

Tp Promise

Application de Suivi de Livraison Asynchrone

Votre objectif est de créer une application en JavaScript qui simule le suivi d'un colis depuis l'entrepôt jusqu'au client final. Le suivi passe par plusieurs étapes de livraison, avec des délais simulés, et vous devrez gérer plusieurs opérations asynchrones.

1. Initialisation du colis

- Créez une fonction asynchrone `initialiserColis` qui simule l'ajout d'un colis dans l'entrepôt.
- Cette fonction retourne un objet contenant les informations du colis (par exemple, un identifiant unique et un nom de destinataire).

2. Simuler les étapes de livraison

- Créez trois fonctions asynchrones :
 - `expedition` : Simule le départ du colis de l'entrepôt vers un centre de tri régional.
 - `transfertRegional` : Simule le déplacement du colis du centre de tri vers un centre de distribution local.
 - `livraisonFinale` : Simule la livraison du colis au client final.
- Chaque fonction doit accepter un objet de colis en paramètre et attendre un délai simulé avant de renvoyer un message indiquant la réussite de l'étape.

3. Gestion des erreurs

- Dans chaque étape de livraison, introduisez une chance d'échec (par exemple, une probabilité de 20 % que l'étape échoue).
- En cas d'échec, la fonction doit lancer une erreur avec un message approprié (par exemple, "Échec lors de l'expédition").

4. Processus global de livraison

- Créez une fonction asynchrone `suivreColis` qui gère l'ensemble du processus de livraison en appelant `initialiserColis`, puis les fonctions `expedition`, `transfertRegional`, et `livraisonFinale` en chaîne, en utilisant `await` pour chaque étape.

- Utilisez `try...catch` pour capturer et gérer les erreurs à chaque étape. Si une étape échoue, relancez automatiquement cette étape jusqu'à ce qu'elle réussisse (en limitant le nombre de tentatives à 3 pour éviter des boucles infinies).

5. Interface utilisateur en console

- Ajoutez des `console.log` dans chaque étape pour afficher des messages d'état : par exemple, "Le colis est en cours d'expédition...", "Transfert régional réussi", etc.

- Affichez un message final indiquant le succès ou l'échec global de la livraison.

- Créez une version qui permet de suivre plusieurs colis en parallèle (utilisez `Promise.all` pour gérer l'exécution simultanée).

- Visualiser la progression : Si possible, ajoutez une barre de progression en console pour chaque étape.