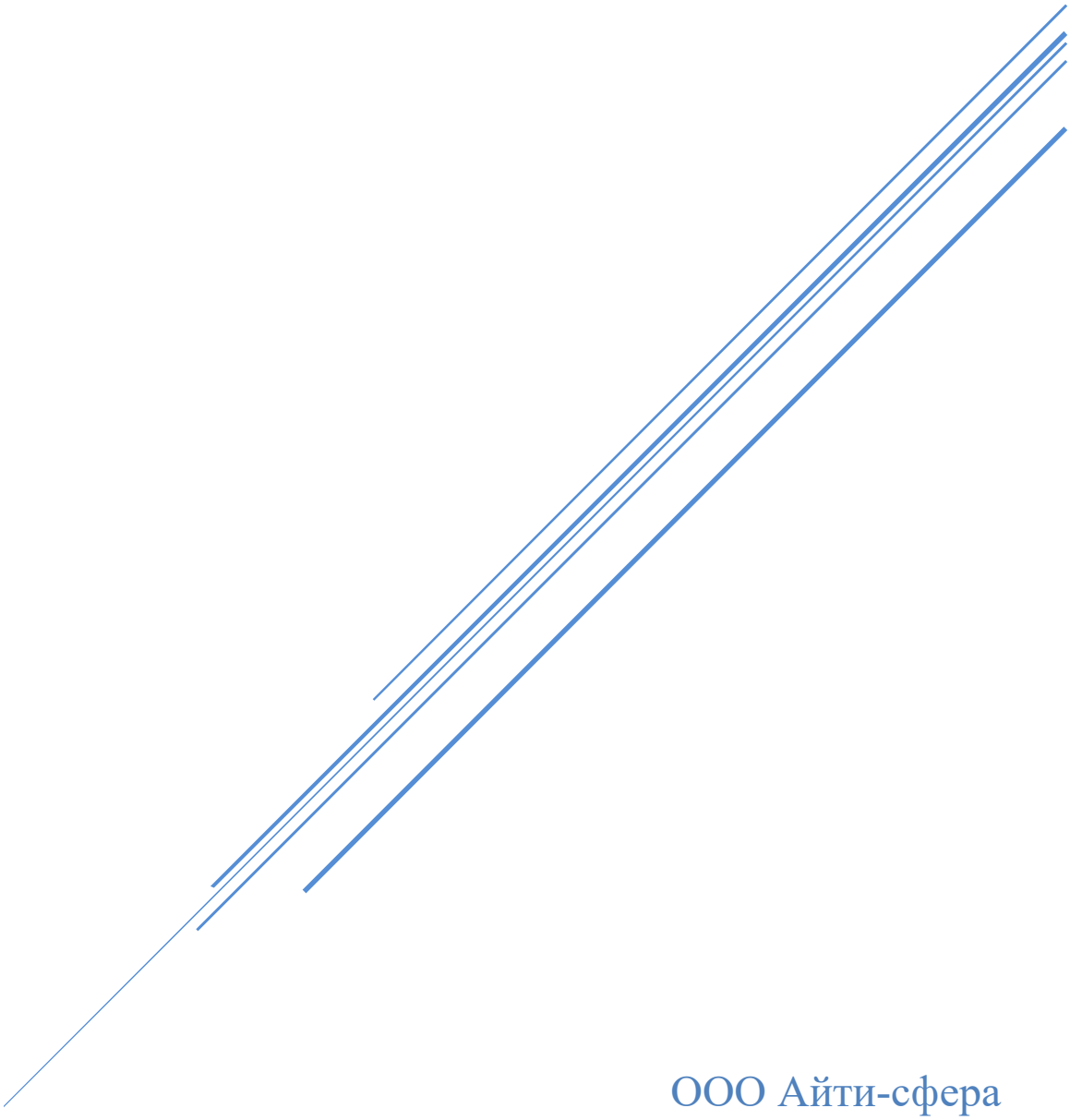
# ИНСТРУКЦИЯ ПО РАЗВЕРТЫВАНИЮ ЭКЗЕМПЛЯРА ПО

## Сервис дополнительного анализа и обработки операций



ООО Айти-сфера

Москва, 2023 г.

## Оглавление

# 1. Введение

## *1.1 Наименование ПО*

Наименование программного обеспечения "*Сервис дополнительного анализа и обработки операций*".

## *1.2 Назначение и характеристика*

Настоящий документ (далее – Описание) распространяется на программное обеспечение (далее – ПО) "*Сервис дополнительного анализа и обработки операций*". "*Сервис дополнительного анализа и обработки операций*" представляет собой техническое решение, которое позволяет после совершения платежа произвести дополнительную операцию. К примеру, после реализации платежа в **ЦУПИС** дополнительной операцией может быть отправка результата данного платежа в **ЕРАИ**.

## *1.3 Описание документа*

Документ содержит сведения о порядке установки программного обеспечения "*Сервис дополнительного анализа и обработки операций*". Также в документе приводятся требования к программному и аппаратному обеспечению.

## *1.4 Определения*

| | |
|---|---|
| **ПО** | Программное обеспечение – совокупность компьютерных программ и связанных с ними данных, которая содержит инструкции по указанию компьютеру, что и как делать. |
| **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 ПО, имеющий специальный счёт, который позволяет торговой компании принимать платежи с помощью банковских карт. |
| **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. |
| **ЦУПИС** | Аббревиатура ЦУПИС расшифровывается как Центр учёта переводов интерактивных ставок. Это финансовая организация-посредник, которая обеспечивает денежные переводы между букмекерской конторой и ее клиентами. И берёт за это процент комиссии. У каждого игрока в легальных спортивных ставках есть собственный счёт в ЦУПИС. |
| **ЕРАИ** | Публично-правовая компания «Единый регулятор азартных игр» создана 23 июня 2021 года Российской Федерацией на основании Федерального закона от 30.12.2020 № 493-ФЗ и распоряжения Правительства Российской Федерации от 09.06.2021 № 1526-р.<br>Компания создана в целях повышения эффективности государственного контроля (надзора) за организацией и проведением азартных игр и обеспечения внебюджетного финансирования спорта в Российской Федерации. |

## *1.5 Перечень программной документации*

- "*Сервис дополнительного анализа и обработки операций*". Техническое задание (ГОСТ 19.201-78);
- "*Сервис дополнительного анализа и обработки операций*". Руководство пользователя;
- "*Сервис дополнительного анализа и обработки операций*". Описание функциональных характеристик программного обеспечения;
- "*Сервис дополнительного анализа и обработки операций*". Описание жизненного цикла разработки ПО;

- "*Сервис дополнительного анализа и обработки операций*". Инструкция по развертыванию экземпляра ПО;
- "*Сервис дополнительного анализа и обработки операций*". Инструкция для проверки в тестовой среде ПО.

# 2. Требования

## *2.1 Требования к эксплуатации серверной части*

Обеспечение функционирования ПО серверной части "*Сервис дополнительного анализа и обработки операций*" реализовано на базе серверной операционной системы Linux. Минимальной конфигурацией аппаратной составляющей являются:

- ✓ Современная ОС: Linux;
- ✓ Оперативная память: 2 ГБ;
- ✓ Свободное дисковое пространство: не менее 20 Gb;
- ✓ Количество логических ядер процессора: 2;
- ✓ Частота процессора: 3.50 GHz.

Возможно разворачивание 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).

## *2.2 Уровень подготовки пользователей*

Для интеграции API пользователь должен иметь квалификацию разработчика не ниже уровня Regular Middle.

# 3. Установка

## 3.1 Описание компонентов

- **multihub-mock-service** – сервис создания тестового окружения для внешних мерчантов;
- **postprocessing-api** – сервис дополнительного анализа и обработки операций;
- **rabbitmq-service** – программный брокер сообщений на основе стандарта AMQP;
- **redis** – резидентная система управления базами данных класса NoSQL;

- **db-pay-service** – свободная объектно-реляционная система управления базами данных.

## 3.2 Порядок установки

Для развертывания экземпляра **ПО,** необходим docker-compose (инструментальное средство, входящее в состав **Docker**. Оно предназначено для решения задач, связанных с развертыванием 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:
 multihub-mock-service:
  image: 'docker.itisphera-test.ru/paymentboom_multihub-mock-service:feat-
postprocessing.96'
  restart: always
  environment:
   - AMQP_USER=rmuser
   - AMQP_PASSWORD=rmpassword
   - LOGGER_LEVEL=debug
   - TRANSPORT_CONNECTION_PROTOCOL=amqp
   - TRANSPORT_CONNECTION_HOSTNAME=rabbitmq-service
   - TRANSPORT_CONNECTION_PORT=5672
   - TRANSPORT_CONNECTION_VHOST=/
   - TRANSPORT_CONNECTION_USERNAME=rmuser
   - TRANSPORT_CONNECTION_PASSWORD=rmpassword
   - TRANSPORT_CONNECTION_FRAME_MAX=0
   - TRANSPORT_CONNECTION_CHANNEL_MAX=0
   - TRANSPORT_CONNECTION_HEARTBEAT=0
   - TRANSPORT_AUTO_RECONNECT=true
   - TRANSPORT_AUTO_RECONNECT_TIMEOUT=5000
   - TRANSPORT_SERVICE_NAME=multihub_mock_service
   - TRANSPORT_SHARED_SERVICE_QUEUE=true
   - TRANSPORT_SHARED_SERVICE_QUEUE_MESSAGE_TTL=10000
   - TRANSPORT_EXCLUSIVE_QUEUE_ACK=false
   - TRANSPORT_SHARED_SERVICE_QUEUE_ACK=false
   - TRANSPORT_MAIN_EXCHANGE_NAME=main
   - TRANSPORT_MAIN_REPLY_EXCHANGE_NAME=main_reply
   - TRANSPORT_PUBLISH_AWAIT_TIMEOUT=10000
   - TRANSPORT_PREFETCH=100
   - TRANSPORT_MAIN_SCHEDULE_EXCHANGE_NAME=main_schedule
   - TRANSPORT_MAIN_SCHEDULE_QUEUE_NAME=main_schedule
   - TRANSPORT_MAIN_GARBAGE_EXCHANGE_NAME=main_garbage
```

```
  - TRANSPORT_MAIN_GARBAGE_QUEUE_NAME=main_garbage
  - DATABASE_MASTER_DATABASE=db_pay
  - DATABASE_MASTER_HOST=db-pay-service
  - DATABASE_MASTER_PORT=5432
  - DATABASE_MASTER_USER=postgres
  - DATABASE_MASTER_PASSWORD=STRONG_password
  - DATABASE_SLAVE_DATABASE=db_pay
  - DATABASE_SLAVE_HOST=db-pay-service
  - DATABASE_SLAVE_PORT=5432
  - DATABASE_SLAVE_USER=postgres
  - DATABASE_SLAVE_PASSWORD=STRONG_password
  - DATABASE_POOL_CONNECTION_TIMEOUT_MILLIS=2000
  - DATABASE_POOL_IDLE_TIMEOUT_MILLIS=30000
  - DATABASE_POOL_MAX=10
  - DATABASE_LOGGING=false
  - SERVER_HOST=0.0.0.0
  - SERVER_PORT=4000
  - SERVICE_NAMES_REFERENCE_API=ph_reference-service
  - SERVICE_NAMES_PAYMENT_STATUS=ph_payment-status
  - SERVICE_NAMES_GATEWAY_CALLBACK_WORKER=ph_callback-worker
  - SERVICE_NAMES_PAYMENT_NOTIFICATION=ph_payment-notification
  - MULTIHUB_MOCK_SERVICE_URL=http://5.182.4.89:4000/v1
  - MULTIHUB_DEFAULT_SCHEMA_ENABLED=true
  - KEY_VALUE_STORAGE_REDIS_HOST=redis
  - KEY_VALUE_STORAGE_REDIS_PORT=6379
  - KEY_VALUE_STORAGE_REDIS_TIMEOUT=5000
  - KEY_VALUE_STORAGE_REDIS_CONNECTION_TIMEOUT=5000
  - MONITORING_SERVICE_NAME=multihub-mock-service
  - PROMETHEUS_HTTP_PORT=9696
  - K8S_POD_NAME=paymentboom_multihub-mock-service
 ports:
  - "4000:4000"
 depends_on:
  - rabbitmq-service
  - db-pay-service
  - redis
postprocessing-api-service:
 image: 'docker.itisphera-test.ru/paymentboom_postprocessing-api:feat-import.60'
 restart: always
 environment:
  - NGINX_WORKERS=2
  - TRANSPORT_CONNECTION_PROTOCOL=amqp
  - TRANSPORT_CONNECTION_HOSTNAME=rabbitmq-service
  - TRANSPORT_CONNECTION_PORT=5672
  - TRANSPORT_CONNECTION_VHOST=/
  - TRANSPORT_CONNECTION_FRAME_MAX=0
  - TRANSPORT_CONNECTION_CHANNEL_MAX=0
  - TRANSPORT_CONNECTION_HEARTBEAT=0
```

```
  - TRANSPORT_AUTO_RECONNECT=true
  - TRANSPORT_AUTO_RECONNECT_TIMEOUT=5000
  - TRANSPORT_SERVICE_NAME=ph_postprocessing
  - TRANSPORT_SHARED_SERVICE_QUEUE=true
  - TRANSPORT_SHARED_SERVICE_QUEUE_MESSAGE_TTL=10000
  - TRANSPORT_EXCLUSIVE_QUEUE_ACK=false
  - TRANSPORT_SHARED_SERVICE_QUEUE_ACK=true
  - TRANSPORT_MAIN_EXCHANGE_NAME=main
  - TRANSPORT_MAIN_REPLY_EXCHANGE_NAME=main_reply
  - TRANSPORT_PUBLISH_AWAIT_TIMEOUT=10000
  - TRANSPORT_PREFETCH=5
  - TRANSPORT_MAIN_SCHEDULE_EXCHANGE_NAME=main_schedule
  - TRANSPORT_MAIN_SCHEDULE_QUEUE_NAME=main_schedule
  - TRANSPORT_MAIN_GARBAGE_EXCHANGE_NAME=main_garbage
  - TRANSPORT_MAIN_GARBAGE_QUEUE_NAME=main_garbage
  - DATABASE_MASTER_DATABASE=db_postprocessing
  - DATABASE_MASTER_HOST=db-pay-service
  - DATABASE_MASTER_PORT=5432
  - DATABASE_SLAVE_DATABASE=db_postprocessing
  - DATABASE_SLAVE_HOST=db-pay-service
  - DATABASE_SLAVE_PORT=5432
  - DATABASE_POOL_CONNECTION_TIMEOUT_MILLIS=2000
  - DATABASE_POOL_IDLE_TIMEOUT_MILLIS=30000
  - DATABASE_POOL_MAX=10
  - DATABASE_LOGGING=false
  - LOGGER_LEVEL=debug
  - ERAI_BASE_URI=https://test.test
  - ERAI_OPERATOR_ID=1
  - ERAI_API_KEY=test
  - ERAI_CERT=""
  - ERAI_CERT_PASSPHRASE=""
  - ERAI_ORIGINAL_HOST=webhook.site
  - ERAI_RETRY_TRANSACTION_NOT_FOUND_REPORT_TIMEOUT=120
  - SCHEDULER_TIME=*/10 * * * * *
  - SCHEDULER_SEND_REPORTS_COUNT_IN_MOMENT=100
  - SCHEDULER_CHECK_REPORTS_COUNT_IN_MOMENT=4
  - PROMETHEUS_HTTP_PORT=9696
  - 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=postprocessing-api-pod-name
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=rmuser
    - 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" ]
  redis:
   image: 'redis:7.2-rc-bullseye'
   environment:
    - ALLOW_EMPTY_PASSWORD=yes
    - REDIS_AOF_ENABLED=no
    - vm.overcommit_memory=1
   volumes:
    - ./redis:/data
```

```
  ports:
   -  "6379:6379"
volumes:
  rabbitmq:
    external: true
  postgres:
    external: true
  redis:
    external: true
```