

# ECF2023\_Fredj\_Corentin

Ceci est le dépôt pour l'ECF Graduate Developer

## Présentation

Ceci est le rendu ECF pour évaluer les compétences.

***Je vous demande s'il vous plaît d'être indulgent ! Veuillez m'excusez par avance .. seulement j'ai codé une application entière (après avoir posé des questions sur le forum de STUDI), et 72H avant la date limite de rendu j'ai remarqué qu'il y avait un annexe avec des consignes précises.. Je ne suis visiblement pas le pingouin le plus glissant de la banquise .. J'ai bien conscience d'être l'unique responsable de ma situation. Je souhaitais tout de même la partager avec vous esperant de la compréhension.***

---

**Voici le git du premier projet que j'ai codé si vous souhaitez vérifier : <https://github.com/DroxKiwi/webappBeWave>**

**J'ai tout de même décidé de faire l'ECF en suivant les consignes (72h c'est court pour rattraper la boulette) ceci explique le manque de commit pour le début de ce projet, ayant utilisé une partie du projet précédent pour monter celui ci. Ceci étant dit ! Merci de prendre le temps d'évaluer.**

---

## Mettre en place le backoffice en local

### PostgreSQL

**Il faut nécessairement PostgreSQL et NPM sur votre machine !**

**Le projet a été codé dans un environnement LINUX !**

### Cloner le projet en SSH depuis git :

```
git clone git@github.com:DroxKiwi/ECF2023_fredj_corentin.git
```

### Cloner depuis HTTPS :

```
https://github.com/DroxKiwi/ECF2023_fredj_corentin.git
```

### Cloner depuis GitHub CLI :

```
gh repo clone DroxKiwi/ECF2023_fredj_corentin
```

### Télécharger les dépendances une fois le projet cloné :

```
cd /chemin_du_projet/studiECF2023_Fredj_Corentin
npm install
```

### Initialiser la base de données en exécutant /bin/init\_database.sh :

Le fichier bash **/bin/init\_database.sh** permet d'initialiser la BDD en local, il est important de vérifier les informations données à PostgreSQL **avant de lancer le script !** Celles ci ne sont pas liées au fichier d'environnement et se trouve dans le script en question !

(...)

```
// Verify information about database HERE !!!!
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'database_dev_studiec',
  password: 'psqlpsw',
})
```

```
// Read the SQL file
(...)
```

Le fichier ***/bin/init\_database.sh*** lancera à son tour le fichier de fixture : ***./Models/init\_models.js***

De même il faut vérifier les valeur données à postgresQL pour que le script fonctionne !

```
(...)
```

```
// Verify information about database HERE !!!!
const pool = new Pool({
  user: 'postgres',
  host: 'localhost',
  database: 'database_dev_studiecf',
  password: 'psqlpsw',
})
```

```
function fixtureLoad(){
  (...)
```

Une fois les informations postgresQL vérifiées et correctes en fonction de votre installation, vous pouvez lancer le script d'initialisation de la BDD

```
cd studiECF2023_Fredj_Corentin/bin
./init_database.sh
```

Le script bash va donc redémarrer postgres (**nécessite l'accord sudo**), créer la base de données ***database\_dev\_studiecf*** et appliquer les modèles et les fixtures dans la base de données.

Il existe 3 fichiers qui exécute l'application :

- **index.js**
- **secureSSL.js**
- **app.js**

***index.js*** est appelé si l'on choisit de lancer l'application en environnement de développement.

***secureSSL.js*** est appelé si l'on choisit de lancer l'application en environnement de production, j'ai initialement codé ce script pour faire tourner l'application avec un serveur **Apache2** sur un droplet **DigitalOcean**.

***app.js*** a été créé pour que **fly.io** puisse connecter l'application à leur serveur.

## Lancer l'application en local :

```
cd ..
npm run dev
```

## Vous connecter et essayer l'application :

A partir de là, l'application est lancée et vous pouvez vous connecter au compte admin injecter en BDD à l'adresse : <http://localhost:3000/>

identifiant : **[admin@admin.com](#)**

mot de passe : **admin**