Репозиторий с исходным кодом: <https://github.com/AngW1SH/Registry>

Настройку SSO следует провести до первого шага, при работе непосредственно с исходным кодом, либо (для проверки корректности работы):

* Выполнить все шаги в соответствии с инструкцией
* Воспользоваться командой docker compose stop server
* Внести необходимые изменения
* Воспользоваться командой docker compose build server
* Воспользоваться командой docker compose up server

С использованием докера

1. Nginx
2. sudo apt update
3. sudo apt install nginx

Конфиг в корне репозитория (nginx.conf), поменять server\_name

2. Certbot (<https://www.nginx.com/blog/using-free-ssltls-certificates-from-lets-encrypt-with-nginx/>)

1. sudo apt install certbot python3-certbot-nginx
2. sudo certbot --nginx -d example.com -d [www.example.com](http://www.example.com)
3. Создать и заполнить файлы ./.env, ./client/.env.local, ./server/.env, ./strapi/.env, ./auth/.env по аналогии с соответствующими .env.example из репозитория
4. Собрать проект при помощи docker-compose.yaml в корне репозитория.
5. В файлах postgres/01\_structure.sql и postgres/02\_contents.sql на гугл диске находится дамп базы данных с уже заранее сгенерированными токенами и данными по некоторым проектам и пользователям.   
   Их следует скопировать в корень (туда же, где и docker-compose.yaml) в /postgres/01\_structure.sql и /postgres/02\_contents.sql.  
    После первого запуска стоит удалить эти файлы из “volumes:” в docker-compose.yaml, чтобы база данных не откатывалась при перезапуске.

В дампе уже сгенерированы API-токены (подробнее: <https://docs.strapi.io/dev-docs/configurations/api-tokens>) для обращения сервера с бизнес-логикой к strapi. Они нужны для корректной связи частей сервиса (бизнес-логики из ./server и хранилища из ./strapi). Если они не будут указаны, административная панель будет работать, но сервер не сможет делать запросы для отображения данных клиенту. В этом случае нужно будет сгенерировать их в административной панели и после этого указать в .env.

**Для упрощения запуска можно сгенерировать один токен с полными правами на CRUD всех сущностей и указать его во всех позициях:**

* В ./server/.env: PROJECTS\_TOKEN, USER\_TOKEN, REQUESTS\_TOKEN, PLATFORM\_TOKEN, FORM\_TOKEN, UPLOAD\_TOKEN, CATEGORY\_TOKEN

Используемые там (сгенерированные в предоставленном дампе базы данных) значения в **./server/.env**:

APP\_KEYS=IO9686jChg6VaUvhS0ZTsw==,3k6jPqltQyDqJ+p9jSm5/Q==,+Pct0ewHTxbvPUYIBE6K+g==,Gthq3rZXaWYrnCcUaaMjGA==

API\_TOKEN\_SALT=imOYZGE6w3d8hzJ1tIl2eQ==

ADMIN\_JWT\_SECRET=lRO8wya/IviJt+Mqhr8UHQ==

TRANSFER\_TOKEN\_SALT=YHFjg68Jb1zJfFgpJ9u4ww==

PROJECTS\_TOKEN=dfcea4fcfc6c4b40a4772700af3cdcbcc8866f4254ea12b7c7ceb6de5e1a6b1c605bd510f148025462d50f67802fb739112aa1232a2ff66fd48b5fde234101d37be45ac62afb248076ad622b38b9c2051f8dd48cc221a198f7f7a9dd6be871ab9bf90a9320ca63889f0bb2bd5ab6c44d8086b6313bbf5cfff574673f1bbd5d4b

USER\_TOKEN=fcf56c55d1def8f1069fb16bad1bad1f59d77127caaefe0b7e3ff8ae213a25b1468b4ab367b707c553ffbaa935769d85034a3a7326e834d814b87359510be7635039ce8d118e226e4cb847c8d6261cb08a1873c449c655d6f968ec921b0eb92d3c8fbebdd8785ee551c45a0b04040003b0e11f303bd736894103402c7dc90a17

REQUESTS\_TOKEN=b55644b09eb9b37ef87df16a8a8984654b1eead4d7f839e214f510b965880592498eaf33c533f734b918eb8066e65e84cbdc75594696835b865a3ffd0c81ae87bd84da0a104550da15273280b0bdd637381171014178b2fe6dd25d26bb758551a83b6160ca73e55fe64a5c2c6633aab09fcab05f2bab73a20d1e669e75e4782b

PLATFORM\_TOKEN=e994f9edd76abe30e058a7e1754a9b066ba0a9370ab6fdaa7d952f8d7e228f93eb84bb7dff63467547aa67b6380dca56001e61eaa900164370a12dc3da062aa947f829a675024dc634f57b807f543ff1f5c019ddc5526cc647ef684e41258a8a5a63815d46b5f9151b6621576908a78c3b769e6942dc3f8e251fee19da1d174c

FORM\_TOKEN=ca6e40b3f5f1ed0a477e16defe17362093c4301e7b9d86acaf1abc8143cc7160a64b5b1112fccaf6e8280c517cd7c676294f6f9a0efe49320a7ccd7963af766fcd28c4a25cf606304cbd91d095de8fa1c1e138a99367c843378bf756002f4b106294e8e7e4c507ae4d0593edd8fbb826a3376860e8af141e91445d941d1ecb9e

UPLOAD\_TOKEN=c11e60c9af45d6d33705163b917052417e925848109b84617aba867c201cc5e04c3e3cec9b4ae67e2ccb29c5d6397ecb8eeef69f6adef7a1a1147f796aa9f10699985c397cca499e2b15437976cbd7bf179b9182c58d14dfe89a794ac6adeffe99e1d514ab35f8471e19912c1f3ffad3d4900d2d5eb6ee173bcaeee88bdc3c52

CATEGORY\_TOKEN=19750355984891d34bdbf1b28b3319d6f3b718883b95a40fc73651ae399017852a42ca9aa02527cfb7da0a40646f27ed387ce6cac6d39f4be62f5ac44dbdabaf3fa09421a1ec517595f8ff5b23170245fab85acd287ecd8895c71628ede468fc6be01234801cc183f4b945c0c6d6c67432abf48240158a3e91e25adb26536050

JWT\_SECRET=FbHyQC9Mc340M/0DuTyGSw==

Без докера

Nvm

1. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh
2. curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
3. source ~/.bashrc
4. nvm install v20.4.0

Pm2

для одновременного запуска нескольких сервисов

npm i pm2 -g

Nginx

1. sudo apt update
2. sudo apt install nginx

Конфиг в корне репозитория (nginx.conf), поменять server\_name

Certbot (<https://www.nginx.com/blog/using-free-ssltls-certificates-from-lets-encrypt-with-nginx/>)

1. sudo apt install certbot python3-certbot-nginx
2. sudo certbot --nginx -d example.com -d www.example.com

Strapi

1. cd PATH\_TO\_APP/strapi
2. npm install
3. npm run build
4. pm2 start npm --name "strapi" -- start - добавляет в список pm2

Meilisearch (<https://www.meilisearch.com/docs/learn/getting_started/installation>)

1. docker pull getmeili/meilisearch:v1.4
2. docker run -it --rm -d -p 7700:7700 -e MEILI\_MASTER\_KEY='MASTER\_KEY' -v $(pwd)/meili\_data:/meili\_data getmeili/meilisearch:v1.4

* MASTER\_KEY должен соответствовать переменной среды MEILISEARCH\_API\_KEY в server/.env.
* Переменная среды MEILISEARCH\_API\_KEY в server/.env должна соответствовать формату: “https://example.com/meilisearch”

Redis (<https://redis.io/docs/install/install-redis/install-redis-on-linux/>)

1. sudo apt install lsb-release curl gpg
2. curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg
3. echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list
4. sudo apt-get update
5. sudo apt-get install redis

* Для выставления пароля отредактировать файл redis.conf, добавив строку: “requirepass examplepassword”. Переменная среды REDIS\_PASSWORD в server/.env должна соответствовать выбранному паролю.

Server

1. cd PATH\_TO\_APP/server
2. npm install - устанавливает зависимости
3. npm run build
4. pm2 start npm --name "server" -- start - добавляет в список pm2

Client

1. cd PATH\_TO\_APP/client
2. npm install - устанавливает зависимости
3. npm run build - обязательны запущенные server и strapi
4. pm2 start npm --name "client" -- start - добавляет в список pm2

Переменные среды

Создать и заполнить файлы client/.env, server/.env, strapi/.env по аналогии с соответствующими .env.example из репозитория

Чтобы остановить процессы:

pm2 stop client / pm2 stop server / pm2 stop strapi

Чтобы перезапустить процессы:

pm2 stop client / pm2 stop server / pm2 stop strapi