

Test Technique React/Typescript :

- **Temps :**

1h

- **Rendu :**

Un dépôt GitHub / Bitbucket accessible publiquement, ou une archive envoyée par e-mail. *Note :* React \geq 17 et TypeScript sont imposés.

L'heure de la réception de l'e-mail contenant l'archive / donnant l'accès au repo GIT, une fois l'exercice terminé, fera foi.

- **Objectif :**

Vérifier les compétences techniques requises pour travailler quotidiennement sur Organilog.

- **Idée générale:**

À l'exception de la version de React et de l'utilisation de TypeScript, tout, dans ce mini-exercice, peut être interprété et réalisé différemment de ce qui a été demandé.

Nous ne recherchons pas des « machines à coder ». Nous cherchons des personnes suffisamment compétentes pour réaliser l'exercice et capables de motiver et d'expliquer leurs choix, quitte à faire légèrement évoluer la ligne directrice, sans pour autant chercher à l'imposer.

- **Ressource DOC utile :**

<https://dummyjson.com/docs>

<https://dummyjson.com/docs/products>

<https://dummyjson.com/docs/products/#limit>

<https://dummyjson.com/docs/products#category>

• Exercice :

• Etape 1 :

Utiliser l'API GET /products de <https://dummyjson.com/products> pour récupérer les 5 premiers produits et les sauvegarder localement dans l'application.

Une fois les données chargées, les appels à l'API ne doivent pas être systématiquement refaits lors du rechargement de l'application dans le navigateur.

L'affichage sera comme suit :

- Une vue liste simple de ces produits, affichant le titre et la description, qui constituera la page d'accueil de l'application.
- Au clic sur un élément de la liste, une vue « lecture » s'ouvre et affiche :
 - Titre
 - Description
 - Prix
 - Note
 - Stock
 - Marque
 - L'image thumbnail principale du produit
 - Toutes les autres images du produit fournies par l'API dans le champ « images », affichées sous forme de mini-galerie (aucune interaction n'est requise ; un simple affichage dans la vue est suffisant).

• Etape 2 :

Ajouter un bouton sur la page d'accueil permettant de charger les 5 produits suivants et de les intégrer au fonctionnement de l'étape 1.

Au même titre que pour l'étape 1, les éléments récupérés ne devront plus nécessiter de nouveaux appels à l'API pour accéder aux informations déjà reçues.

• Etape 3 (BONUS) :

Ajouter un bouton sur la vue détaillée d'un produit, nommé « Voir catégorie », qui, au clic, ira récupérer les informations via l'API

GET <https://dummyjson.com/products/category/NOM-DE-LA-CATEGORIE-PRODUIT>

et les affichera sous forme de liste (une simple liste, sans fonctionnalité supplémentaire) avec le titre et la description des produits de la catégorie.

Pour ce bonus, il n'est pas obligatoire d'éviter les appels systématiques à l'API lors du clic sur le bouton.

• Note :

L'intégration est un plus, mais ce n'est pas l'élément évalué. L'essentiel est d'obtenir quelque chose de fonctionnel.

Aucune limitation n'est imposée quant à l'utilisation de bibliothèques : vous êtes libre d'utiliser ce qui vous semble pertinent pour atteindre le résultat souhaité. Seuls **React** ≥ 17 et

TypeScript sont imposés.