

Test Technique Frontend React : Affichage d'Adresses

Objectif

Ce test technique a pour objectif d'évaluer votre maîtrise des bases du développement frontend moderne, notamment React, TypeScript ou JavaScript, ainsi que votre capacité à interagir avec une API externe, gérer l'état d'une application et afficher des données de manière structurée.

Cahier des Charges

Vous êtes libre de choisir les outils et technologies que vous souhaitez (ex. : Vite, Create React App, Next.js, etc.), ainsi que le langage (TypeScript ou JavaScript).

1. Initialisation du Projet

Vous êtes libre d'utiliser les outils que vous souhaitez (ex: Create React App, Vite, Next.js, etc.) et de choisir entre **TypeScript** ou **JavaScript**.

2. Récupération des Données (Fetch)

- L'application doit effectuer une requête HTTP GET vers l'API publique française de géocodage (Base Adresse Nationale).
- Documentation :
<https://www.data.gouv.fr/dataservices/api-adresse-base-adresse-nationale-ba>
n
- Endpoint : `https://api-adresse.data.gouv.fr/search/?q=votre_recherche`
- L'application doit permettre à l'utilisateur d'effectuer une recherche (par exemple à partir d'une ville ou d'un code postal, comme "Bordeaux" ou "33000").
- Une gestion simple des erreurs de l'API ou des problèmes réseau est attendue.

3. Affichage des Données

Les résultats retournés par l'API doivent être affichés sous forme de liste dans l'interface utilisateur.

Chaque élément doit afficher les informations principales de l'adresse (par exemple : adresse complète, ville, code postal).

L'affichage doit être réactif et mis à jour dynamiquement à l'aide de React.

4. Environnement et Outils

Vous avez une liberté totale quant au choix des outils de développement. Veuillez simplement lister les technologies utilisées dans votre solution.

Catégorie	Outils/Technologies Utilisés
Framework/Librairie	React
Langage	TypeScript ou JavaScript
Bundler/CLI	[Ex: Vite, Next.js, CRA]
Gestion d'état	[Ex: Hooks locaux, Redux, Zustand, etc.]
CSS/Styling	[Ex: CSS Modules, Tailwind, Styled Components]

Critères d'Évaluation

Les points suivants seront examinés lors de l'évaluation :

- Fonctionnalité** : l'application fonctionne-t-elle comme attendu (requête API et affichage des résultats) ?
- Qualité du code** : lisibilité, clarté, respect des bonnes pratiques et conventions.
- Gestion des états** : prise en compte des états de chargement, d'erreur et de données.
- Structure du projet** : organisation des fichiers et des composants.
- Bonus** : hébergement de l'application (facultatif, non pénalisant).

Soumission

Merci de nous transmettre votre projet sous la forme d'un dépôt Git (GitHub, GitLab ou autre) par email à l'adresse suivante : duncan.legoff@kapptivate.com

Bon courage !