

# PROJET

## Virtualisation & Conteneurisation Application Pokemon

BERT Thomas, CHERIFI Chahinez, DUFLAUT Quentin, MEJAHED Salim, PORCHETTE Ralph



docker

esiea

INGÉNIEURS D'UN NUMÉRIQUE UTILE

## Table des matières :

<b>1) Introduction :</b> .....	<b>3</b>
Lien vers le GITHUB :.....	3
<b>2) Test du fonctionnement de l'application :</b> .....	<b>3</b>
A) Ajout d'un pokémon.....	3
Base de données mysql :.....	5
B) Supprimer d'un Pokémon.....	6
Aperçu du Frontend :.....	6
Backend :.....	6
Base de données :.....	7
<b>3) Conclusion.....</b>	<b>7</b>

## 1)Introduction :

Dans le cadre de ce projet, nous avons développé une application web codée en python, dédiée aux passionnés de Pokémon. Cette application permettra aux utilisateurs d'ajouter, de consulter et de supprimer des Pokémon via une interface simple et intuitive.

L'application a été dockerisée afin de faciliter son déploiement et d'assurer une exécution cohérente sur différentes plateformes. Docker permet de créer des conteneurs isolés qui contiennent toutes les dépendances nécessaires à l'application, ce qui simplifie le processus de configuration et évite les conflits entre environnements. Cela permet également de garantir que l'application fonctionne de manière identique sur tous les systèmes où Docker est installé. Dans ce contexte, nous utilisons Docker Compose pour orchestrer les différents services nécessaires à l'application (par exemple, le serveur, la base de données, etc.).

**NB : afin d'accéder à notre base de données SQL, le mot de passe est : myrootpassword**

Lien vers le GITHUB :

[https://github.com/PrKen/Virtu\\_Final\\_Project](https://github.com/PrKen/Virtu_Final_Project)

## 2)Test du fonctionnement de l'application :

### A)Ajout d'un pokémon

Nous avons donc développé notre application et l'avons associé à notre base de données.

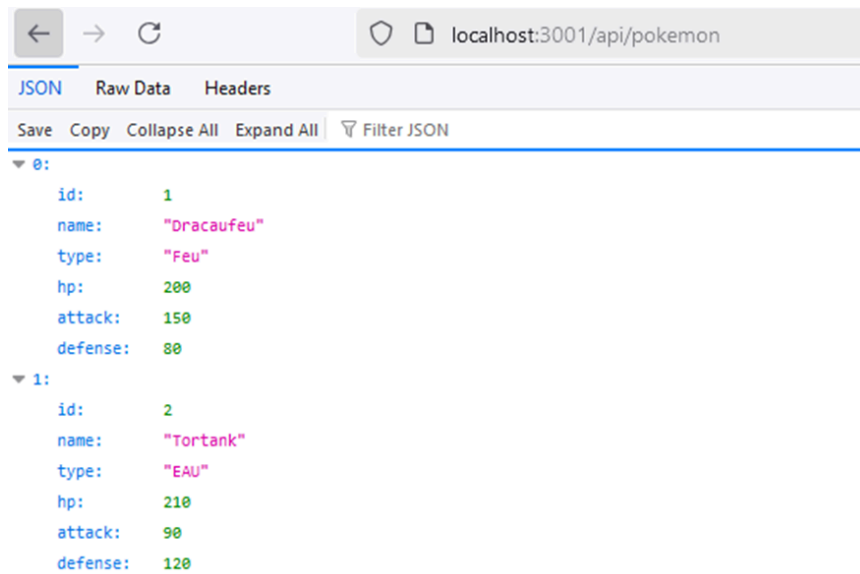
Voici, un aperçu de l'application, où nous avons ajouté ici, un pokémon ainsi que ses informations correspondantes.

Name	Type	HP	Attack	Defense	Actions
Dracaufeu	Feu	200	150	80	Supprimer

En parallèle, le backend nous permet de vérifier le bon enregistrement des informations de chaque pokémon. En effet, ici nous pouvons voir les requêtes faites (GET et POST -> ajout d'un pokémon):

```
backend-1 | Fetched Pokémon: []
backend-1 | GET /api/pokemon 200 34.578 ms - 2
backend-1 | Fetched Pokémon: []
backend-1 | GET /api/pokemon 200 1.502 ms - 2
backend-1 | GET /favicon.ico 404 2.270 ms - 150
backend-1 | Adding Pokémon: { name: 'Dracaufeu', type: 'Feu', hp: 200, attack: 150, defense: 80 }
backend-1 | POST /api/pokemon 201 11.436 ms - 41
backend-1 | Fetched Pokémon: [
backend-1 |   {
backend-1 |     id: 1,
backend-1 |     name: 'Dracaufeu',
backend-1 |     type: 'Feu',
backend-1 |     hp: 200,
backend-1 |     attack: 150,
backend-1 |     defense: 80
backend-1 |   }
backend-1 | ]
backend-1 | GET /api/pokemon 200 2.806 ms - 77
backend-1 | Adding Pokémon: { name: 'Tortank', type: 'EAU', hp: 210, attack: 90, defense: 120 }
backend-1 | POST /api/pokemon 201 4.473 ms - 41
backend-1 | Fetched Pokémon: [
backend-1 |   {
backend-1 |     id: 1,
backend-1 |     name: 'Dracaufeu',
backend-1 |     type: 'Feu',
backend-1 |     hp: 200,
backend-1 |     attack: 150,
backend-1 |     defense: 80
backend-1 |   },
backend-1 |   {
backend-1 |     id: 2,
backend-1 |     name: 'Tortank',
backend-1 |     type: 'EAU',
backend-1 |     hp: 210,
backend-1 |     attack: 90,
backend-1 |     defense: 120
backend-1 |   }
backend-1 | ]
backend-1 | GET /api/pokemon 200 2.449 ms - 151
backend-1 | GET /favicon.ico 404 0.568 ms - 150
```

Comme notre backend le montre ci-dessous, chaque pokémon créé est enregistré avec un ID, un nom, un type et ses informations de combat ( hp, attack, et défense). Ces derniers sont ajoutés au fur et à mesure.



Base de donnés mysql :

Afin de tester le bon fonctionnement de notre base MySQL, dans une autre invite de commande, nous avons exécuté les commandes suivantes, avec le bon nom du conteneur :

***docker exec -it pokemon\_app-mysql-1 mysql -u root -p***

```

mysql> use pokemon_db ;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from pokemon ;
+----+-----+-----+-----+-----+-----+
| id | name   | type | hp  | attack | defense |
+----+-----+-----+-----+-----+-----+
| 1  | Dracaufeu | Feu  | 200 | 150    | 80      |
| 2  | Tortank  | EAU  | 210 | 90     | 120     |
+----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> |

```

Suite à cela, nous pouvons voir que les pokémons ont été ajoutés dans mysql.

## B) Supprimer d'un Pokémon

Maintenant, nous avons essayé de supprimer un pokémon de notre application afin de tester le bon fonctionnement de l'application.

### Aperçu du Frontend :

Nous avons, via l'application web supprimé, un pokémon, et afin de s'assurer que la suppression a été réalisée avec succès, nous avons vérifiés que les données n'existent plus dans la base données et le backend.

Pokémon Manager

**Add a New Pokémon**

Add Pokémon

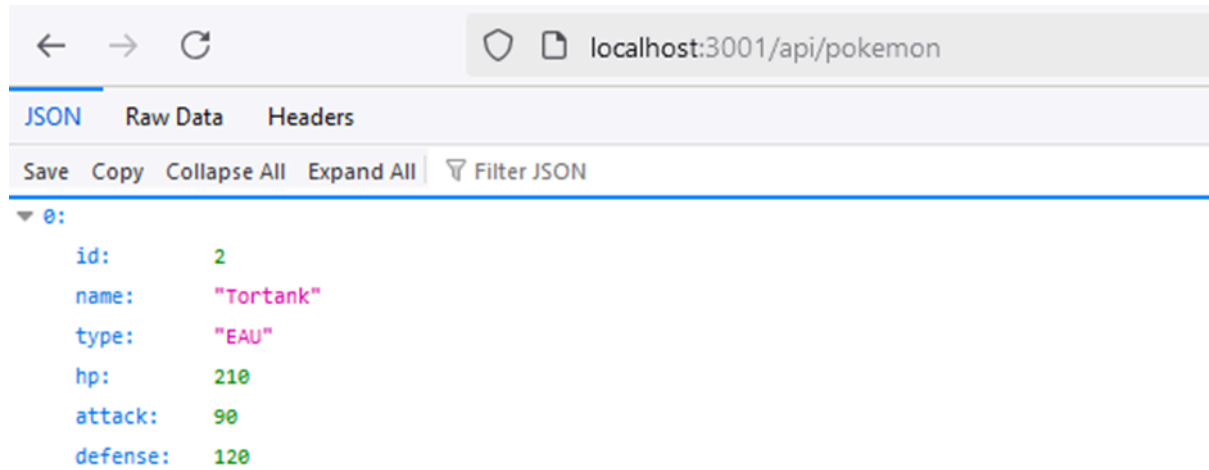
Pokémon List

Name	Type	HP	Attack	Defense	Actions
Tortank	EAU	210	90	120	<span style="background-color: red; color: white; padding: 2px 5px;">Delete</span>

### Backend :

Nous pouvons confirmer que la requête est bien passée.

```
backend-1 | DELETE /api/pokemon/1 204 6.782 ms - -
```



Base de données :

```

mysql> select * from pokemon ;
+----+-----+-----+-----+-----+-----+
| id | name   | type | hp  | attack | defense |
+----+-----+-----+-----+-----+-----+
| 2  | Tortank | EAU  | 210 | 90     | 120     |
+----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

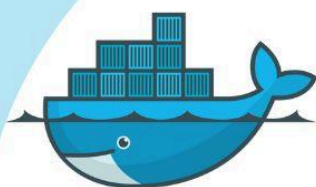
En vérifiant la base de données nous constatons que l'id '1' a bien été supprimée.

### 3) Conclusion

En conclusion, ce projet a permis de développer une application web fonctionnelle et interactive dédiée à la gestion de Pokémon, combinant un frontend et un backend. Grâce à la dockerisation, nous avons assuré une gestion simplifiée des environnements.

Les tests effectués sur les différentes fonctionnalités, telles que l'ajout, la consultation et la suppression de Pokémon, ont confirmé le bon fonctionnement de l'application et son intégration avec la base de données MySQL.





docker

esiea

INGÉNIEUR E.S. & UN NUMÉRIQUE UTILE