

Введение

Цель документации

Документация предназначена для описания архитектуры, установки, эксплуатации и поддержки SaaS решения с микросервисной архитектурой. Она служит справочным материалом для разработчиков, администраторов и пользователей системы.

Обзор системы

Система состоит из множества микросервисов, каждый из которых выполняет свою задачу и взаимодействует с другими сервисами через стандартизированные API. Основные компоненты системы включают аутентификацию, обработку данных, хранение данных и управление пользователями.

Языки программирования

Приложение написано с использованием языков: JavaScript, Python, PHP

Общая архитектура

Основные компоненты

- Аутентификационный сервис (Auth Service)
- Rest-API сервис (Api Service)
- ETL процессы (Airflow Service)
- Сервис хранения данных (DB Service)
- Сервис хранения файлов (Files Service)
- Сервис отчетов (Reports Service)
- Сервис телеграмм бота (Telegrambot Service)
- Веб-интерфейс админ панели (Admin Service)
- Веб-интерфейс Macrobank (Erp Service)
- Веб-интерфейс Macrobank KSB (KSB Service)

Auth Service

Назначение: Сервис отвечает за аутентификацию, хранение пользователей и выдачу токенов доступа.

База данных: PostgreSQL

Язык программирования: Python

Api Service

Назначение: Обработка и анализ данных, поступающих от пользователей и внешних источников.

База данных: MongoDB

Язык программирования: Node JS

Files Service

Назначение: управление хранением и доступом к файлам.

Язык программирования: Node JS

Airflow Service

Назначение: Платформа для создания, планирования и мониторинга рабочих процессов.

База данных: PostgreSQL

Язык программирования: Python

DB Service

Назначение: Управление и обслуживание баз данных системы.

Reports Service

Назначение: Сервис отвечает за хранение шаблонов и генерации XLS таблиц

База данных: PostgreSQL

Язык программирования: Python

Telegrambot Service

Назначение: Содержит конфигурацию телеграмм бота и методы обращения к нему.

Admin Service

Назначение: Веб-интерфейс панели администрации (краткое описание)

Фреймворк: Vue.js с использованием Nuxt.js, построенный на архитектуре SPA (Single Page Application)

ERP Service

Назначение: Веб-интерфейс главного приложения (краткое описание)

Фреймворк: Vue.js с использованием Nuxt.js, построенный на архитектуре SPA (Single Page Application)

Системные требования

Сервер (хостинг)

- Unix подобная система
- Docker

Клиент

- Браузер на базе Chromium последней версии

Установка и деплоймент

- Клонировать репозиторий
- Настроить переменные окружения
- Собрать docker образы
- Развернуть docker сервисы

Мониторинг и логирование

Инструменты мониторинга

- Grafana

Логи и их хранение

- Логи хранятся в Elasticsearch. (Уточнить у Ярика)
- Kibana используется для визуализации логов.