

## Guide de maintenance : App Pokemon

Cette application de gestion de Pokémons a été implémentée avec le Framework Angular ( v 9.0.5). Nous utilisons firebase de Google pour héberger l'application (hosting) et sa base de données (real time database).

L'application est disponible à la page suivante : <https://pokedex-mt.web.app/>

Un lien vers le repository github : <https://github.com/ThomasCaillet79/ng-pokemons-app-master> (repo privé, demander l'accès).

Aucune sécurité n'a été implémentée côté Back.

Commandes clés :

- Installation des dépendances : **npm install**
- Lancement de l'application : **npm run start**

### 1. Les rôles

#### 1) Admin

Un seul compte existant :

- Identifiant : pikachu
- Mot de passe : pikachu2

Peut ajouter, modifier, supprimer un Pokémon à la liste de l'ensemble des Pokémon de l'application.

#### 2) Non admin

Chaque utilisateur peut créer ou utiliser un compte existant. Il peut tirer au sort un Pokémon à ajouter à son pokédex.

### 2. Composants

Nom	Description
<b>App</b>	Parent de l'application
<b>Login</b>	Connexion / Inscription
<b>PageNotFound</b>	Page unique pour les routes inexistantes
<b>Loader</b>	Temps de chargement
<b>SearchPokemon</b>	Recherche des Pokémon de son pokédex
<b>PokemonForm</b>	Formulaire d'un Pokémon (Nom, Img, etc...)
<b>ListPokemon</b>	Ensemble des Pokémon présent dans son pokédex (tous les Pokémon si admin)
<b>EditPokemon</b>	Edition d'un Pokémon
<b>AddPokemon</b>	Ajout d'un Pokémon

### 3. Module

Nom	Description
<b>App</b>	Parent de l'application
<b>AppRouting</b>	Routes globales
<b>PokemonsRouting</b>	Routes gestion Pokémons

### 4. Service

Nom	Description
<b>InMemoryData</b>	Intercepte les requêtes de l'application (Dev.)
<b>Auth</b>	Authentification Requêtage de la base de données (/users).
<b>Pokemons</b>	Gestion des pokemons Requêtage de la base de données (/pokemons).

### 5. Directives et Pipes

Nous avons utilisé plusieurs directives et pipes natifs d'Angular.

Nous avons également créé notre propre directive : BorderCard, qui permet de modifier la bordure de la carte Pokémon au passage de la souris.

Et aussi un pipe d'affichage en fonction du type de Pokémon.

### 6. Pokemon

Exemple de Pokemon :

```
{
  "cp": 6,
  "created": "Fri Mar 11 2022 11:15:05 GMT+0100 (heure normale d'Europe centrale)",
  "hp": 28,
  "id": 1,
  "name": "Salamèche",
  "picture": "https://assets.pokemon.com/assets/cms2/img/pokedex/detail/004.png",
  "types": [
    "Feu"
  ]
}
```

## 7. User

```
{
  "admin": false,
  "listePoke": [
    3,
    0
  ],
  "password": "bidule"
}
```

Un utilisateur admin ne possède pas d'attribut listePoke. Cet attribut est un tableau contenant les ids des pokemons de l'utilisateur. Le mot de passe de l'utilisateur est inscrit directement dans la base de données. Un cryptage de ce mdp est à envisager dans le cas d'une « vraie » application.

## 8. Requête à la BDD : Firebase

Notre base de données ne contient aucune sécurité et est accessible par n'importe qui.

URL DB : <https://pokemon-beb38-default-rtdb.europe-west1.firebaseio.com>

### 1. Pokemon

Requête	URI	Description	Response
GET	/pokemons.json	Retourne l'ensemble des pokémons présent dans la BDD.	Pokemon[]
GET	/pokemons/x.json	Renvoie le pokemon possédant l'id x.	Pokemon
PUT	/pokemons/x.json	Modifie le pokemon possédant l'id x.	Pokemon
DELETE	/pokemons/x.json	Supprime le pokemon possédant l'id x.	Pokemon

Pour ajouter un nouveau pokemon à la base de donnée, nous avons opté pour un PUT /pokemons/x.json avec un id x non présent dans la base de données (incrémenté de 1).

### 2. User

Requête	URI	Description	Response
GET	/users/bidule.json	Retourne le user bidule.	User
GET	/users.json	Renvoie la liste des utilisateurs.	User[]
PUT	/users/bidule.json	Modifie le user bidule.	User
DELETE	/users/bidule.json	Supprime le user bidule.	User

Si l'on souhaite ajouter le pokemon possédant l'id 5 à l'utilisateur bidule, il suffit de faire un update à /users/bidule/listePoke/y.json avec 5 en body et y étant le nombre de Pokémons possédés par l'utilisateur.