Technical Documentation for Clac des Doigts (English Version)

Product Name: Rest API - Chicken using Node.js , Express server, and MySql
Product version: node.js version v16.17.1, MySql phpmyadmin v 5.2.0
Code Editors used: Visual Studio code and PhpMyadmin

Packages Required for the Node.js Backend:

- ☐ Bcrypt
- ☐ Body-parser
- ☐ Dotenv
- ☐ Express
- ☐ Jsonwebtoken
- ☐ mysql2

Dev dependencies : nodemon
Testing: Postman.com
Date: 18/06/2023
Author: Jayani Gunasekara.

Product Overview:
This product is designed for the company Clac des Doigts as a test project for the interview.

Product Objective:
The objective of the API is to receive requests and responses from the Chicken table in the farm database. The product is also extended to receive the data from the Cage table in the farm database.
The product is already developed as a public API. But it can be upgraded to private API with a key authentication for user login if it is required.

Product Testing on Local Machine:

1. Download the **farm.sql** database file on GitHub and import it into your local MySql server.
2. Get the database credentials and replace them in **config.database.js**(Username, password, port) in the source code.
3. Run **npm install** on your code editor to install the recommended packages and dev dependencies to run the backend.

Procedure:

- Create the Database using MySQL

Database name: **farm**
Table names: ***chicken , cages, sections***

- Entity Relationship Diagram

First I have defined the relationships of the tables. There is a relation between chicken table and cage

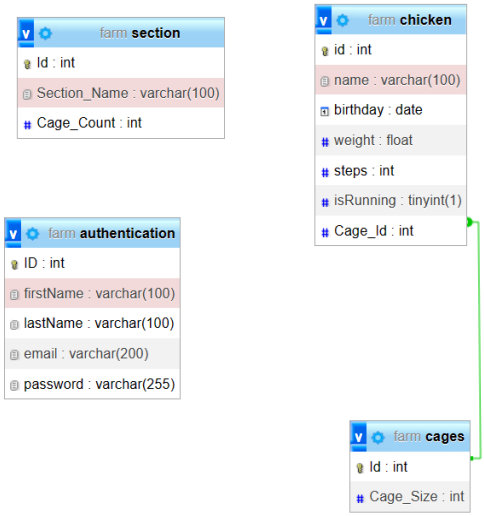

table as defined in this ERP diagram

Routing:

Create chickens

http://localhost:5000/webservice/chicken/create

Req: POST

Ex: Body -> row , type :json


```
{
"name": "chickenTest15",
"birthday": "2022-08-09",
"weight": 2.3,
"steps": 1,
"isRunning": 0,
"Cage_Id": 2
}
```


Get chickens

http://localhost:5000/webservice/chicken

Req: GET

Res:


```
{
  "success": 1,
  "data": [
    {
      "id": 12,
      "name": "chickenTest3",
      "birthday": "2023-05-01T22:00:00.000Z",
      "weight": 1.5,
```

```
      "steps": 0,
      "isRunning": 1,
      "Cage_Id": 5
    },
    {
      "id": 14,
      "name": "chickenTest14",
      "birthday": "2022-03-31T22:00:00.000Z",
      "weight": 2.3,
      "steps": 2,
      "isRunning": 0,
      "Cage_Id": 2
    },
}
```


Get chicken by ID

http://localhost:5000/webservice/chicken/11

Req: GET{ID}

Res:


```
{
  "success": 1,
  "data": {
    "id": 15,
    "name": "chickenTest15",
    "birthday": "2022-08-08T22:00:00.000Z",
    "weight": 2.3,
    "steps": 1,
    "isRunning": 0,
    "Cage_Id": 2
  }
}
```


Update chicken

http://localhost:5000/webservice/chicken/update

Req: PATCH

Ex:Body -> row , type :json


```
{"id": 12,
"name": "chickenTest12",
"birthday": "2023-05-01T22:00:00.000Z",
"weight": 1.5,
"steps": 0,
"isRunning": 1,
"Cage_Id": 5}
```


Delete Chicken

http://localhost:5000/webservice/chicken/delete

Req: DELETE

Ex: Body -> row , type :json

```
{
"id":15
}
```

Update chicken steps

http://localhost:5000/webservice/chicken/run

Req: PATCH

Ex: Body -> row , type :json

```
{
"id":16
}
```

Get the Number of chickens in a cage by cage id

http://localhost:5000/webservice/chicken/chickenCount/2

Req: GET {Cage_id}

Res:

```
{
  "success": 1,
  "data": [
    {
      "Count": 2
    }
  ]
}
```

Product Release:

Github:

https://github.com/jayanigunasekara/Rest-Api-using-node.js-Express

Product Future development:

Can be implemented with user login to restrict unauthorized access by generating a token to access the chicken API.