

Test technique Morpheus Branchement

❖ SUJET

L'équipe Morpheus a signé des contrats avec deux clients.

Le premier nous donne un fichier JSON d'annonces immo que nous devons importer sur leboncoin. Les libellés des champs sont plutôt explicites dans le fichier. Cependant, certains champs ne sont pas nécessaires pour leboncoin.fr.

Le second est un fichier XML d'annonces emploi d'une agence d'Intérim. On nous demande de concaténer les champs ***description_poste*** et ***description_entreprise*** pour pouvoir afficher toutes ces informations dans le corps de l'annonce.

Ces champs comportent des balises HTML mais Leboncoin n'accepte pas le HTML. Il faut donc nettoyer ces champs tout en gardant un rendu compréhensible (garder les sauts de ligne et les listes).

Avec le projet Symfony fourni, traitez et transformez les deux fichiers d'annonces pour qu'ils puissent être acceptés par l'API documentée ci-dessous.

Des tests unitaires sont nécessaires concernant les Hooks et leurs fonctions.

❖ BONUS

Le code postal n'est pas présent dans les données XML mais on aimerait tout de même l'avoir. Il vous est demandé de proposer une solution technique pour récupérer le code postal mais l'implémentation est optionnelle.

❖ RENDU

Vous devez rendre un projet Symfony fonctionnel qui transforme des annonces pour un envoi vers l'API leboncoin. Vous pouvez aussi rendre un court rapport de quelques lignes décrivant certains problèmes à résoudre ou des pistes d'amélioration.

Les données des clients ne sont pas forcément de bonne qualité ou à l'inverse intègrent des informations dont l'API leboncoin n'a pas besoin. Cependant, pour améliorer la qualité des annonces importées, vous pouvez proposer des solutions pour intégrer davantage de données. Vous pouvez implémenter ces solutions pour prouver le concept proposé mais tout ceci est optionnel.

Une attention sera apportée à la qualité du code mais faites au plus simple avec le temps que vous avez.

API leboncoin

Documentation

Libellé	Type	Min	Max	Requis	Description
id	string	1	100	oui	Référence de l'annonce.
title	string	1	100	oui	Titre de l'annonce.
body	string	1	500	oui	Description de l'annonce.
vertical	enum (string)	1	100	oui	Catégorie générale de l'annonce.
price	int	1	-	non	Prix de l'objet vendu.
city	string	1	100	oui	Ville où est vendu l'objet.
zip_code	string	5	5	non	Code postal de la ville.
pro_ad	boolean	1	1	oui	Vrai si l'annonce est postée par un professionnel.
images	array (string)	1	10	non	Tableau contenant les liens vers les images de l'annonce.

Enum vertical

- Immobilier
- Vacance
- Emploi
- Formation professionnelle
- Auto
- Moto
- Autre

Vertical Immobilier

La verticale Immo possède plusieurs champs spécifiques :

Libellé	Type	Min	Max	Requis	Description
category	enum (int)	1	1	oui	Catégorie: - 1 (Vente) - 2 (Location) - 3 (Colocation) - 4 (Bureaux et commerces)
type	enum (string)	0	1	non*	*Requis si catégorie = 4 Type de vente: - 1 : Vente (catégorie 1 ou 4) - 2 : Location (catégorie 2 ou 3) - 3 : Viager (catégorie 1)

Vertical Emploi

La verticale Emploi possède plusieurs champs spécifiques :

<i>Libellé</i>	<i>Type</i>	<i>Min</i>	<i>Max</i>	<i>Requis</i>	<i>Description</i>
contract	enum (int)	1	1	oui	Contrat: - 1 : CDI - 2 : CDD - 3 : Interim - 4 : Stage/Alternance
salary	int	0	-	non	Le salaire proposé.