

Инструкция по установке

Проект состоит из двух частей - клиентской и серверной. Сервер отвечает за обработку постановки задач. Клиент отвечает за интерфейс взаимодействия с системой.

Серверная часть реализована на Java (17+) и Spring Boot. Клиентская часть реализована на Python (3.11+)

Обмен данными между клиентом и сервером - Json по RestAPI. Поддерживаются Linux/Windows/MacOS Программа рассчитана на создание сторонних клиентов

Инструкция для Linux

Запуск серверной части

1. Установить Java

```
sudo apt update && sudo apt upgrade -y  
sudo apt install openjdk-17-jdk openjdk-17-jre  
java -version
```

2. Распаковать архив с программой

```
cd ~/Downloads  
unzip task_manager.zip -d <куда_распакуем>
```

3. Запустить терминал Нажимаем сочетание `ctrl + alt + t`

4. Перейти в папку с программой сервера Пишем в терминале:

```
cd ~/<куда_распаковали>/task_manager/src/server/task_scheduler
```

5. Запустить сервер командой в терминале:

```
java -jar target/task_scheduler-1.0-SNAPSHOT.jar
```

Для работы сервера терминал должен оставаться запущенным, база данных сохраняется при следующих запусках

Запуск клиентской части

1. Установить Python 3 (версия от 3.10.12) На Linux python установлен по умолчанию, рекомендуется обновить до версии не ниже 3.10.12. Проверить текущую версию можно прописав в терминал:

```
python3 --version
```

2. Распаковать архив с программой
3. Запустить терминал
Нажимаем сочетание `ctrl + alt + t`
4. Перейти в папку с программой клиента через терминал Пишем в терминале:

```
cd ~/<куда_распаковали>/task_manager/src/client
```

5. (Опционально) Установить виртуальное окружение python Прописываем команды в терминал:

```
sudo apt install python3-venv  
python3 -m venv venv  
source venv/bin/activate
```

6. Установить все зависимости командой `pip install -r requirements.txt`
7. В файле конфигурации `config.json` в папке программы задать в параметре `url` адрес и порт сервера (при тестовом запуске можно не редактировать)
8. Для запуска программы программы в режиме обычного пользователя написать в терминале:

```
python3 user.py
```

9. Для запуска программы программы в режиме администратора написать в терминале:

```
python3 admin.py
```

Пример работы с программой приведен в прилагаемом видео.

Инструкция для Windows

Запуск серверной части

1. Установить Java Установка Java 17 производится с [официального сайта](#)

- Необходимо выбрать в списке в нижней части экрана версию JDK 17 и загрузить установщик x64 MSI Installer
- Запустить загрузчик и нажать кнопку Install
- Перезагрузить компьютер в случае появления соответствующего сообщения

Иногда после установки система не видит Java, в таком случае нужно вручную указать системные переменные. Чтобы это сделать:

- В строке "Поиск" выполните поиск: Система (Панель управления)
- Нажмите на ссылку Дополнительные параметры системы.
- Нажмите Переменные среды. В разделе Системные переменные нажмите кнопку Создать.
- В окне создания системной переменной укажите имя переменной JAVA_HOME и значение C:\Program Files\Java\jdk-17\bin в случае, если система установлена на диске C:\.
- Нажмите ОК. Закройте остальные открытые окна, нажимая ОК.

2. Распаковать архив с программой в ту же папку
3. Запустить терминал Powershell: В строке поиска Windows вбиваем Powershell и открываем от имени администратора
4. Перейти в папку с программой сервера Пишем в терминале:

```
cd Downloads/task_manager/src/server/task_scheduler
```

5. Запустить сервер командой в терминале:

```
java -jar target/task_scheduler-1.0-SNAPSHOT.jar
```

Для работы сервера терминал должен оставаться запущенным, база данных сохраняется при следующих запусках

Запуск клиентской части

1. Установить Python 3 (версия от 3.10.12)
 - В магазине Microsoft Store ищем "Python 3.12" и скачиваем
 - Перезагрузить компьютер в случае появления соответствующего сообщения
2. Распаковать архив с программой
3. Запустить терминал Powershell: В строке поиска Windows вбиваем Powershell и открываем от имени администратора
4. Перейти в папку с программой клиента через терминал Пишем в терминале:

```
cd Downloads/task_manager/
```

5. (Опционально) Установить виртуальное окружение python Прописываем команды в терминал:

```
python -m venv venv  
source ./venv/Scripts/activate
```

6. Установить все зависимости командой в терминале:

```
pip install -r requirements.txt
```

7. В файле конфигурации `config.json` в папке программы задать в параметре `url` адрес и порт сервера (при тестовом запуске можно не редактировать). Если программа в дальнейшем будет запускаться двойным кликом на исполняемый файл, `config.json` настраивается в папке администратор и в папке пользователь

8. Для запуска программы в режиме обычного пользователя доступно 2 варианта:

- переместить папку `пользователь` из папки `exe files` на рабочий стол и запустить в ней исполняемый файл `менеджер`
- выполнить последовательно команды в терминале из папки с проектом:

```
cd src/client/  
python user.py
```

9. Для запуска программы в режиме администратора доступно 2 варианта:

- переместить папку `администратор` из папки `exe files` на рабочий стол и запустить в ней исполняемый файл `менеджер`
- выполнить последовательно команды в терминале из папки с проектом:

```
cd src/client/  
python admin.py
```

Пример работы с программой приведен в прилагаемом видео.