUTO_RECONNECT_TIMEOUT=5000
- TRANSPORT_SERVICE_NAME=ruser
- TRANSPORT_PUBLISH_AWAIT_TIMEOUT=15000
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_STATUS=ph_payment-status
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_NOTIFICATION=ph_payment-notification
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENT_REGISTER=ph_payment-registration
- SETTINGS_TRANSPORT_SERVICES_NAMES_SECOND_CPS=second-cps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ECOMMPAY=ecommpay_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTROPAY_DIRECT=directa24_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_FIRST_CPS=first-cps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_QIWI_TOP_UP_CARD=qiwi-top-up-card_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTROPAY_CARD=astropay-card_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_CONTACT=contact_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ECOPAYZ=ecopayz_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_WIRECARD=wirecard_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_HALYK_BANK=halyk-bank_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_RBK_MONEY=rbk_money_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYEER=payeer_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYSAGE=paysage_plugin

- SETTINGS_TRANSPORT_SERVICES_NAMES_DAOWALLET=daowallet_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_SETTLE_PAY=settle_pay_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_ASTOPAY_ONE_TOUCH=astopay_onetouch_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_COINSPAID=coinspaid_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_REFERENCE_API=ph_reference-service
- SETTINGS_TRANSPORT_SERVICES_NAMES_APPLICATION_BALANCES_SERVICE=ph_application-balances-service
- SETTINGS_TRANSPORT_SERVICES_NAMES_JETON=jeton_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_LEOGAMING=leogaming_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_APS=aps_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_MUCHBETTER=muchbetter_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_EZEEWALLET=ezeewallet_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_BTQFINANCE=btqfinance_plugin

- SETTINGS_TRANSPORT_SERVICES_NAMES_CONNECTUM=connectum_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_INTERKASSA=interkassa_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_BCPO=bcpo_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYDEX=paydex_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_GETAPAY_PSP=getapay_psp_plugin
- SETTINGS_TRANSPORT_SERVICES_NAMES_JETONCASH=jetoncash_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_INWIZO=inwizo_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYCOS=paycos_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMEGA=paymega_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_WALLET_COM=wallet_com_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_MERCURY0=mercuryo_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_FINANA=finana_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_DECARDS2P=decards2p_integration_service

- SETTINGS_TRANSPORT_SERVICES_NAMES_P2PAY=p2pay_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYDO=paydo_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_ZIPPY=zippy_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_OXPROCESSING=oxprocessing_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_CONNPAY=connpay_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAY_PORT=pay_port_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_GATE_EXPRESS=gate_express_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYMENTS_ROUTER=payments_router
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYES=payes_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAY4FUN=pay4fun_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_PAYRETAILERS=payretailers_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_CYPPIX=cypix_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_JUMPFINANCE=jumpfinance_integration_service
- SETTINGS_TRANSPORT_SERVICES_NAMES_MAGIC_PAYMENTS=magic_payments_integration_service
- DB_MASTER_NAME=db_pay
- DB_MASTER_HOST=db-pay-service
- DB_MASTER_PORT=5432
- DB_MASTER_CONN_TIMEOUT_MILLIS=3000
- DB_MASTER_IDLE_TIMEOUT_MILLIS=10000
- DB_MASTER_MAX_CLIENTS=10
- DB_MASTER_PING_TIMEOUT_MILLIS=60000
- DB_SLAVE_NAME=db_pay
- DB_SLAVE_HOST=db-pay-service
- DB_SLAVE_PORT=5432
- DB_SLAVE_CONN_TIMEOUT_MILLIS=3000
- DB_SLAVE_IDLE_TIMEOUT_MILLIS=10000
- DB_SLAVE_MAX_CLIENTS=10

- DB_SLAVE_PING_TIMEOUT_MILLIS=60000
- LOG_LEVEL=0
- CARD_ENCRYPTION_KEY=lolkek
- DATABASE_MASTER_PASSWORD=STRONG_password
- DATABASE_MASTER_USER=postgres
- DATABASE_SLAVE_PASSWORD=STRONG_password
- DATABASE_SLAVE_USER=postgres
- TRANSPORT_CONNECTION_PASSWORD=rmpassword
- TRANSPORT_CONNECTION_USERNAME=rmuser
- K8S_POD_NAME=gateway-api-lol-kek
- DB_MASTER_USER=postgres
- DB_MASTER_PASSWORD=STRONG_password
- DB_SLAVE_USER=postgres
- DB_SLAVE_PASSWORD=STRONG_password

ports:

- "8080:8080"

depends_on:

- rabbitmq-service
- db-pay-service

rabbitmq-service:

image: rabbitmq:3.11.8-management

hostname: rabbitmq-service

restart: always

environment:

- RABBITMQ_DEFAULT_USER=rmsuser
- RABBITMQ_DEFAULT_PASS=rmpassword
- RABBITMQ_SERVER_ADDITIONAL_ERL_ARGS=-rabbit channel_max 0

volumes:

- ./rabbitmq:/var/lib/rabbitmq

command:

- bash

- -c

- |

```
cd /opt/rabbitmq/plugins/
```

```
apt update
```

```
apt install -y wget
```

```
wget https://github.com/rabbitmq/rabbitmq-delayed-message-exchange/releases/download/3.11.1/rabbitmq_delayed_message_exchange-3.11.1.ez
```

```
rabbitmq-plugins enable rabbitmq_delayed_message_exchange
```

```
rabbitmq-server start
```

ports:

- "15672:15672"

- "5672:5672"

db-pay-service:

container_name: db-pay-service

image: 'postgres:14'

environment:

POSTGRES_PASSWORD: STRONG_password

POSTGRES_USER: postgres

POSTGRES_DB: db_pay

restart: on-failure

volumes:

- ./postgres:/var/lib/postgresql/

ports:

- "5432:5432"

command: ["postgres", "-c", "wal_level=logical"]

volumes:

rabbitmq:

external: true

postgres:

external: true