

Sanazoo

[GitHub repo](#)

package.json version express version sequelize version typescript version

SanaZoo is a very popular zoo !

First created in C with XML files, it is now developed with nodejs and swagger, for your eyes only

Our project

In this school project, we have to realize a complete API to manage a zoo, using Express and Sequelize as a base.

This project has been tested and integrated both on heroku, but also thanks to docker whose image is detailed below

Project Syllabus : [Syllabus.pdf](#)

Gantt chart

This project was carried out using a gantt chart :

[image-20210424163444370.png](#)

Data model used for DB

Here is our DB model used for this project :

[Planode-Zoo.png](#)

Contributions

Noé LARRIEU-LACOSTE	followers
-------------------------------------	-----------

Swann HERRERA	followers
Clément BOSSARD	followers

Information about code

Docker integration

Our docker image is built in 2 step :

- First we build all the project with dev dependencies
- Then we only keep production dependencies with compiled project

This reduce drastically the size of the image

```
## Stage 1 building the code
FROM node:lts-alpine as builder
WORKDIR /usr/app
COPY package*.json ./
RUN npm ci
COPY . .
RUN npm run build

## Stage 2 final stage with builded code
FROM node:lts-alpine
WORKDIR /usr/app
COPY package*.json ./
RUN npm ci --production

COPY --from=builder /usr/app/dist ./dist

ENV PORT=3000 \
    DB_PORT=3306 \
    DB_DRIVER='mysql' \
    DB_HOST='localhost' \
    DB_NAME='zoo' \
    DB_USER='root' \
    DB_PASSWORD=''
```

```
CMD node dist/src/index.js
```

Env

Environment variable	Default	Description
PORT	3000	Express listen port
DB_DRIVER	mysql	Driver for sql connection for sequelize
DB_HOST	localhost	Host domain / IP for DB
DB_NAME	zoo	DB Schema name
DB_USER	zoo	DB user
DB_PASSWORD	empty	DB password

Main dependencies

Dependency	Version	Description
Express	express version	Web API Framework
Date FNS	date-fns version	Useful libraries to manipulates dates
Dotenv	date fns version	Used to load <code>.env</code> file
Argon2	argon2 version	Used to encrypt users password
Mysql2	mysql version	DB driver
Sequelize	sequelize version	Orm libraries to bind class to DB entities
Swagger-jsdoc	version	Used to implements swagger page
Swagger-ui-express	version	Used to implements swagger page
Yup	yup version	Form validation library used to validate data in post body of our requests
Typescript	typescript version	Very useful to use types in JS based framework

API Endpoints

Postman Environment

You can check our endpoints with postman directly on this URL:

<https://documenter.getpostman.com/view/11568150/TzjvdwNA>

API Description :

- [Affluence](#)
 - [Daily](#)
 - [Daily by enclosure](#)
 - [Live enclosure affluence](#)
 - [Monthly](#)
 - [Monthly by enclosure](#)
 - [Total](#)
 - [Total by enclosure](#)
 - [Weekly](#)
 - [Weekly by enclosure](#)
 - [1Yearly](#)
 - [1Yearly by enclosure](#)
- [Animal](#)
 - [Create](#)
 - [Delete](#)
 - [Get all](#)
 - [Get by id](#)
 - [Move Enclosure](#)
 - [Update](#)
- [Animal Health Book](#)
 - [Create entry](#)
 - [Delete](#)
 - [Get All](#)
 - [Get All By Animal](#)
 - [Get One](#)
 - [Update](#)
- [Enclosure](#)
 - [Add One](#)
 - [Delete One](#)
 - [Edit One](#)

- [Get All](#)
- [Get All Animals In Enclosure](#)
- [Get All By Type](#)
- [Get One](#)
- [Enclosure Images](#)
 - [Add](#)
 - [Delete One](#)
 - [Edit One](#)
 - [Gell All From Enclosure](#)
 - [Get All](#)
 - [Get One](#)
- [Enclosure Service-book](#)
 - [Create service-book](#)
 - [Delete service-book](#)
 - [Edit service-book](#)
 - [Gell All From Employee](#)
 - [Get All](#)
 - [Get All From Enclosure](#)
 - [Get One](#)
- [Enclosure Type](#)
 - [Create](#)
 - [Delete](#)
 - [Get All](#)
 - [Get One](#)
 - [Update](#)
- [Entry](#)
 - [Add entry](#)
 - [Get all](#)
 - [Get by enclosure](#)
 - [Get by pass](#)
 - [Get by user](#)
 - [Remove entry](#)
- [Maintenance](#)
 - [Get All by State](#)

- [Get Best Month](#)
- [Update Maintenance State](#)
- [Pass](#)
 - [Add enclosure access](#)
 - [Create](#)
 - [Delete pass](#)
 - [Get all](#)
 - [Get all by user id](#)
 - [Get by id](#)
 - [Remove enclosure access](#)
 - [Update pass](#)
- [Pass Night](#)
 - [Add availability](#)
 - [Delete passnight](#)
 - [Get all](#)
 - [Get all valid](#)
 - [Update passnight](#)
- [Pass Type](#)
 - [Add pass to pass type](#)
 - [Create](#)
 - [Delete](#)
 - [Get all](#)
 - [Get by id](#)
 - [Update](#)
- [Planning](#)
 - [Add](#)
 - [Delete](#)
 - [Get All](#)
 - [Get Calendar](#)
 - [Get One](#)
 - [Get Open Date](#)
 - [Update](#)
- [Specie](#)
 - [Create](#)

- [Delete](#)
- [Get All](#)
- [Get By ID](#)
- [Update](#)
- [Statistics](#)
 - [Count all pass](#)
 - [Count all pass by types](#)
 - [Count animal](#)
 - [Count animal by enclosure](#)
 - [Count enclosure](#)
 - [Count expired pass](#)
 - [Count user](#)
 - [Count valid pass](#)
 - [Count valid pass by types](#)
- [User](#)
 - [Change Password](#)
 - [Create](#)
 - [Delete](#)
 - [Force Delete](#)
 - [Get All](#)
 - [Get By ID](#)
 - [Login](#)
 - [Logout](#)
 - [ME](#)
 - [Register](#)
 - [Restaure User](#)
 - [Update by admin](#)
 - [Update client only by employee](#)
- [User Role](#)
 - [Affect User](#)
 - [Create](#)
 - [Delete](#)
 - [Get All](#)
 - [Get by ID](#)

- [Update](#)
 - [Swagger](#)
-

Swagger

This project contain a complete swagger test environment to use API, you can access it on `https://domain.example/swagger`

It look FABULOUS :

[image-20210424170307083.png](#)

Revision #2

Created 2022-05-09 20:57:38 UTC by Noé Larrieu-Lacoste

Updated 2022-05-09 21:06:56 UTC by Noé Larrieu-Lacoste