

# Projet Web Serveur (Pokemon World)

May 21, 2023

Gaillor Jowardo JINORO  
Timothée FONTENILLE

## 1 Les fonctionnalités demandées

- Inscription et connexion des utilisateurs
- Attribution d'un pokémon aléatoire lors de l'inscription
- Distribution d'un nouveau pokémon chaque jour à la première connexion de la journée
- Echange de pokémons entre joueurs
- Montée en niveau des pokémons
- Affichage de la liste de tous les pokémons avec leurs caractéristiques et niveaux
- Affichage de la liste des pokémons possédés par chaque joueur avec leurs caractéristiques et niveaux
- Gestion des erreurs et des exceptions

## 2 Les principales classes de l'application

- User
- Pokemon
- Trade
- LevelUp
- Connexion

### 3 Les tables de la base de données

- **Users** (id, firstname, lastname, username, password, email)
- **Pokemons** (table\_id, id, name, level, owner\_id)
- **Trades** (id, pokemon1\_id, pokemon2\_id, user1\_id, user2\_id)
- **LevelUps** (id, pokemon\_id, user\_id)
- **Connexions** (user\_id, first\_connexion, last\_connexion)

### 4 Les jointures

- Users.id = Pokemons.owner\_id
- Users.id = Connexions.user\_id
- Trades.pokemon1\_id = Pokemons.id
- Trades.pokemon2\_id = Pokemons.id
- Trades.user1\_id = Users.id
- Trades.user2\_id = Users.id
- LevelUps.pokemon\_id = Pokemons.id
- LevelUps.user\_id = Users.id

### 5 Les URIs

- **GET /** : affiche la page d'accueil du site
- **GET /register** : affiche le formulaire d'inscription
- **POST /register** : enregistre un nouvel utilisateur
- **GET /connexion** : affiche le formulaire de connexion
- **POST /connexion** : connecte un utilisateur existant
- **GET /logout** : déconnecte l'utilisateur courant
- **GET /profile/:id** : affiche le tableau de bord de l'utilisateur courant
- **GET /profile/:id/trade** : enregistre l'échange de pokémon de l'utilisateur
- **GET /trades/:id** : affiche les joueurs disponibles avec leurs pokémons pour échanger
- **POST /trades/:id** : enregistre l'échange réalisé
- **POST /profile/:id/levelup** : enregistre l'augmentation de niveau choisi par l'utilisateur ayant comme identifiant ":id"

## 6 Remarques

Dans ce projet, nous tenons à préciser que certaines fonctionnalités n'ont pas pu être gérées. Voici la liste de celles-ci :

- **Gestion de demande d'amis**
- **Attente d'une confirmation avant qu'un échange ait lieu**

A défaut, nous avons donné la possibilité à un utilisateur de faire un échange de pokémon avec n'importe quel utilisateur. En effet, une fois que l'utilisateur a choisi le pokémon et celui avec qui il souhaite échanger, l'échange est tout de suite fait sans attente de confirmation de l'autre.

## 7 Fonctionnement du projet

- **Etape 1:** Après avoir démarré le projet avec la commande `./gradlew run` dans le terminal, tapez directement dans la barre d'adresse du navigateur `http://localhost:8081/register` (Accéder à la page d'inscription) ou `http://localhost:8081/connexion` (Accéder à la page de connexion).
- **Remarque:** Pour garder un bon fonctionnement du site, nous tenons à ce qu'un utilisateur soit d'abord inscrit ou connecté pour qu'il puisse naviguer dans les URIs. A contrario, des messages s'afficheront pour lui communiquer cela.
- **Etape 2:** Une fois connecté, vous pouvez naviguer dans le site en utilisant les URIs spécifiés ci-dessus.

## 8 Diagramme UML

Voici le diagramme UML qui décrit au mieux les relations qui existent entre les classes principales du projet.

