|  |  |
| --- | --- |
|  |  |
| Система управления проектами разработки месторождений «СУПРМ»Инструкция по установке ПО |
| Санкт-Петербург 2024 |

Содержание

[1 Введение 3](#_Toc168578641)

[1.1 Общее описание 3](#_Toc168578642)

[1.2 Описание архитектуры 3](#_Toc168578643)

[2 Описание по Установки и запуску 5](#_Toc168578644)

[2.1 Рекомендуемые версии Node.js и npm 5](#_Toc168578645)

[2.2 Схема устанавливаемых компонент. 5](#_Toc168578646)

[2.3 Инструкция по запуску IPTD Блок 1 6](#_Toc168578647)

[2.3.1 Инструкция по запуску backend 6](#_Toc168578648)

[2.3.2 Инструкция по запуску frontend 7](#_Toc168578649)

[2.4 Инструкция по запуску IPTD Блок 2 8](#_Toc168578650)

[2.4.1 Инструкция по запуску backend 8](#_Toc168578651)

[2.4.2 Инструкция по запуску frontend 9](#_Toc168578652)

[Служба поддержки: 10](#_Toc168578653)

#  Введение

## Общее описание

Программный продукт «СУПРМ» представляет собой инструмент непрерывного анализа и проектирования активов компании, осуществляющих разработку месторождений углеводородного сырья, позволяющий в оперативном режиме учитывать все изменения в структуре ресурсной базы, разрабатывать проектные решения и реализовывать их в технологических расчетах, получать оценки экономической эффективности и формировать выходные отчетные формы.

## Описание архитектуры

Продукт реализуется как web-решение, основные компоненты которого представлены на рисунке 1.



Рис. 1. Архитектура СУПРМ

Frontend-компонент отвечает за ввод/вывод данных и отчетных документов, за отображение результатов расчётов и проектных решений.

Backend-компонент отвечает за управление проектами, за обработку данных и за взаимодействие с внешней ИС QGIS.

СУБД-компонент отвечает за хранение первичных данных, промежуточных результатов и отчетной документации.

Версии компоненты системы:

Frontend-компонент: Javascript

Backend-компонент: Python 3.8, Java 8

СУБД-компонент: PostgreSQL

Исходный код программы расположен

https://rosreestr:glpat-oH3ysmNHXYVyyJZJrDsy@gitlab.lanit-tercom.com/suprm/iptd;

#  Описание по Установки и запуску

## Рекомендуемые версии Node.js и npm

Рекомендуемая версия Node.js - 16.

Рекомендуемая версия npm - 8. Обратите внимание, что при использовании версии npm 20 могут возникнуть проблемы при запуске.

## Схема устанавливаемых компонент

Схема компонент и порядок установки представлен на рисунке 2.



Рис.2 Схема компонент системы и последовательность установки.

## Инструкция по запуску IPTD Блок 1

### Инструкция по запуску backend

#### Создание .env файла:

* Создайте файл с названием .env на основе файла .env.example.
* В открывшемся файле .env установите следующие значения:
* IPTD\_USE\_SUPERSET=False
IPTD\_BACK\_PATH=<путь\_до\_папки\_back>
IPTD\_FRONT\_PATH=<путь\_до\_папки\_ui>
* Убедитесь, что пути к папкам back и ui указаны правильно. Это важно для корректного запуска jar пакетов в бэкенде.

#### Запуск Docker контейнера с базой данных:

* Перейдите в директорию back.
* Запустите Docker контейнер с базой данных, установите дамп БД:

docker run -d -p 5432:5432 --name pg\_dev -e POSTGRES\_PASSWORD=simple postgres

docker exec -it pg\_dev psql -U postgres -c "create database iptd;"
cat stand-2024\_04\_18\_18\_05\_20-dump.sql | docker exec -i pg\_dev psql -U postgres -d iptd

* Обратите внимание, что порт 5432 используется в качестве примера. Вы можете использовать другой порт, но не забудьте обновить его в поле IPTD\_BD\_PORT файла .env.
* stand-2024\_04\_18\_18\_05\_20-dump.sql - это пример дампа базы данных.

#### Настройка PyCharm:

* В PyCharm откройте настройки конфигуратора запуска.
* В поле Script path укажите путь до файла back/server.py.
* В поле Working directory установите значение /back.

#### Установка PostgreSQL:

* Перед запуском backend убедитесь, что в вашей системе установлен PostgreSQL. Это необходимо для работы драйверов базы данных в backend.

#### Первый запуск backend:

* Рекомендуемая версия Python - 3.10 (более поздняя версия может вызвать ошибку установки).
* После успешного выполнения предыдущих шагов запустите backend.
* При первом успешном запуске backend самостоятельно заполнит базу данных.

### Инструкция по запуску frontend

#### Переход в директорию ui:

* Откройте терминал и перейдите в директорию, содержащую код frontend.
* cd path/to/ui

#### Создание файла .env:

* Создайте файл .env в директории ui и скопируйте содержимое файла .env.example в него.

#### Установка зависимостей:

* Введите следующую команду в терминале для установки зависимостей проекта:

npm install --legacy-peer-deps

* Эта команда обеспечит установку всех необходимых зависимостей, включая зависимости, которые могут вызвать ошибки из-за их устаревшей версии.

#### Запуск frontend в режиме разработки:

* После успешной установки зависимостей введите следующую команду для запуска frontend в режиме разработки:
* npm run dev

## Инструкция по запуску IPTD Блок 2

### Инструкция по запуску backend

#### Создание .env файла:

* Создайте файл с названием .env на основе файла .env.example.
* В открывшемся файле .env установите следующие значения:
* IPTD\_USE\_SUPERSET=False
IPTD\_BACK\_PATH=<путь\_до\_папки\_back>
IPTD\_FRONT\_PATH=<путь\_до\_папки\_ui>
* Убедитесь, что пути к папкам back и ui указаны правильно. Это важно для корректного запуска jar пакетов в backend.

#### Запуск Docker контейнера с базой данных:

* Перейдите в директорию back.
* Запустите Docker контейнер с базой данных, запустите установку дампа БД:

docker build -t iptd\_bd
docker run --name iptd\_bd -p 5433:5432 iptd\_bd

* Обратите внимание, что порт 5433 используется в качестве примера. Вы можете использовать другой порт, но не забудьте обновить его в поле IPTD\_BD\_PORT файла .env

#### Установка PostgreSQL:

* Перед запуском backend убедитесь, что в вашей системе установлен PostgreSQL. Это необходимо для работы драйверов базы данных в бэкенде.

#### Первый запуск backend:

* Рекомендуемая версия Python - 3.10 (более поздняя версия может вызвать ошибку установки)
* После успешного выполнения предыдущих шагов запустите backend.
* При первом успешном запуске backend самостоятельно заполнит базу данных.

### Инструкция по запуску frontend

#### Переход в директорию ui:

* Откройте терминал и перейдите в директорию, содержащую код fronend.
* cd path/to/ui

#### Создание файла .env:

* Создайте файл .env в директории ui и скопируйте содержимое файла .env.example в него.

#### Установка зависимостей:

* Введите следующую команду в терминале для установки зависимостей проекта:

npm install --legacy-peer-deps

* Эта команда обеспечит установку всех необходимых зависимостей, включая зависимости, которые могут вызвать ошибки из-за их устаревшей версии.

#### Запуск frontend в режиме разработки:

* После успешной установки зависимостей введите следующую команду для запуска frontend в режиме разработки:
* npm run dev

# Служба поддержки:

ООО «ЛАНИТ-ТЕРКОМ»

198504, Санкт-Петербург, г. Петергоф, Чичеринская ул., д. 2, литера А, пом. 5-Н

тел.+7 812 922 20 91

e-mail: sales@lanit-tercom.com

Ответственное лицо:

тел.+7 953 357 79 37

e-mail: vasily.kazantsev@lanit-tercom.ru