

Wirow development

1. General prerequisites

- OS Linux (Ubuntu preferred)
- CMake v3.18
- Ninja build system v1.10
- Nodejs LTS version
- Yarn v1.22.5
- Clang compiler v10 or GCC compiler v9
- libcunit1-dev
- Uncrustify code formatting tool
- VS Code

2. Ubuntu / Debian setup

2.1. Required packages

```
sudo apt-get install git make ninja-build libcunit1-dev \
gcc g++ build-essential make \
uncrustify python-is-python3 \
yasm autoconf pkgconf libtool \
autoconf automake make yasm
```

2.2. CMake

```
wget -O - https://apt.kitware.com/keys/kitware-archive-latest.asc 2>/dev/null
| gpg --dearmor - | sudo tee /etc/apt/trusted.gpg.d/kitware.gpg >/dev/null

sudo apt-add-repository 'deb https://apt.kitware.com/ubuntu/ focal main'
sudo apt-get update

sudo apt-get install cmake
```

2.3. Nodejs

```
curl -sL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt-get install -y nodejs
```

2.4. Yarn

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -  
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee  
/etc/apt/sources.list.d/yarn.list
```

```
sudo apt update && sudo apt install yarn
```

2.5. CMake

```
wget -O - https://apt.kitware.com/keys/kitware-archive-latest.asc 2>/dev/null  
| gpg --dearmor - | sudo tee /etc/apt/trusted.gpg.d/kitware.gpg >/dev/null  
  
sudo apt-add-repository 'deb https://apt.kitware.com/ubuntu/ focal main'  
sudo apt-get update  
  
sudo apt-get install cmake
```

3. Project IDE

Visual Studio Code

Extensions list recommend for import before work

```
cat ./docs/vscode-extensions.txt | xargs -L 1 echo code --install-extension
```

4. Development build

```
git clone gitlab@dev.softmotions.com:softmotions/greenrooms.git  
cd ./greenrooms  
git submodule update --recursive --init
```

```
mkdir ./build && cd ./build  
  
cmake .. -G Ninja -DCMAKE_BUILD_TYPE=Debug \  
-DIW_EXEC=ON \  
-DBUILD_TESTS=ON \  
-DCREATE_DEV_USER=ON  
  
ninja
```

Run the server

```
./build/src/wirow -c config/testing.ini
```

In order to do clean rebuild just do `rm -rf ./build/*` then reconfigure and rebuild project again.

Development user with admin rights will be created on first run: (`-DCREATE_DEV_USER=ON`)

```
Username: dev
Password: grStart011
```

5. Release build

Release build is a self contained, static linked and doesn't depend on any shared libraries. [musl libc](#) is used.

5.1. Prerequisites

- OS Linux (Ubuntu preferred)
- CMake v3.18
- GNU Make
- Ninja build system v1.10
- Nodejs LTS version
- Yarn v1.22.5
- Clang compiler v10 or GCC compiler v9

5.2. Musl setup

```
git clone https://github.com/richfelker/musl-cross-make
cd ./musl-cross-make
cp ./config.mak.dist ./config.mak

nano ./config.mak
# Then uncomment:
TARGET = x86_64-linux-musl

make && make install

export MUSL_HOME=<musl git sources>/output
```

5.3. Build release

If you don't need integration with license server - just omit `-DLICENSE_REQUEST_FILE` option from cmake build arguments.

License request file

```
{
  "accessToken": "b9710f7b-e090-4777-8011-d0292664d367", // License server
access token
  "server": "https://localhost:8443", // Licence server address
  "owner": "License owner name",
  "login": "owner_login", // License owner login name on wirow server
  "email": "owner@email", // License owner email
  "address": "Owner address", // License owner address
  "terms": "perpetual" // License terms and conditions
}
```

Build

```
git clone gitlab@dev.softmotions.com:softmotions/greenrooms.git

cd ./greenrooms
git submodule update --recursive --init

mkdir ./build && cd ./build

# Note: MUSL_HOME environment variable must be set
cmake .. -G Ninja -DCMAKE_TOOLCHAIN_FILE='<absolute path to/musl-linux-x86-
64-tc.cmake>' \
        -DCMAKE_BUILD_TYPE=Release \
        -DIW_EXEC=ON \
        -DLICENSE_REQUEST_FILE='<absolute path to license request
file.json>'

ninja
```

You will find **wirow** executable in **./build/src**