Conception & Développement d'Applications et Services Web/Serveur

TD8 : CMS Headless, wrapper HTML basé sur des templates

L'objectif du TD est de construire le catalogue LBS à l'aide d'un CMS "Headless" et de construire un interface "server side" pour consulter ce catalogue fonctionnant comme un wrapper HTML traduisant les données JSON fournies par l'api du CMS en pages HTML.

1. Mise en place du service catalogue

Le service catalogue est réalisé par le CMS Headless "Directus" (https://directus.io/) (tutoriel : https://grafikart.fr/tutoriels/directus-nodejs-1972).
Dans la composition docker, le service est réalisé par 2 conteneurs : un conteneur pour DIrectus lui-même, et un conteneur pour sa base de données.

Un extrait du docker-compose.yml nécessaire vous est fourni. Il faut l'ajouter à votre configuration existante. Avec cette configuration, l'outil sera utilisable depuis votre machine hôte :

- pour la partie administration du CMS : http://localhost:19055/
- pour l'api : http://localhost:19055/items/

Une fois le service démarré (attention, le 1er démarrage peut prendre un peu de temps), vous devrez vous connecter au panneau d'administration (documentation : https://docs.directus.io/configuration/data-model/) pour créer :

- une collection `categories` : id (int AI), libelle (string), description (texte)
- une collection `sandwiches` : id (int AI), libelle (string), description (texte), prix (decimal)
- une association 1-N `categories –> sandwiches` (attention, utiliser le mode avancé pour créer l'association inverse)

Créer ensuite quatre catégories (classique, veggie, world, chaud), et huit sandwiches répartis dans ces différentes catégories.
Examiner les données renvoyées par l'api à l'aide de postman. Exécuter l'ensemble de requêtes permettant d'obtenir les sandwichs de la catégorie "classique".
Essayer d'utiliser les paramètres globaux de requêtes disponibles `field, filter, aliases` dans l'api pour affiner les requêtes et les résultats retournés, et notamment obtenir dans la même requête les données des sandwichs d'une catégories (en particulier, libelle, prix). (documentation https://docs.directus.io/reference/query/)

2. un wrapper HTML pour le catalogue

L'objectif est de réaliser une interface web utilisant un rendu côté serveur basé sur des templates (twig) pour la gestion du catalogue.
Cette interface est un *wrapper HTML* : il s'agit d'un micro-service dont le rôle est d'accéder à l'api catalogue pour récupérer des données JSON et les mettre en forme dans des documents HTML affichables par un navigateur.
Ce wrapper est donc réalisé par un service PHP hébergé dans un conteneur docker. Il est demandé de réaliser 2 affichages :

- affichage de la liste des catégories (libellé, description)
- pour une catégorie, affichage de la liste des sandwichs (libellé, description, tarif)

Chaque fonctionnalité correspond à une url : web.catalogue.local/categories et web.catalogue.local/categorie/{libelle}
Chaque fonctionnalité est réalisée par une action dans un contrôleur qui interroge l'API pour obtenir les données puis crée une vue pour le rendu HTML. Les vues sont construites à l'aide de templates Twig.