



Метро в тандеме

▼ Введение

Обзор основных компонентов

Эта документация описывает развертывание системы, состоящей из нескольких контейнеров Docker, которые обеспечивают работу баз данных, ETL-процесса, бекенда и фронтенда. Все компоненты взаимодействуют друг с другом для предоставления полноценного решения.

Система состоит из следующих основных компонентов:

- **Фронтенд:** Веб-интерфейс для взаимодействия с пользователем
 - <https://метровтандеме.рф>
- **Бекенд:** Сервис, предоставляющий API и логику приложения.
 - <https://api.метровтандеме.рф>
- **База данных:** Используется PostgreSQL для хранения данных.
- **ETL-процесс:** Обработка данных и их перемещение из JSON в хранилище.

▼ Требования к АО и ПО

Список аппаратных и программных требований

Аппаратные требования

- Процессор: Минимум двухъядерный процессор
- Оперативная память: 4 ГБ и выше
- Место на диске: 20 ГБ и выше

- Поддержка виртуализации на уровне процессора

Программные требования

- Docker версии 19.03 и выше
- Docker Compose версии 1.25.0 и выше
- Поддержка виртуализации на уровне ОС

Инструкции по установке и настройке необходимого ПО

Установка Docker:

Для Windows и MacOS скачайте и установите Docker Desktop с официального сайта Docker.

Для Linux выполните следующие команды:

```
sudo apt-get update  
sudo apt-get install docker
```

Установка Docker Compose:

Выполните следующую команду для установки Docker Compose:

```
sudo apt-get install docker-compose
```

▼ Архитектура

Обзор архитектуры веб-сервиса, включая используемые технологии, компоненты и их взаимодействие

Backend (Kotlin с использованием Spring Boot)

Веб-сервис построен на основе Kotlin с использованием Spring Boot, что обеспечивает быструю разработку и управление приложениями.

Основные используемые технологии включают:

- **Spring Boot:** Фреймворк для создания автономных, готовых к продакшену приложений на основе Spring.
- **Spring Security:** Инфраструктура для обеспечения аутентификации и авторизации в приложениях.
- **Spring Data JPA:** Упрощает взаимодействие с базами данных через JPA (Java Persistence API) и Hibernate.
- **Spring Actuator:** Предоставляет возможности мониторинга и управления приложением в реальном времени.
- **Liquibase:** Читает .XML файлы с описанием миграции баз данных.

Frontend (Svelte с использованием Vite)

Фронтенд приложения разработан на базе Svelte с использованием Vite для быстрой сборки и разработки. Важные компоненты включают:

- **Svelte:** Прогрессивный фреймворк JavaScript для построения пользовательских интерфейсов.
- **Vite:** Быстрый инструмент сборки для современных веб-приложений с поддержкой быстрого развертывания и горячей перезагрузки.
- **Tailwind CSS:** Библиотека CSS, которая упрощает стилизацию компонентов и создание адаптивного дизайна.

Взаимодействие компонентов

- **Фронтенд и бекенд:** Фронтенд обращается к бекенду через API, предоставляемые Spring Boot. Это включает запросы на получение данных, отправку данных и аутентификацию пользователя (JWT).
- **База данных:** Бекенд взаимодействует с PostgreSQL для хранения и извлечения данных. Используется Spring Data JPA для выполнения операций CRUD (Create, Read, Update, Delete).

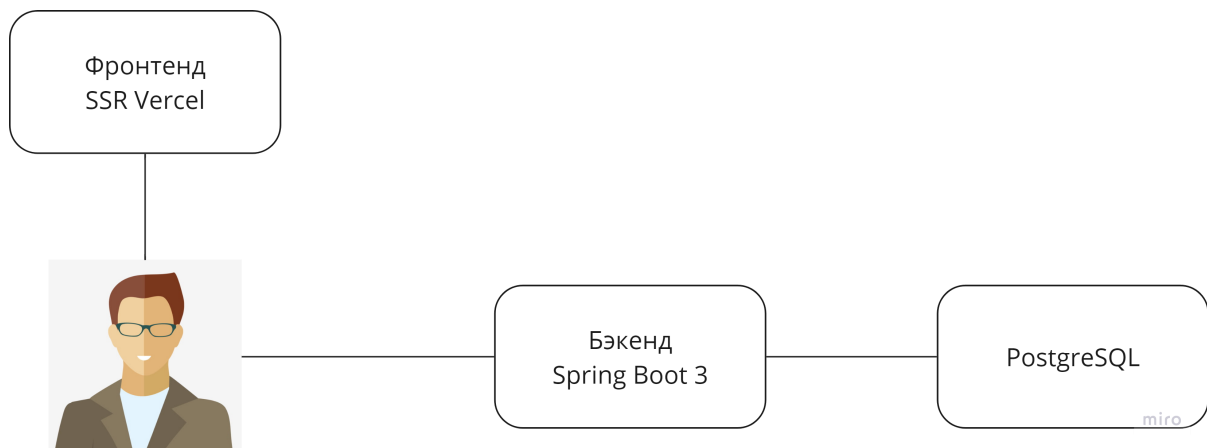
- **Безопасность и аутентификация:** Spring Security обеспечивает защиту API и управление доступом к различным ресурсам в зависимости от роли пользователя.

Развёртывание и сборка

- Проект поддерживается и собирается при помощи инструментов сборки Gradle и npm (Node Package Manager).

Диаграммы, поясняющая структуру и поток данных

Пользователь получает SSR страницу с сервиса Vercel, затем браузер пользователя с загруженной страницей обращается к API хоста бэкенда, в запросах с данными идет вызов запросов к базе данных.



▼ Инструкция по развертыванию

Клонирование репозитория:

Перейдите в директорию, где вы хотите разместить проект, и выполните команду:

```
git clone https://github.com/funcid/metro-in-tandem.git
cd metro-in-tandem
```

Запуск проекта:

```
docker-compose up --build
```

База данных (PostgreSQL): будет доступна на порту, указанном в переменной `POSTGRES_PORT` в файле `.env`. По умолчанию это порт 5432

Бэкенд: будет доступен на порту 8080

Фронтенд: будет доступен на порту 3000

ETL: перенесет данные из JSON файлов и загрузит их в базу

▼ Инструкция по эксплуатации

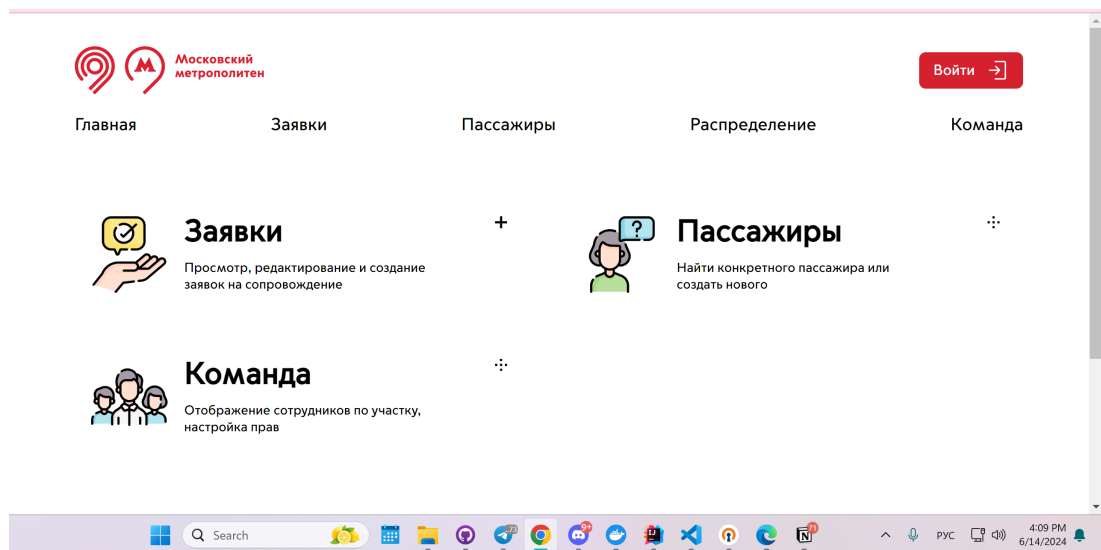
▼ Описание доступных API методов и их параметров и ожидаемых ответов

Динамическое описание API методов доступно в Swagger по ссылке:

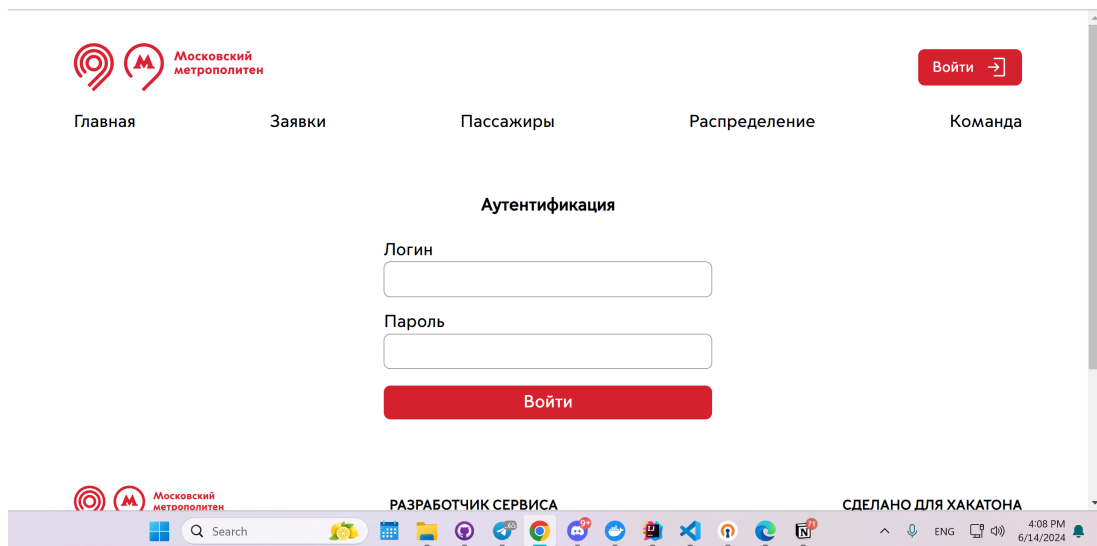
<https://api.метровтандеме.рф/swagger-ui/index.html>

▼ Инструкции по аутентификации и авторизации

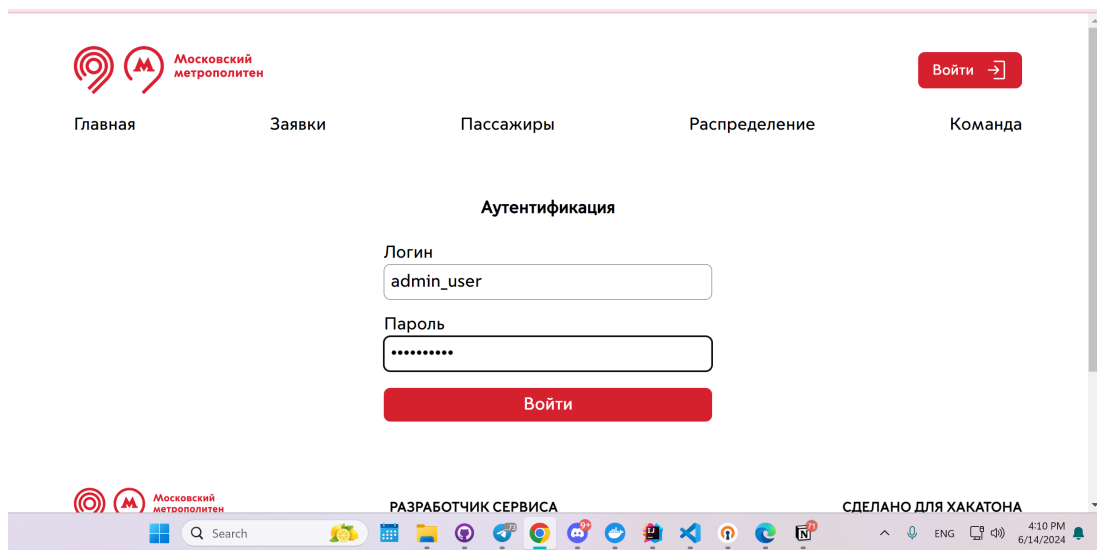
1. Посетите сайт <https://метровтандеме.рф>



2. Нажмите "Войти" или любую кнопку из меню навигации



3. Введите логин и пароль, по умолчанию, доступны следующие учетные записи с ролями:



1. Администратор

```
admin_user  
admin_pass
```

2. Специалист

```
specialist_user  
specialist_pass
```

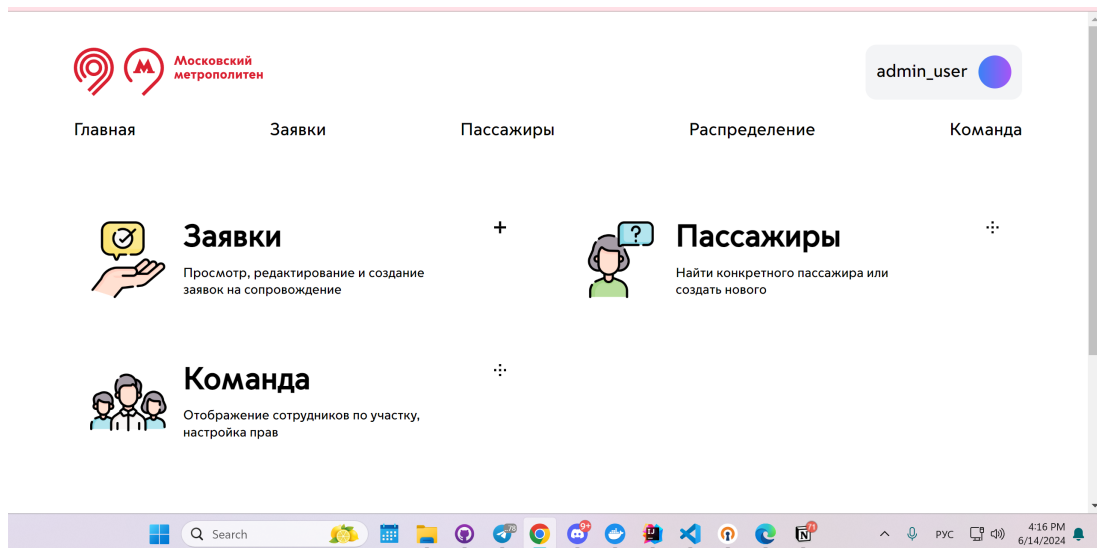
3. Оператор

```
operator_user  
operator_pass
```

4. Работник

```
admin_user  
admin_pass
```

4. Справа сверху исчезнет кнопка "Войти" и появится активная учетная запись



5. Для выхода из учетной записи нажмите на кнопку справа сверху, далее: Выйти

