

Procédure d'installation Équipe 2 - Partie Combats

1. Préambule

Vous trouverez dans ce document la procédure pour installer la partie combats du projet Korrigan.

2. Outils utilisés

- Un compte Gitlab
- PHP 7.2 ou plus : `php -v`
- Docker : `docker -v`
- Node JS : `node -v`
- NPM : `npm -v`
- React
- Slim
- Une base de donnée MySQL

3. Clonage

Pour cloner les données du projet, il faut lancer la commande **git clone** avec le lien suivant : <https://gitlab.com/glucchini/projet-2-korrigans> ou alors en ssh avec le lien suivant : `git@gitlab.com:glucchini/projet-2-korrigans.git`.

4. Préparation environnement back-end

Pour bien préparer votre environnement back-end il faut dupliquer les différents fichiers .skel et leur changer les extensions. Voici les fichiers à modifier :

- **Code/Back/env.skel** à modifier en .env
 - Fichier dans lequel se trouve les variables de la base de données tel que l'utilisateur, le mot de passe et le port de communication.
- **Code/Back/php-tweak.skel** à modifier en php-tweak.ini
 - Pour le réglage de l'affichage des erreurs
- **Code/Back/data/www/includes/params.skel** à modifier en params.php
 - Pour le réglage des accès à la base de données ainsi que d'autres settings de l'application

Ces fichiers convertis doivent être consignés dans le fichier .gitignore pour que chaque membre du groupe puisse posséder son propre environnement.

5. Préparation environnement front-end

Afin de ne pas alourdir le dépôt Git par la suite, il faut indiquer les différents fichiers et dossiers à ignorer dans un fichier .gitignore présent dans le dossier **Code/Front/front-korrigans** ou à la racine du projet.

Le Dockerfile présent dans le dossier permet d'effectuer différentes tâches lors de son exécution :

- Il utilise l'image officielle de NodeJS pour les tâches suivantes,
- Il définit le dossier /app comme dossier de travail,
- Il ajoute le chemin /app/node_modules/.bin à la variable \$PATH,
- Il installe les dépendances nécessaires dans les package.json et package-lock.json,
- Il exécute une installation npm dans le but d'installer react localement,
- Il finit par lancer le serveur à l'aide de npm start, qui devient accessible à l'adresse du serveur et son port déclaré dans le docker-compose.

6. Démarrage

- Se rendre dans le dossier /Code/Back et lancer la commande docker-compose up -d
- Se rendre dans le dossier /Code/Front/front-korrigans et lancer la commande docker-compose up -d --build

7. Accès au services

- PHP My Admin : <https://korrigans-team2-db.lpweb-lannion.fr>
- Webservices : <https://korrigans-team2-ws.lpweb-lannion.fr>
- Front : <https://korrigans-team2.lpweb-lannion.fr>

8. Utilisation des routes de l'API

Se référer au document suivant : ./Gestions de projet/API/API DOCUMENTATION.pdf

9. Arrêter les services

Pour arrêter les services :

- Se rendre dans le dossier /Code/Back et lancer la commande docker-compose down. Vous pouvez supprimer les dossiers /Code/Back/data/mysql et /Code/Back/data/www/logs et /Code/Back/data/www/slim afin de faire une installation fraîche.
- Se rendre dans le dossier /Code/Front/front-korrigans et lancer la commande docker-compose down. Pour faire une installation fraîche, vous pouvez supprimer le dossier node_modules et relancer un docker-compose up -d --build.

10. Exemple de fichier GIT Ignore

```
Code/Back/docker-compose.yml
Code/Back/data/mysql
Code/Back/data/www/slim
Code/Back/files/php-tweak.ini
Code/Back/.env
Code/Back/data/www/includes/params.php
Code/Back/data/www/logs
```

```
Code/Front/front-korrigans/node_modules
Code/Front/front-korrigans/.pnp
Code/Front/front-korrigans/.pnp.js
Code/Front/front-korrigans/coverage
Code/Front/front-korrigans/build
```