

Déploiement:

L'environnement doit être muni d'un serveur SQL ainsi que d'un environnement Node.JS.

Mise en place de la base de donnée:

Pour installer le serveur SQL sur linux, utiliser la commande suivante:

```
$ sudo apt-get install mysql
```

Puis suivre les étapes d'installation jusqu'à la configuration des identifiants de connexion. Une fois SQL installé, charger la base de donnée se trouvant dans le fichier ***init_db.sql*** avec la commande :

```
$ sudo mysql -pPassword < ./init_db.sql
```

La base de données contient la table des "chickens" ainsi que celle **des fermes**. Ainsi chaque chicken est affilié à une ferme avec une clef secondaire.

Chargement de Node:

Exécuter la commande suivante pour charger les dépenses du projet :

```
$ npm install
```

Note:

Ne pas oublier de changer le mot de passe de connexion à la base de donnée, *ligne 11*.

Pour lancer le serveur, exécuter :

```
$node ./index.js
```

Utilisation:

1) Lecture: (GET)

Méthode	URI	Paramètre Body	Description
GET	/chicken	None	Renvoie tous les chickens
GET	/chicken/:id	None	Renvoie un chicken par son ID
GET	/chicken/name/:v	None	Renvoie un chicken par son ID

2) Ecriture: (POST)

Note:

- Le format des paramètres JSON est spécifié dans le fichier exemple ***“test_request.json”***
- La date doit être sous le format américain.

Méthode	URI	Paramètre Body	Description
POST	/chicken	JSON: voire “test_request.json”	Ajoute un chicken comme spécifié dans le JSON

3) Modification: (PUT)

Note:

- Le format des paramètres JSON est spécifié dans le fichier exemple ***“test_modif.json”***
- La date doit être sous le format américain.

Méthode	URI	Paramètre Body	Description
PUT	/chicken/:id	JSON: voire “test_modif.json”	modifie un chicken par un ID comme spécifié dans le JSON

4) Suppression: (DELETE)

Méthode	URI	Paramètre Body	Description
DELETE	/chicken/:id	None	Supprime un chicken par son ID

5) Run: (PATCH)

Méthode	URI	Paramètre Body	Description
PATCH	/chicken/run/:id	None	Met IS_RUNNING à True, et incrémente STEPS

Format de la réponse:

Sous format JSON, celle-ci possède deux/trois champs :

- “**error**” contenant un boolean (false si tout c’est bien passé)
- “**data**” contenant les données demandées
- “**message**” contenant un message décrivant si l’action c’est bien dérouler ou non