## WHAT: MVC dataflow diagram for loco saas app with postgresql DB and redis servers

WHEN: 01/01/2024 updated.
WHO: T.J. Yang, inspired by https://se-education.org/learningresources/contents/ruby/Rails.html
HOW: Diagram using Dia at http://dia-installer.de/download/index.html.en \$ LOCO\_APP\_NAME=myapp LOCO\_TEMPLATE=saas loco new -p /tmp
\$ cd /tmp/myapp/.devcontainer && docker-compose up -d
\$ cd /tmp/myapp/ && cargo test && cargo installl --path . && myapp-cli start --server-and-worker \$ cd /tmp/myapp/; tree -L 3 myapp localhost:5432 localhost:6379 myapp — Cargo.lock localhost:1025 - Cargo.toml Database cargo.tomi
config
development.yaml
production.yaml
test.yaml - examples 4 playground.rs frontend myapp/src/views/ - dist src/models/ assets index.html - auth.rs entities user.rs - favicon.ico - articles.rs – index.html — mod.rs — notes.rs – package.json – pnpm-lock.yaml prelude.rs - README.md users.rs - mod.rs  $\bigcirc$ index.css main.jsx notes.rs users.rs - vite.config.js 7 нтмі migration — Cargo.toml — README.md src — lib.rs 5 — m20220101\_000001\_users.rs — m20231103\_114510\_notes.rs — m20240101\_031322\_articles.rs – main rs - README.md myapp/src/controllers/ articles.rs — auth.rs — guide.rs - app.rs – a<sub>Pr</sub> – bin └─ main.rs -rollers mod.rs 3 - notes.rs – articles.rs - quide.rs - mod rs notes.rs 2 user.rs - fixtures [GET] /api/ health frontend notes.yaml [GET] /api/\_ping [GET] /api/articles assets index.html - lib.rs [POST] /api/auth/forgot [POST] /api/auth/login [POST] /api/auth/register [POST] /api/auth/reset – mailers — favicon.ico — index.html - auth package.json auth.rs mod.rs Router [POST] /api/auth/verify [GET] /api/guide [POST] /api/guide/echo — pnpm-lock.yaml — README.md models
articles.rs index.css main.jsx vite.config.js entities [GET] /api/notes [POST] /api/notes [GET] /api/notes/:id - mod.rs notes.rs users.rs [DELETE] /api/notes/:id [POST] /api/notes/:id [GET] /api/user/current - tasks – mod.rs seed.rs views auth.rs mod.rs 8 HTML user.rs - workers http://localhost:3000/api/user/current downloader.rs # Health checks tests curl localhost:3000/api/\_ping ; echo models — articles.rs — mod.rs curl localhost:3000/api/\_health ; echo # Hello curl localhost:3000/api/guide ; echo snapshots users.rs curl localhost:3000/api/home/hello; echo - mod.rs requests "Frost article" curl -X POST -H "Content-Type: application/json" -d '{ "title": "Your Title", "content": "Your Content xxx" } localhost:3000/api/articles — articles.rs — auth.rs – guide.rs — mod.rs — notes.rs # Users
curl localhost:3000/api/articles # get a list of artichle
# Users
curl localhost:3000/api/user/current;echo # get name of current users - prepare data.rs - snapshots - user.rs curl localhost:3000/api/user;echo # get list of users

tasks mod.rs seed.rs

25 directories, 60 files

