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 calhost:5432 localhost:6379 localhost:1025 myapp - Cargo.lock — Cargo.toml — config Database Redis mailer development.yaml – production.yaml – test.yaml 4 examples – playground.rs - frontend myapp/src/views/ – dist src/models/ assets index.html - articles.rs - mod.rs entities user.rs - articles.rs - mod.rs - favicon.ico - index.html - package.json notes.rs – pnpm-lock.yaml – README.md prelude.rs users.rs mod rs 6 index.css main.jsx users.rs 7 нтмі vite.config.js migration
Cargo.toml - README.md 5 — lib.rs - m20220101 000001 users.rs - m20231103_114510_notes.rs m20240101_031322_articles.rs - main.rs myapp/src/controllers/ README.md articles.rs auth.rs — app.rs guide.rs mod.rs – bin └─ main.rs User.al 3 - controllers
- articles.rs
- auth.rs
- guide.rs notes.rs user.rs - mod.rs — notes.rs — user.rs user#inde 2 [GET] /api/ health frontend - fixtures [GET] /api/_ping notes.yaml assets
index.html
favicon.ico [GET] /api/articles [POST] /api/auth/forgot [POST] /api/auth/login - lib.rs - mailers [POST] /api/auth/register - index.html auth auth.rs — package.json — pnpm-lock.yaml — README.md [POST] /api/auth/reset [POST] /api/auth/verify Router mod.rs routes.rs [GET] /api/guide [POST] /api/guide/echo [GET] /api/notes models articles.rs index.css main.jsx [POST] /api/notes [GET] /api/notes/:id [DELETE] /api/notes/:id — mod.rs notes.rs users.rs [POST] /api/notes/:id [GET] /api/user/current - tasks mod.rs seed.rs - views 8 HTML auth.rs mod.rs user.rs workers
downloader.rs
mod.rs http://localhost:3000/api/user/current # Health checks tests curl localhost:3000/api/_ping ; echo - models curl localhost:3000/api/_health; echo # Hello curl localhost:3000/api/guide; echo - articles.rs - mod.rs - snapshots users.rs - mod.rs curl localhost:3000/api/home/hello; echo # Post article requests curl -X POST -H "Content-Type: application/json" -d '{ articles.rs Curl -X POS1 -H "Collient-type: application/json -o ; "title": "Your Title", "content": "Your Content xxx" }' localhost:3000/api/articles curl localhost:3000/api/articles # get a list of artichle - auth.rs — guide.rs — mod.rs notes.rs # Users prepare data.rs curl localhost:3000/api/user/current;echo # get name of current users curl localhost:3000/api/user;echo # get list of users

user.rs - tasks mod.rs seed.rs 25 directories, 60 files