

Общество с ограниченной ответственностью «Си Икс Лаб»

## **РЕШЕНИЕ ПО СОЗДАНИЮ АВТОМАТИЗИРОВАННЫХ РАБОЧИХ МЕСТ**

### **СХВОХ V6.0**

#### **ИНСТРУКЦИЯ ПО УСТАНОВКЕ**

Листов 18

Контактное лицо:

Владелец продукта: Сухаренко Маргарита Игоревна

e-mail: [msukharenko@cxlabs.ru](mailto:msukharenko@cxlabs.ru)

Москва

2024

## СОДЕРЖАНИЕ

1. Состав системы .....	3
2. Исходные коды.....	4
3. Требования к техническому и программному обеспечению .....	5
4. Установка системы.....	13
5. Установка дополнений разработчика.....	16

## 1. Состав системы

В состав решения CXBOX v6.0 входят следующие компоненты:

- Ядро;
- Системные дополнения: «S3 File Storage», «Интеграционный модуль», «Базовая авторизация», «Аутентификация», «Поддержка различных СУБД»;
- Дополнения разработчика: «Шаблонизатор кода», «Дополнение быстрой навигации», «Автодополнение кода», «Поиск и исправление ошибок», «Инструменты мониторинга», «Интерактивная документация», «Навигация к документации»;
- Бизнес-дополнения: «Интеграция с email», «Push-уведомления», «Предварительный просмотр документов»;
- Дополнения администратора: «Scheduler».

## 2. Исходные коды

Для хранения исходных кодов используется распределенная система управления версиями Git. Она используется для отслеживания изменений в исходном коде программного обеспечения и управления разработкой проектов.

Git предоставляет разработчикам возможность сохранять историю изменений, создавать ветки для параллельной разработки, сливать изменения из разных веток и откатывать изменения к предыдущим версиям. Это позволяет упростить совместную работу в команде и управление разработкой проектов.

Одной из главных особенностей Git является его распределенная природа. Каждый разработчик имеет полную копию репозитория Git на своем локальном компьютере, что позволяет им работать независимо от подключения к сети или удаленному серверу. Когда разработчик выполняет коммит изменений, они сохраняются локально, а затем могут быть отправлены на удаленный Git-сервер (Remote Git) для совместного доступа и резервного копирования.

Remote Git - это термин, используемый для обозначения удаленного репозитория Git. Remote Git-сервер обычно предоставляет доступ к репозиторию через сетевое соединение, такое как HTTP, SSH или другие протоколы. Публикация исходного кода на удаленном Git-сервере позволяет другим разработчикам получить доступ к проекту, вносить изменения и синхронизироваться с общей историей изменений.

Исходные коды для загрузки доступны со следующих Git-remote серверов:

1. Внутренние Remote репозитории компании.
2. <https://gitflic.ru/project/cxbox/cxbox-demo>
3. <https://github.com/CX-Box/cxbox-demo> (зеркало для п.2)

### 3. Требования к техническому и программному обеспечению

Для сборки приложения на основе CXBOX v6.0 необходим персональный компьютер со следующей рекомендованной конфигурацией:

- многоядерный современный Intel или AMD процессор с частотой не менее 1800 МГц;
- объем оперативной памяти не менее 8 Гб;
- объем свободного места на жестком диске – не менее 10 Гбайт (рекомендовано использовать SSD-диск);
- разрешение экрана — не менее 1920 x 1080 пикселей;
- клавиатура, манипулятор «мышь»;
- монитор с диагональю экрана не менее 19 дюймов;
- последняя 64-разрядная версия Windows, macOS или Linux;
- наличие связи с Интернет (для подключения репозитория и установления необходимых зависимостей).

Перед сборкой приложения на компьютер должно быть установлено следующее программное обеспечение:

#### 3.1. Установка ПО на Windows

- Backend:
  - Java версии 17 (<https://jdk.java.net/java-se-ri/17-MR1>  
<https://axiomjdk.ru/pages/downloads/#/java-17-lts>);

##### **RI Binary (build 17.0.0.1+2) under the GNU General Public License version 2**

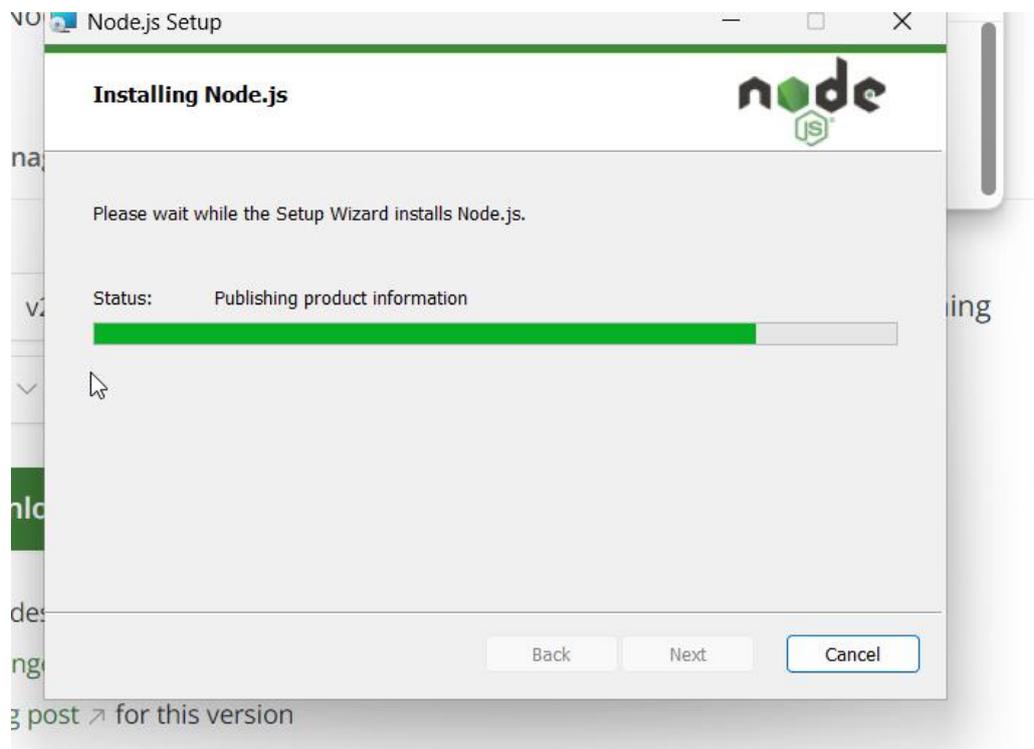
- Oracle Linux 8.7 x64 Java Development Kit (sha256) 179 MB
- **Windows 11 x64 Java Development Kit (sha256) 178 MB**

##### **Загрузка Java**

- Apache Maven версии 3.9.9 (<https://maven.apache.org/download.cgi>);

– Frontend:

- Node.js версии 20.18.1 (<https://nodejs.org/en/download/prebuilt-installer>):



### Установка Node.js

- NPM 10.8.2 (устанавливается вместе с node.js – отдельной установки не требует);
- Yarn 4.0.2 (<https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable> npm install --global yarn);

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PowerShellLatest

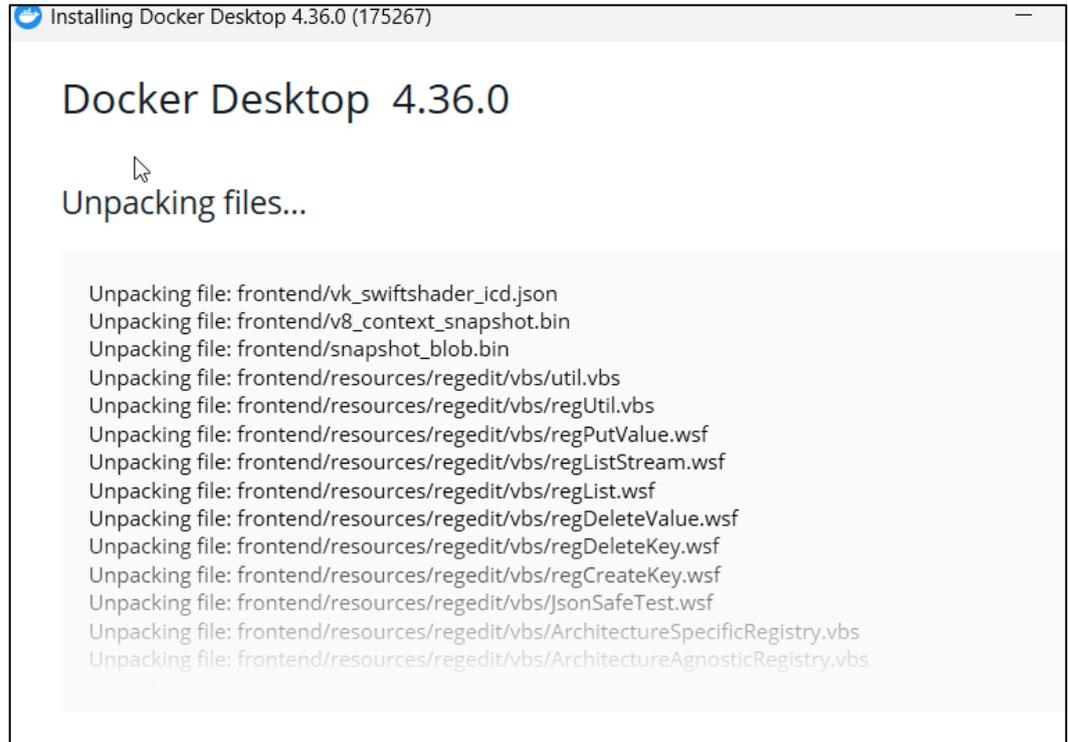
PS C:\Users\vboxuser> npm install --global yarn

changed 1 package in 2s
PS C:\Users\vboxuser> |
```

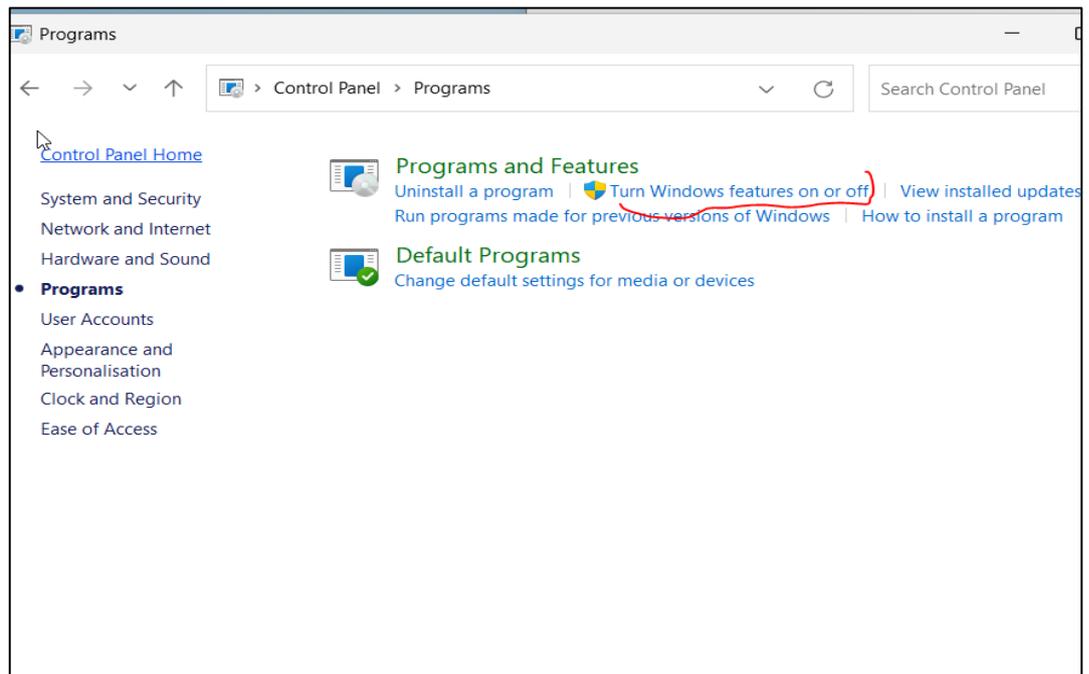
### Инсталляция Yarn

– Инфраструктура:

- Docker (<https://docs.docker.com/desktop/setup/install/windows-install/>);
- Docker Compose (<https://docs.docker.com/compose/install/>);
- <https://www.docker.com/products/docker-desktop/>



### Загрузка Docker Desktop

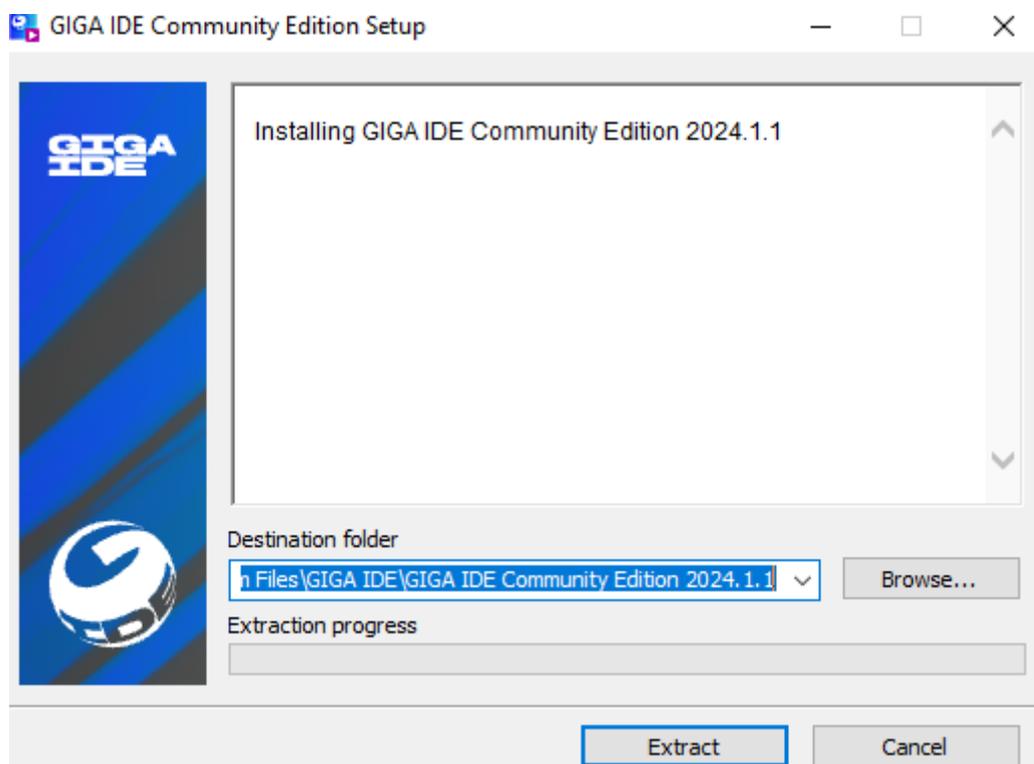


### Включение виртуализации

– Среда разработки:

- GigaIDE 2024.1.1 (<https://gitverse.ru/features/gigaide/desktop/download/>).

Альтернатива IntelliJ IDEA Community (<https://www.jetbrains.com/ru-ru/idea>).



Установка Giga IDE

### 3.2. Установка ПО на Linux

– Backend:

- Java версии 17 (<https://jdk.java.net/java-se-ri/17-MR1> <https://axiomjdk.ru/pages/downloads/#/java-17-lts> );

```
RI Binary (build 17.0.0.1+2) under the GNU General Public License version 2
  • Oracle Linux 8.7 x64 Java Development Kit (sha256) 179 MB
```

Установка java

- Apache Maven версии 3.9.9 (<https://maven.apache.org/download.cgi> );

– Frontend:

- Node.js версии 20.18.1 (<https://github.com/nvm-sh/nvm/blob/master/README.md>)

Package Manager Prebuilt Installer Prebuilt Binaries Source Code

Install Node.js v20.18.1 (LTS) on Linux using nvm

```
1 # installs nvm (Node Version Manager)
2 curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
3
4 # download and install Node.js (you may need to restart the terminal)
5 nvm install 20
6
7 # verifies the right Node.js version is in the environment
8 node -v # should print `v20.18.1`
9
10 # verifies the right npm version is in the environment
11 npm -v # should print `10.8.2`
```

Bash [Copy to clipboard](#)

Package managers and their installation scripts are not maintained by the Node.js project.  
If you encounter issues, please reach out to the package manager's maintainers.

Node.js includes [npm \(10.8.2\)](#).

[Read the changelog](#) for this version

[Read the blog post](#) for this version

[Learn how to verify signed SHASUMS](#)

## Установка Node.js

```
root@vbox:~# sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  curl
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 315 kB of archives.
After this operation, 500 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-10+deb12u8 [315 kB]
Fetched 315 kB in 0s (670 kB/s)
Selecting previously unselected package curl.
(Reading database ... 151308 files and directories currently installed.)
Preparing to unpack ../curl_7.88.1-10+deb12u8_amd64.deb ...
Unpacking curl (7.88.1-10+deb12u8) ...
Setting up curl (7.88.1-10+deb12u8) ...
Processing triggers for man-db (2.11.2-2) ...
root@vbox:~# installs nvm (Node Version Manager)
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
-bash: syntax error near unexpected token `('
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             Dload  Upload   Total   Spent    Left   Speed
100 16563  100 16563    0     0  64243    0 --:--:-- --:--:-- --:--:--  64197
=> Downloading nvm as script to '/root/.nvm'

=> Appending nvm source string to /root/.bashrc
=> Appending bash_completion source string to /root/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
root@vbox:~#
```

## Инсталляция curl

- Далее перезайти в терминал;

- NPM 10.8.2 (устанавливается вместе с node.js – отдельной установки не требует);

```
vboxuser@vbox1:~$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.1/install.sh | bash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload  Total  Spent    Left  Speed
100 16563  100 16563    0     0  63425      0  --:--:-- --:--:-- --:--:-- 63703
=> Downloading nvm as script to '/home/vboxuser/.nvm'

=> Appending nvm source string to /home/vboxuser/.bashrc
=> Appending bash_completion source string to /home/vboxuser/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm bash_completion
vboxuser@vbox1:~$ export NVM_DIR="$([ -z "${XDG_CONFIG_HOME-}" ] && printf %s "${HOME}/.nvm" || printf %s "${XDG_CONFIG_HOME}/nvm")"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
```

### Установка nvm

- Далее перезайти в терминал;
- Yarn 4.0.2 (<https://classic.yarnpkg.com/lang/en/docs/install/#windows-stable> npm install --global yarn);

```
vboxuser@vbox1:~$ npm -v
10.8.2
vboxuser@vbox1:~$ node -v
v20.18.1
vboxuser@vbox1:~$ npm install -global yarn

added 1 package in 960ms
npm notice
npm notice New major version of npm available! 10.8.2 -> 11.0.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.0.0
npm notice To update run: npm install -g npm@11.0.0
npm notice
vboxuser@vbox1:~$ \
```

### Проверка установки

– Инфраструктура:

- Docker (<https://docs.docker.com/engine/install/>,  
<https://docs.docker.com/engine/install/linux-postinstall/> )
- Docker Compose ( <https://docs.docker.com/compose/install/linux/>)

```
vboxuser@vbox1:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io doc
ker-buildx-plugin docker-compose-plugin
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Package docker-ce is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

Package docker-ce-cli is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'docker-ce' has no installation candidate
E: Package 'docker-ce-cli' has no installation candidate
E: Unable to locate package containerd.io
E: Couldn't find any package by glob 'containerd.io'
E: Couldn't find any package by regex 'containerd.io'
```

### Установка docker

```
vboxuser@vbox1:~$ sudo groupadd docker
vboxuser@vbox1:~$ sudo usermod -aG docker $USER
vboxuser@vbox1:~$
```

### Последующие действия

```
bash: cd: too many arguments
vboxuser@vbox1:~$ sudo apt install docker-compose
[sudo] password for vboxuser:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libslirp0 pigz slirp4netns
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cgroupfs-mount containerd
  criu docker.io libbinutils libctf-nobfd0 libctf0 libgprofng0 libint1-perl
  libint1-xs-perl libmodule-find-perl libnet1 libproc-processtable-perl
  libsort-naturally-perl libterm-readkey-perl needrestart python3-attr
  python3-distutils python3-docker python3-dockerpty python3-docopt
  python3-dotenv python3-json-pointer python3-jsonschema python3-lib2to3
  python3-protobuf python3-pyrsistent python3-rfc3987 python3-texttable
  python3-uritemplate python3-webcolors python3-websocket python3-yaml runc
  tini
Suggested packages:
  binutils-doc containernetworking-plugins docker-doc aufs-tools btrfs-progs
  debootstrap rinse rootlesskit xfsprogs zfs-fuse | zfsutils-linux
```

### Установка docker-compose

- Среда разработки:
    - o GigaIDE 2024.1.1 (<https://gitverse.ru/features/gigaide/desktop/download/>).
- Альтернатива IntelliJ IDEA Community (<https://www.jetbrains.com/ru-ru/idea>).
- Загружаем GigaIDE разархивируем, запускаем.

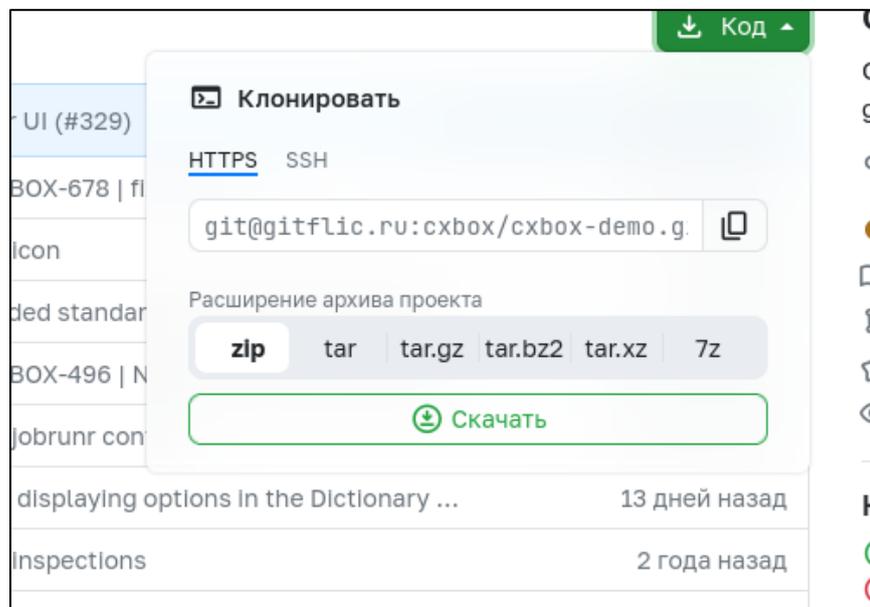
```
vboxuser@vbox: ~/Downloads/gigaide-CE-241.15989.150/bin$ ./idea.sh
CompileCommand: exclude com/intellij/openapi/vfs/impl/FilePartNodeRoot.trieDescend bool exclude = true
2024-12-24 17:00:28,851 [ 8230] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
- Cannot find scheme: VibrantInk from plugins: epoch.DBN
2024-12-24 17:00:28,855 [ 8234] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
- Cannot find scheme: Monocai from plugins: epoch.DBN
2024-12-24 17:00:28,907 [ 8286] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
- Cannot find scheme: Twilight from plugins: epoch.DBN
2024-12-24 17:00:28,908 [ 8287] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
- Cannot find scheme: WarmNeon from plugins: epoch.DBN
2024-12-24 17:00:28,910 [ 8289] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
- Cannot find scheme: Cobalt from plugins: epoch.DBN
2024-12-24 17:00:28,911 [ 8290] WARN - #c.i.o.e.c.i.EditorColorsManagerImpl
```

**Запуск GigaIDE**

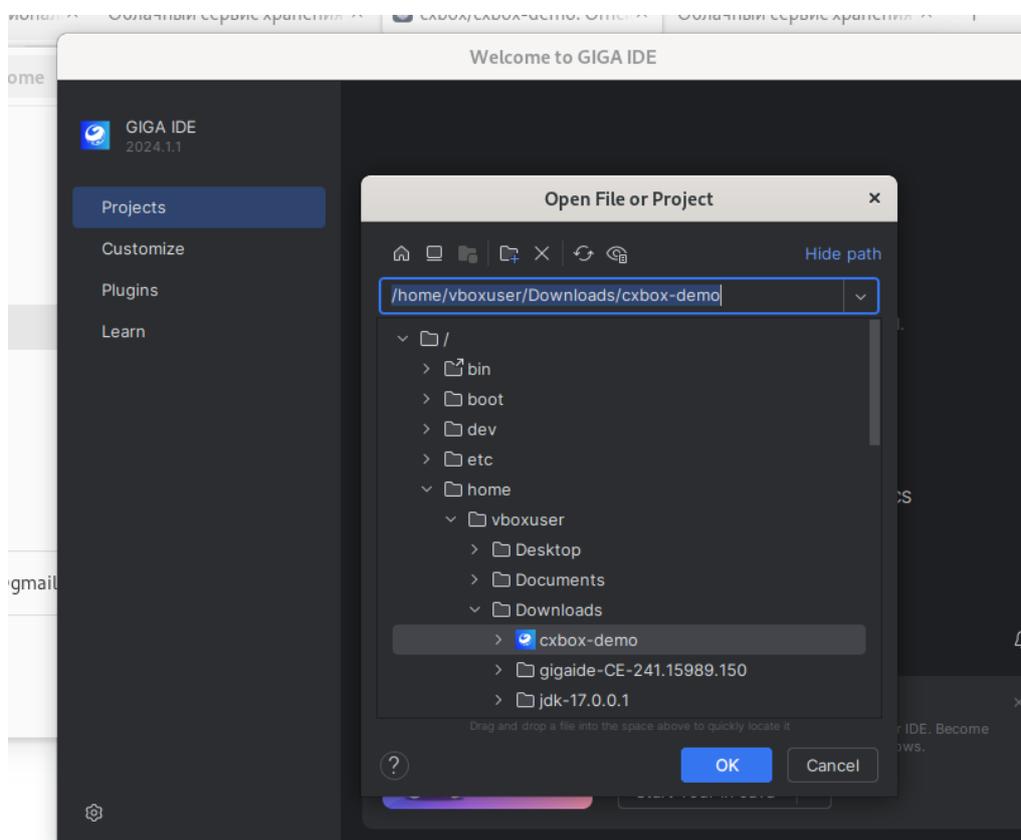
## 4. Установка системы

Установка решения CXBOX v6.0 (включает в себя ядро, системные дополнения, бизнес дополнения и дополнения администратора) производится в следующем порядке:

1. Скачать архив репозитория (<https://gitflic.ru/project/cxbox/cxbox-demo>):

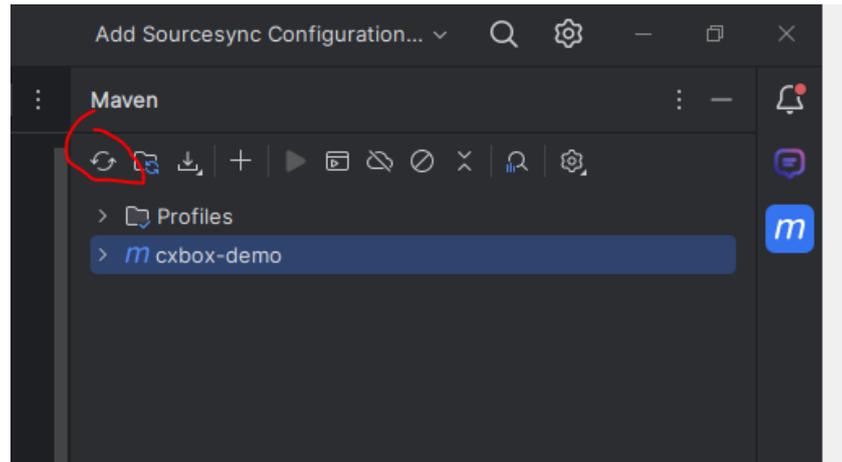


Скачивание репозитория

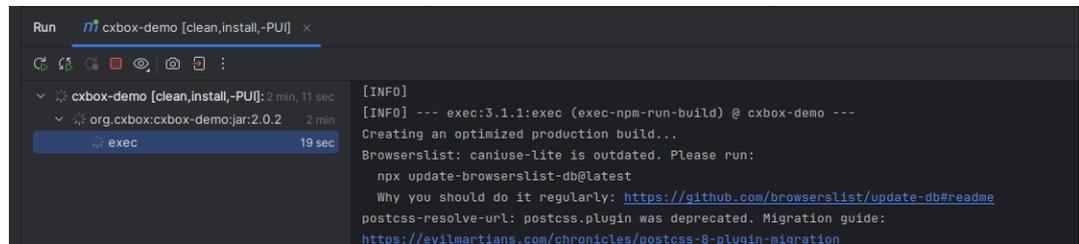


Открытие проекта

2. Установить зависимости и собрать фронт-энд и бэк: `mvn clean install -PUI -U`

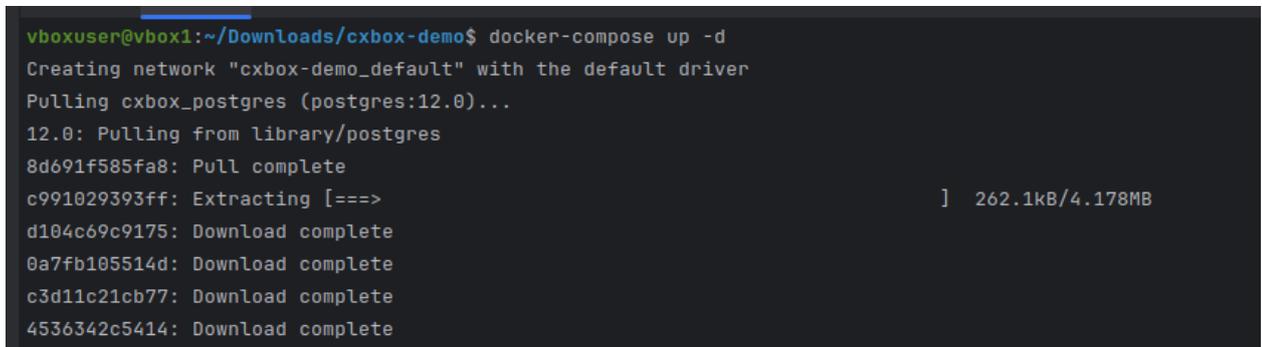


### Подтягиваем зависимости



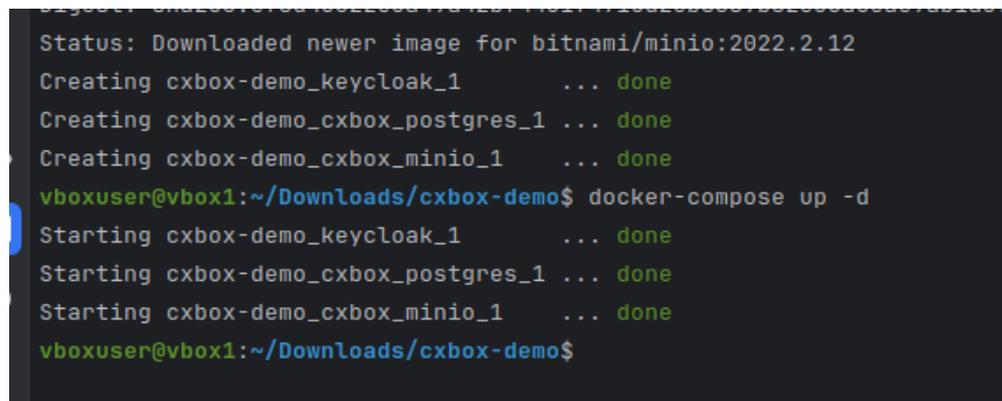
### Сборка проекта при помощи maven

3. Запустить среду: `docker-compose up -d`



### Запуск контейнеров окружения

3.1. Проверяем поднятие контейнеров: `docker ps`



### Проверка контейнеров

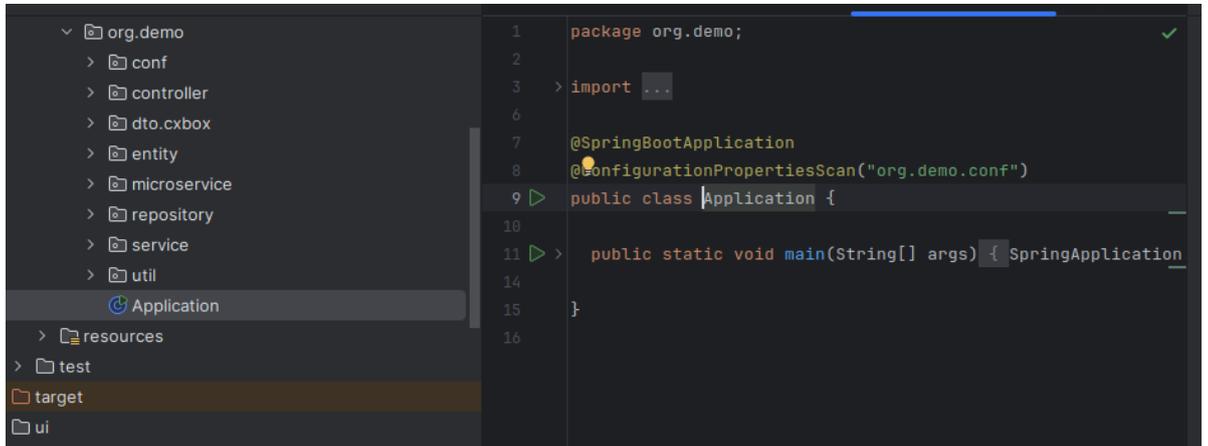
3.2. В случае проблем при запуске контейнера keycloak:

В файле docker-compose.yml удалить строку

```
- ./tmp/keycloak-db:/opt/jboss/keycloak/standalone/data
```

Запустить еще раз docker-compose up -d

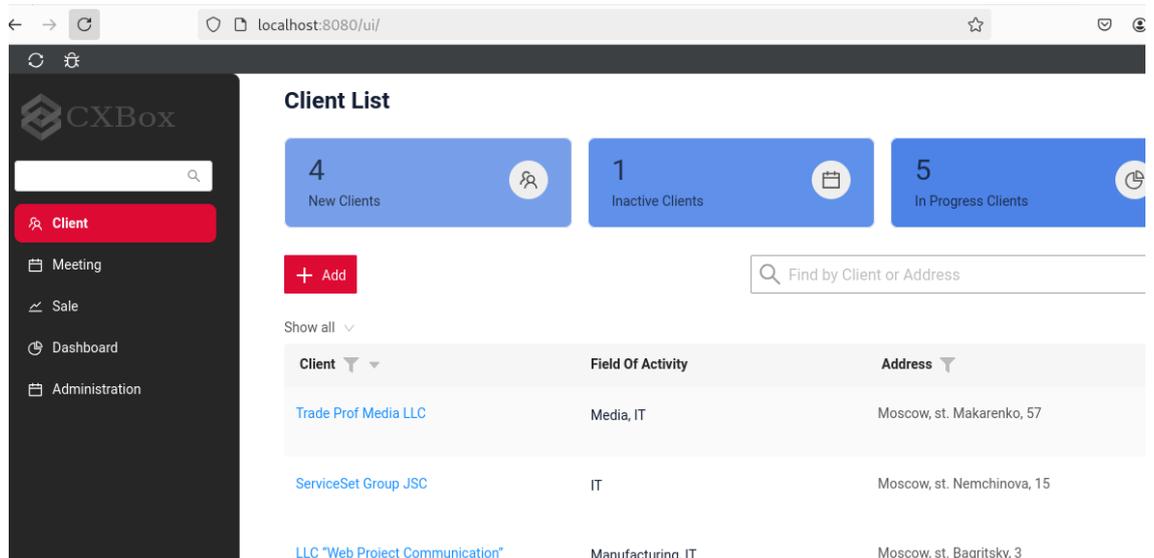
4. Запустить приложение: нажать на зеленую кнопку в Application.java.



```
1 package org.demo;
2
3 > import ...
4
5
6
7 @SpringBootApplication
8 @ConfigurationPropertiesScan("org.demo.conf")
9 > public class Application {
10
11 > public static void main(String[] args) { SpringApplication
12
13
14
15 }
16
```

### Запуск приложения

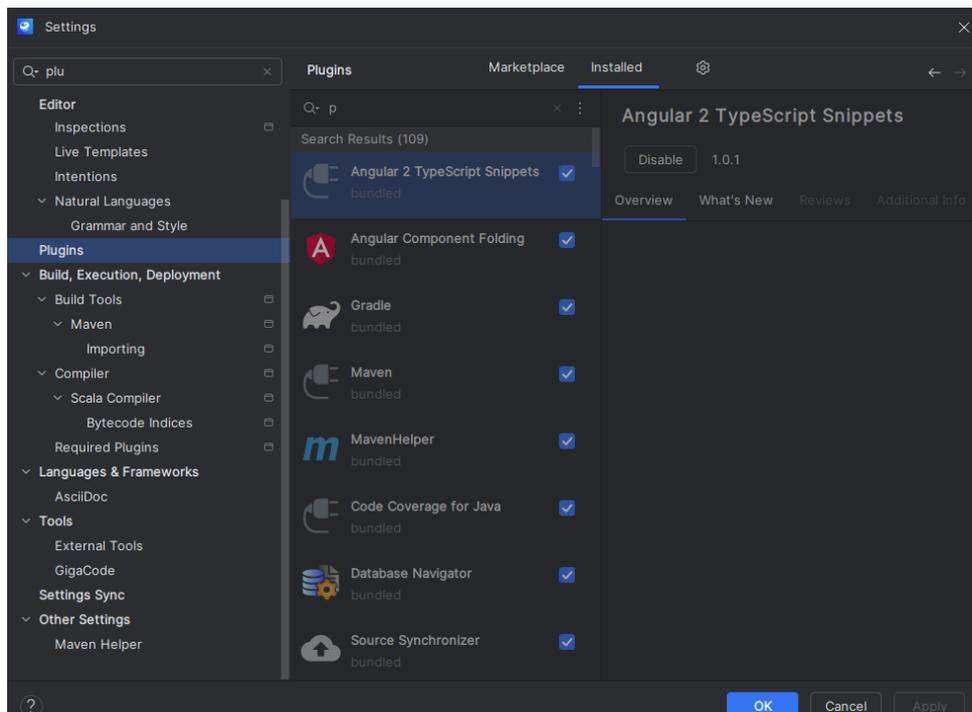
5. Зайти в приложение localhost:8080



## 5. Установка дополнений разработчика

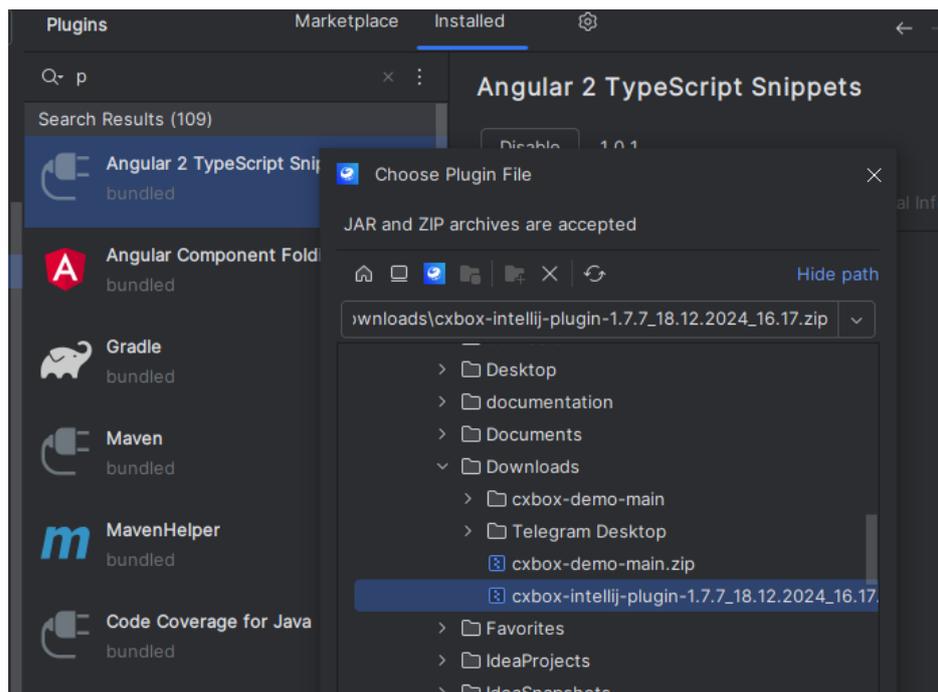
Установка дополнений разработчика производится в следующем порядке:

1. Загрузить Plugin по ссылке [https://disk.yandex.ru/d/MWls9m\\_TPiBdQA](https://disk.yandex.ru/d/MWls9m_TPiBdQA)
2. Запустить GigaIDE.
3. Открыть настройки (нажать Ctrl+Alt+S). Написать Plugins в поиске.



Экран настройки-плагины

4. Открыть  и нажать Install Plugin from Disk.



Добавление плагина

5. Выбрать файл архива дополнения и нажать ОК.

Нажать ОК для применения изменений и перезапустить GigaIDE .

```
1  {
2  "name": "clientList",
3  "title": "",
4  "type": "List",
5  "bc": "client",
6  "fields": [
7    {
8      "title": "Client",
9      "key": "fullName" String: ,
10     "type": "input",
11     "drillDown": "true",
12     "width": 300
13   },
14   {
15     "title": "Field Of Activity",
16     "key": "fieldOfActivity" MultivalueField: ,
17     "type": "multipleSelect",
18     "width": 300
19   },
20   {
21     "title": "Address",
22     "key": "address" String: ,
23     "type": "input",
24     "width": 300
25   },
26   {
27     "title": "Importance",
28     "key": "importance" ClientImportance: ,
```

**Работа плагина**

```
1 {
2   "name": "clientList",
3   "title": "",
4   "type": "List",
5   "bc": "client",
6
7
8
9   String: ,
10
11   ",
12
13   "Activity",
14
15   "Activity" MultivalueField: ,
16   "Select",
17
18
19
20
21
22   String: ,
23
24
25
26
27   "nce",
28   " ClientImportance: ,
29   "y",
30   "bgColorKey": "color" String: ,
31   "width": 300
32 },
33 {
34   "title": "Status",
35   "key": "status" ClientStatus: ,
```

**Работа плагина**