

## TD 5 - TGC PokeApp

---

### Objectif

L'objectif de ce TD est de créer une application e-commerce de cartes Pokémon. L'application permettra de consulter les cartes disponibles, de les ajouter au panier et de passer commande.

### Étape 1 : Initialiser le Projet

**En tant que développeur, je veux initialiser un nouveau projet Vue.js pour créer une application web qui permet de consulter et d'acheter des cartes Pokémon.**

Vous devrez configurer **vue-router** et **Pinia**. Pour **vue-router**, vous pouvez vous référer au TD précédent. Pour **Pinia**, suivez la documentation officielle : [Pinia Documentation](#). N'oubliez pas de travailler en utilisant l'Option API.

### Étape 2 : Afficher la liste des cartes

**En tant qu'utilisateur, je veux voir la liste des Pokémon disponibles pour pouvoir les consulter.**

Sur la page d'accueil, l'utilisateur doit pouvoir voir la liste des Pokémon disponibles. Chaque carte doit afficher son nom, son image et son prix. Vous utiliserez l'API [PokéAPI](#) pour récupérer les données des Pokémon. Utilisez la propriété **base\_experience** pour définir le prix de chaque carte.

Pour appeler l'API, vous pouvez vous baser sur le **httpClient** du TD précédent.

### Sans filet

Je vais maintenant jouer le rôle d'un client et vous demander de nouvelles fonctionnalités à implémenter :

- **En tant qu'utilisateur, je veux pouvoir rechercher un Pokémon par son nom.**
  - **En tant qu'utilisateur, je veux voir les Pokémon sous forme de jolies cartes.**
  - **En tant qu'utilisateur, je veux voir les détails d'un Pokémon sur une page dédiée.**
  - **En tant qu'utilisateur, je veux voir un panier vide dans le header de l'application.**
  - **En tant qu'utilisateur, je veux pouvoir ajouter un Pokémon au panier.**
  - **En tant qu'utilisateur, je veux voir le nombre de Pokémon dans mon panier.**
  - **En tant qu'utilisateur, je veux pouvoir augmenter ou diminuer la quantité d'un Pokémon dans mon panier.**
  - **En tant qu'utilisateur, je veux voir le prix total de mon panier.**
  - **En tant qu'utilisateur, je veux voir mon panier sur une page dédiée.**
  - **En tant qu'utilisateur, je veux pouvoir supprimer un Pokémon de mon panier.**
  - **En tant qu'utilisateur, je veux voir dans la liste des Pokémon si un Pokémon est déjà dans mon panier.**
-

**La commande est simulée.** Il n'y a pas de backend pour gérer les commandes. Vous pouvez simplement afficher un message de confirmation après que la commande a été passée.

- En tant qu'utilisateur, je veux pouvoir passer commande si mon panier est valide.
  - En tant qu'utilisateur, je veux pouvoir passer commande avec une page de confirmation.
  - En tant qu'utilisateur, je veux voir un message de confirmation après avoir passé commande.
  - En tant qu'utilisateur, je veux voir que mon panier est vide après avoir passé commande.
- 

- En tant qu'utilisateur, je veux pouvoir filtrer les Pokémon par type.
- En tant qu'utilisateur, je veux pouvoir retrouver mes commandes passées.

## Contraintes

- L'application doit être responsive.
  - Votre application doit être esthétique. Vous pouvez utiliser des librairies de composants comme **Vuetify** ou **TailwindCSS**. (Vuetify vous permettra de gagner du temps pour les cartes, par exemple).
  - Vous devez gérer les différentes pages de l'application avec **vue-router**. Une page '/', 'pokemon/:id', 'cart' et 'confirmation' sont obligatoires.
  - Le code doit être découpé en composants.
  - Votre panier doit être stocké dans **Pinia**.
  - Vous devez utiliser au moins une **mixin** (par exemple, pour valider le panier).
  - Vous devez utiliser au moins une **directive personnalisée** (par exemple, pour afficher dynamiquement si un Pokémon est dans le panier).
- 

Bon courage !