Bienvenu dans le

Challenge API Pokemon

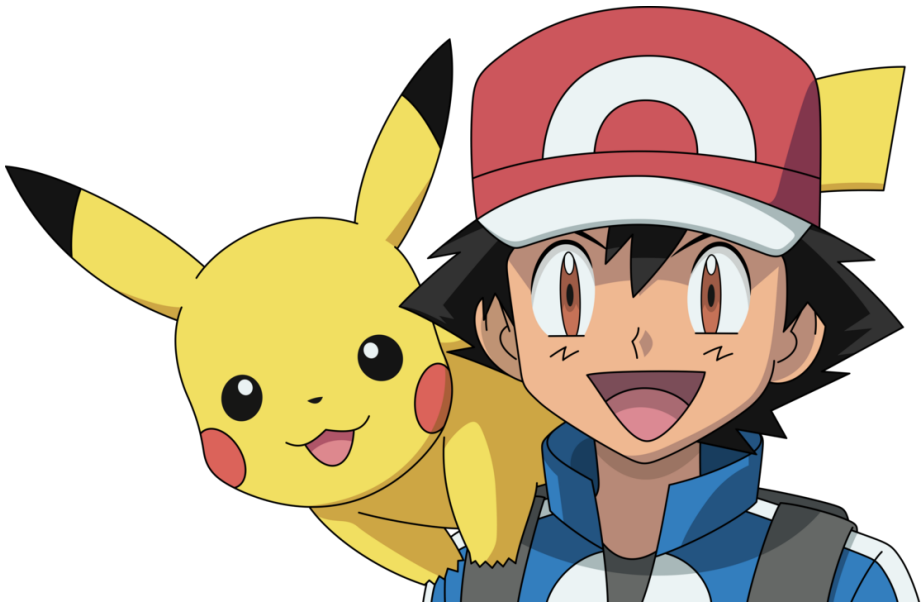
Énoncé :

L'**objectif** de cet exercice est de créer **une mini API en JavaScript avec Node.js** pour manipuler une liste de cartes Pokémon **stockées dans un fichier JSON appelé "pokemonList.json"**. Cette API permettra **d'ajouter, de modifier, de supprimer et de rechercher des cartes Pokémon**. Une interface utilisateur sera également développée pour communiquer directement avec cette API.

Niveaux: Jaune => Bonus facultatif

Exemple de User Stories possibles:

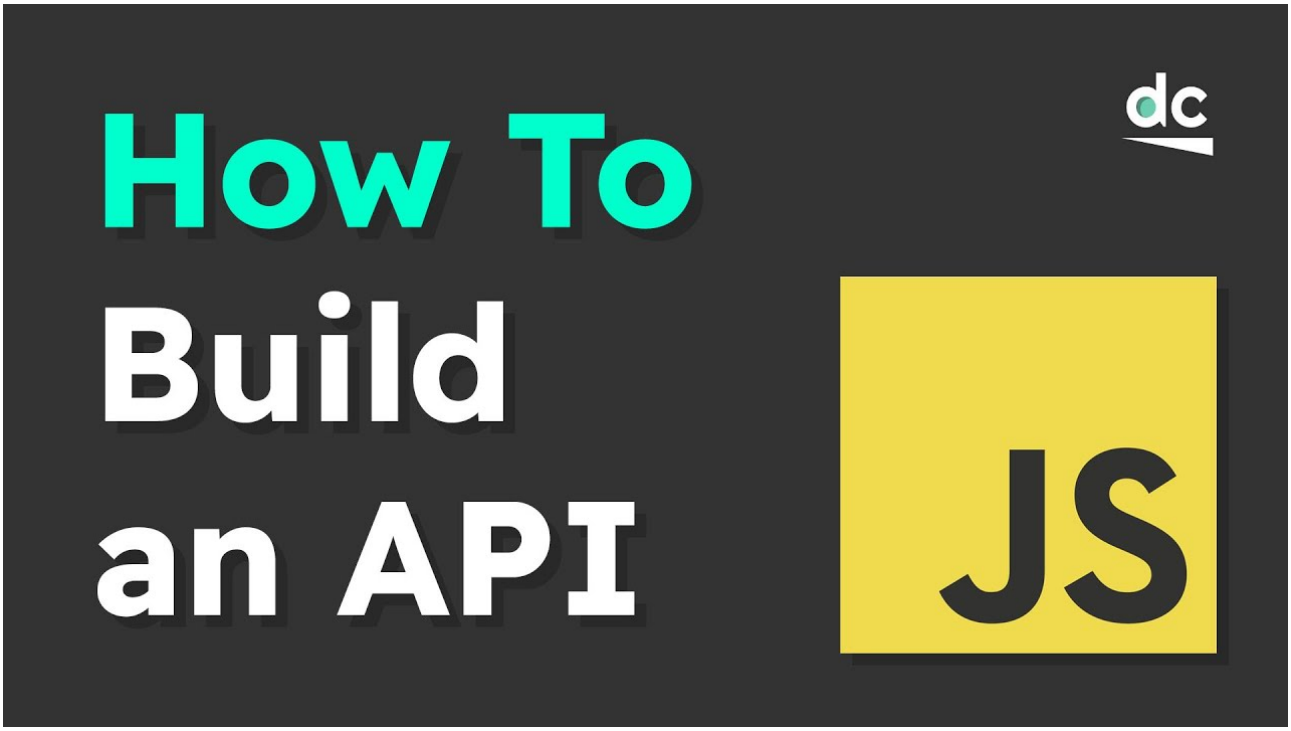
1. En tant qu'utilisateur, je veux pouvoir afficher la liste de toutes les cartes Pokémon disponibles dans pokemonList.json sur la page d'accueil (index.html).
2. En tant qu'utilisateur, je veux pouvoir éditer une carte Pokémon en cliquant sur le bouton "éditer" à côté de chaque carte. Le formulaire d'édition doit s'afficher avec les informations de la carte sélectionnée pré-remplies.
3. En tant qu'utilisateur, je veux pouvoir supprimer une carte Pokémon en cliquant sur le bouton "supprimer" à côté de chaque carte.
4. En tant qu'utilisateur, je veux pouvoir rechercher une carte Pokémon par nom en utilisant le formulaire de recherche de la page d'accueil. La vue de la carte Pokémon correspondante doit s'afficher.
5. En tant qu'utilisateur, je veux pouvoir ajouter une nouvelle carte Pokémon en utilisant le formulaire d'ajout disponible sur la page d'accueil. Après l'ajout, la liste de toutes les cartes doit être mise à jour avec la nouvelle carte ajoutée.
6. En tant qu'utilisateur, je veux recevoir des messages d'erreur clairs en cas d'erreurs de saisie ou de problèmes de connexion à l'API.
7. En tant que développeur, je veux que l'API soit sécurisée en utilisant des méthodes d'authentification pour les requêtes de modification et de suppression.
8. En tant que développeur, je veux que l'API soit bien documentée avec des instructions d'utilisation et des exemples de code pour faciliter la maintenance et le développement futur.



Quelques Pistes (futurs recherches google, demandes au chat et autre…) :

- Pour manipuler les fichiers JSON en JavaScript, vous pouvez utiliser **les fonctions readFile et writeFile du module fs** de Node.js.
- Pour créer une API RESTful, vous pouvez utiliser le module **Express.js** de Node.js. **Il facilite la création de routes et la manipulation des requêtes et des réponses HTTP.**
- Pour créer une interface utilisateur, vous pouvez utiliser des bibliothèques de frontend telles que React, Vue.js ou Angular. Cependant, si vous débutez, **vous pouvez commencer par créer simplement une page HTML/CSS/JS simple sans utiliser de bibliothèques ou de frameworks**.
- Pour stocker les données de manière sécurisée, vous pouvez utiliser une base de données, mais **pour cet exercice, vous pouvez simplement stocker les données dans un fichier JSON.**
- **Pour gérer les formulaires HTML et les requêtes HTTP POST, vous pouvez utiliser le module body-parser d'Express.js.**
- Pour gérer les erreurs et les messages d'erreur, vous pouvez **utiliser les fonctions try-catch** pour capturer les erreurs et les renvoyer en tant que réponse JSON.

Tuto d’aide:



Site de l'article:

https://dcode.domenade.com/tutorials/how-to-build-a-rest-api-with-nodejs-and-express

Autre Tuto:

la structure du projet:

- **index.js** : le point d'entrée de votre application Node.js. C'est ici que vous allez définir les routes de votre API, traiter les requêtes et les réponses, et manipuler les données stockées dans **pokemonList.json.**
- **package.json** : un fichier de configuration de Node.js qui définit les dépendances de votre application, entre autres.
- **pokemonList.json** : un fichier JSON qui stocke les données des cartes Pokémon.
- **public/** : un dossier qui contient les fichiers statiques de votre interface utilisateur.
- **index.html** : le fichier HTML qui contient la structure de votre page web.
- **style.css** : un fichier CSS qui contient les règles de style pour votre page web.
- **main.js** : **un fichier JavaScript qui contient la logique de votre interface utilisateur, comme la manipulation des données de formulaire, la récupération des données à partir de l'API, et la mise à jour de l'interface utilisateur en conséquence.**

Cette structure vous est fournie pré-remplie ici: **https://github.com/PierreDenaes/apiPokeChallenge.git**

Une fois le dossier cloné, ouvrez le dans vs code et afficher un terminal pour envoyer la commande « npm install »

La commande pour démarrer votre serveur est « node start » une fois démarrer vous aurez accès votre api sur l'url « localhost:3000 »

Une liste d'url d'image de pokemon pouvant vous servir de jeux d'essais (data de test) est présente dans le README.md du projet.

Postman est une application permettant de tester des API, **on télécharge on install : https://www.postman.com/**

Ce projet peut être effectué seul ou en binôme dans un délais maximum de 10 jours pour le réaliser (soir et week-end compris lol…).

Le Livrable :

Vous avez la possibilité de 3 niveaux de réalisation dans la conception de cette Api, seul le premier niveau est obligatoire sur le délais a parti.

Tout doit être envoyé sur GitHub, Gitlab… ce que vous voulez mais versionné.

1 niveaux (obligatoire)

Créations de l'api sans interface front. Les différentes routes de l'api sont utilisables avec Postman. L'api renvoie un flux de donnée au format Json, les données de l'api sont modifiable ou effaçable.

2 niveaux (facultatif)

Créations d'une interface front en Html, Css, Js pour communiquer avec l'api.

3 niveaux (bonus)

Sécurisation de l'api avec méthode d'authentification
Création de la documentation.

Quel(le) Challenger serez vous ?