ZROJ Core

本项目包含了 ZROJ 的所有核心库。

This project is currently under active development.

recommand VSCode extensions: `mtxr.sqltools`, `rust-lang.rust-analyzer`, `mhutchie.git-graph`, `Vue.volar`

Diesel:

```
diesel migration run/redo
diesel print-schema > server/src/data/mysql/schema.rs
```

Testing

First init database and generate data for testing:

```
cargo run --bin gen_testdata
```

Then start the dev server:

```
cargo run --bin test_all -- --nocapture
```

CD into `crates/passwd` and run

```
wasm-pack build --features wasm
# if wasm-pack complains about wasm-opt, you may instead execute
wasm-pack build --dev --features wasm
```

to build the `passwd` WASM package for front end.

CD into `web` and run

```
pnpm i
pnpm dev
```

to start the frondend dev server.

Formatting and Linting

```
# just give hints on linting
cargo clippy -- --allow "clippy::type_complexity"
# lint your code
cargo clippy --fix --allow-dirty --all-features -- --allow
"clippy::type_complexity"
cargo fmt
```

Add Hooks before commit

This hook helps to prevent committing to the `master` branch directly.

Editing `.git/hooks/pre-commit` as:

```
#!/bin/zsh

branch="$(git rev-parse --abbrev-ref HEAD)"

if [ "$branch" = "master" ]; then
  echo "You can't commit directly to master branch"
  exit 1
fi
```

and make it executable.

Document Generation

```
cargo doc --no-deps # generate classic rust docs
```

DevContainer Setup

After creating a dev environment using Docker Desktop, you may first edit `/etc/apt/sources.list` as

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-free

deb https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free
# deb-src https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye-security main contrib non-free

# For LLVM installation
deb http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye main
deb-src http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye main
# 17
deb http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye-17 main
deb-src http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye-17 main
# 18
deb http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye-18 main
deb-src http://apt.llvm.org/bullseye/ llvm-toolchain-bullseye-18 main
```

Then run

```
# add LLVM GPG key, or apt update will complain
```

```
wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key | sudo apt-key add -
# update source
apt update

# https://stackoverflow.com/questions/52445961/how-do-i-fix-the-rust-error-linker-cc-not-found-for-debian-on-windows-10
apt install build-essential
# install LLVM
apt-get install clang-17 lldb-17 lld-17

# setup proxy (if necessary)
export http_proxy=http://host.docker.internal:6666 # optional
export https_proxy=http://host.docker.internal:6666 # optional

# install rustup
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

# install wasm-pack
curl https://rustwasm.github.io/wasm-pack/installer/init.sh -sSf | sh

# install MySQL server and start the service
apt install mariadb-server
service mariadb start
mysql_secure_installation
# install libmysqlclient
apt-get install default-libmysqlclient-dev

# create a user for test
mariadb
> GRANT ALL ON *.* TO 'test'@'localhost' IDENTIFIED BY 'test' WITH GRANT OPTION;
> FLUSH PRIVILEGES;
> exit

# install NVM
```

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
# install Node.js v18
nvm install 18 # you may need to restart the terminal
# enable pnpm
corepack enable pnpm
pnpm setup
# this is necessary, otherwise `pnpm i` will complain
pnpm config set store-dir /root/.local/share/pnpm/store
```

Don't forget to install rust-analyzer extension into the container.