

Rapport Pokedex - PokeApi

Défis Techniques

Intégration de l'API Pokémon : L'une des principales difficultés auxquelles j'ai été confronté était l'intégration de l'API Pokémon. Il a fallu comprendre comment effectuer des appels à cette API pour obtenir des informations sur les Pokémon, comme leurs noms, numéros et types. La gestion des données renvoyées par l'API a également été un point d'attention.

Gestion de l'état de l'application : Pour que l'application fonctionne correctement, il était essentiel de gérer son état. Cela signifiait suivre les Pokémon ajoutés à l'équipe de l'utilisateur, ainsi que la mémorisation de ces données entre les sessions. Cela a nécessité la manipulation de la mémoire locale pour stocker ces informations.

Recherche en temps réel : La mise en œuvre de la fonction de recherche en temps réel a été un autre défi. Cela signifiait que les utilisateurs pouvaient trouver des Pokémon en fonction de leur nom à mesure qu'ils saisissaient leur recherche. Cela a impliqué de gérer les délais pour éviter des appels inutiles à l'API.

Réutilisation de composants React : Un autre défi a été de s'assurer que les composants React que j'ai créés étaient suffisamment modulaires et réutilisables. Cela permettrait leur utilisation dans différentes parties de l'application sans modifications majeures.

Solutions Apportées

Pour résoudre ces défis, j'ai utilisé des concepts et des outils appris pendant mon apprentissage. J'ai utilisé des appels asynchrones avec des promesses JavaScript pour intégrer l'API Pokémon. J'ai également utilisé le hook `useState` de React pour gérer l'état de l'application et stocker les données des équipes de Pokémon dans la mémoire locale.

La fonction de recherche en temps réel a été mise en œuvre en utilisant des délais pour optimiser les performances. Enfin, j'ai conçu les composants React de manière modulaire, ce qui a permis de les réutiliser dans différentes parties de l'application sans problème.