Documentation technique pour Clac des Doigts (version Français)

Nom du produit: Rest API - Poulet utilisant Node.js , Express server, et MySql
Version du produit: node.js version v16.17.1, MySql phpmyadmin v 5.2.0
Code Éditeurs utilisés: Visual Studio code et PhpMyadmin

Paquets requis pour le backend Node.js :

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

Dev dependencies : nodemon
Test: Postman.com
Date: 18/06/2023
Auteur: Jayani Gunasekara.

Description du produit :
Ce produit a été conçu pour la société Clac des Doigts en tant que projet test pour l'entretien.

Objectif du produit :
L'objectif de l'API est de recevoir des requêtes et des réponses de la table "chicken" de la base de données de farm. Le produit est également étendu pour recevoir les données de la table Cage dans la base de données de farm.
Le produit est déjà développé en tant qu'API publique. Mais il peut être transformé en API privée avec une clé d'authentification pour la connexion de l'utilisateur si cela est nécessaire.

Test du produit sur une machine locale :

1. Téléchargez le fichier de base de données **farm.sql** sur GitHub et importez-le sur votre serveur MySql local.
2. Obtenez les informations d'identification de la base de données et remplacez-les dans **config.database.js**(Username, password, port) dans le code source.
3. Exécutez **npm install** sur votre éditeur de code pour installer les paquets recommandés et les dépendances de développement pour exécuter le backend.

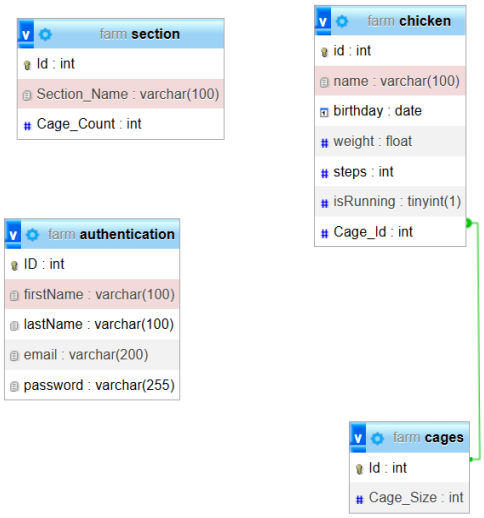
Procédure :

- Créer la base de données à l'aide de MySQL
  Nom de la base de données : ***farm***

- Noms des tables: ***chicken , cages, sections***

- Diagramme des relations entre entités

J'ai d'abord défini les relations entre les tables. Il existe une relation entre la table chcken et la table cage
comme défini dans ce diagramme ERP



Routing:

Créer des poulets
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
}
```


Obtenir des poulets
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
    },
}
```


Obtenir le poulet par 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
  }
}
```


Mise à jour du poulet

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}
```

Supprimer le poulet
http://localhost:5000/webservice/chicken/delete
Req: DELETE
Ex: Body -> row , type :json

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

Mise à jour des steps
http://localhost:5000/webservice/chicken/run
Req: PATCH
Ex: Body -> row , type :json

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

Obtenir le nombre de poulets dans une cage par identifiant de cage
http://localhost:5000/webservice/chicken/chickenCount/2
Req: GET {Cage_id}
Res:

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

Release du produit :
Github

Produit Développement futur :
Peut être mis en œuvre avec la connexion de l'utilisateur pour restreindre l'accès non autorisé en générant un jeton pour accéder à l'API chicken.