

TP2 – Création d’une application avec Vite

Module R3-01 – Vue JS

Edition : 16/12/2023

Objectifs du TP :

1. Comprendre les concepts fondamentaux de VueJS 3 (Composants, props, appels API).
2. Utiliser le framework Vite pour le développement rapide.
3. Intégrer et consommer des données provenant de l'API Pokebuild.
4. Créer des composants réutilisables pour structurer l'application.

Prérequis :

- Connaissance de base en HTML, CSS, et JavaScript.
- Connaissance des propriétés et des directives en VueJS (voir TP1).
- Node.js et npm installés sur la machine

1. Installation

Créer votre projet Vue en utilisant Vite, nous l’appellerons “tp2-pokemon-vuejs”, profitez-en également pour installer la librairie Axios, nous nous en servirons plus tard.

2. Création de notre composant pokemon.

- Ce composant sera affiché sous forme de card.
- Le format des données doit respecter cette structure et toutes les informations devront être présentes :

```
{
  "name": "Bulbizarre",
  "image": "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/1.png",
  "apiTypes": [
    {
      "name": "Plante",
      "image": "https://static.wikia.nocookie.net/pokemongo/images/c/c5/Grass.png"
    }
  ],
  "apiGeneration": 1
}
```

- La couleur d’arrière-plan de la card devra être affichée en fonction du premier type du pokémon affiché dans le tableau.
- La totalité des types contenus dans le tableau devra être affiché.

3. Création de notre composant pokedex :

Notre pokedex ne sera ni plus ni moins que notre liste de cards pokemon, pour l’instant dupliquez votre composant créé auparavant dans ce composant et vérifiez que vos deux cards s’affichent correctement. Pour l’instant tout n’est que statique.

4. Lister les pokémons de manière dynamique avec les props

Dans le composant pokedex, créez dans vos data un tableau de pokemons, ce tableau reprendra la structure que vous avez implémenté au-dessus. Faites ensuite une boucle v-for sur ce tableau pour afficher vos composants pokemon. N'oubliez pas de passer votre pokemon courant en props et d'afficher les informations de celui-ci via le composant pokemon.

5. Utilisation de l'API Pokebuild.

Pour afficher les informations de nos pokemons sans avoir à en écrire manuellement 500, nous allons utiliser une API permettant de nous retourner les informations au format JSON de nos différents pokémons. Si vous avez bien respecté la structure mise en place précédemment, l'intégration se fera sans aucun souci. N'oubliez pas que vous pouvez utiliser axios.

Voici l'api à utiliser :

<https://pokebuildapi.fr/api/v1/pokemon/limit/20>

Note : nous avons volontairement limité à 20 le nombre de retour pour ne pas avoir un temps de chargement trop conséquent.

6. Affichage du nombre de pokemons par type

Créer un composant permettant de voir le nombre de pokemons présents par type sur les résultats retournés par l'API. (Feuille : 3, Eau : 2, Feu: 3...) n'hésitez pas à utiliser les images des types mises à disposition par l'API.

7. Ajout d'un nouveau pokémon

Créer un champ input qui vous permettra de pouvoir insérer un nombre entre 1 et 850.

Ce numéro est le numéro de pokedex du pokémon sélectionné. Pour récupérer un pokémon par son id, vous pouvez utiliser cette API :

<https://pokebuildapi.fr/api/v1/pokemon/850> avec 850 qui correspond à l'identifiant pokedex du pokemon.

Une fois le pokemon récupéré, l'ajouter dans le tableau. Si un pokémon est déjà existant, alors il faudra l'ignorer. Ce pokemon devra s'afficher correctement dans l'ordre du pokedex.