BileMo RESTFul API

Spécifications

1 - Demande

Contexte du projet

L'entreprise BileMo propose un catalogue de mobiles haut de gamme. Son business model est de vendre en B2B, elle ne fait donc pas de vente en direct. L'entreprise souhaite proposer aux plateformes qui le souhaitent, l'accès à son catalogue via des API.
L'objectif sera donc d'exposer un certain nombre d'API pour que les les applications des plateformes puissent effectuer leurs opérations.

Besoin exprimé.

Exposition d'une API avec les fonctionnalités suivantes:

Pour les produits

- Consultation de la liste des produits BileMo
- Consulter le détail d'un produit BileMo

Pour chaque client (plateforme)

- Consulter la liste de ses utilisateurs inscrits
- Consulter le détail d'un utilisateur inscrit
- Ajouter un nouvel utilisateur
- Supprimer un utilisateur

Sécurité

- Seul un client enregistré pourra accéder aux API.
- Les clients devront être authentifiés soit via un serveur OAuth soit via JWT.

Contraintes techniques

- Créer une API RESTFul (respectant les niveaux 1,2 et 3 du modèle de Richardson)
- Les données devront être exposées au format json
- Gestion de la mise en cache afin d'optimiser les requêtes en direction de l'API.

2 - Solution proposée

Spécifications fonctionnelles

Les ressources de l'API

Cette API étant à destination des plateformes, elle n'explosera donc que les ressources qui leur seront nécessaires :

Les produits

Les produits BileMo seront uniquement consultables par les plateformes. L'API permettra donc les actions suivantes:

- Consulter la liste des produits
- Consulter le détail d'un produit

Note: Tous les clients enregistrés auront accès sans restriction à l'ensemble du catalogue produit BileMo.

Les utilisateurs

Chaque client pourra, au travers de l'API gérer ses utilisateurs. Chaque utilisateur sera obligatoirement lié à un client, et un client n'aura accès qu'à sa propre liste d'utilisateurs.
Liste des opérations possibles :

- Consulter la liste des utilisateurs
- Consulter le détail d'un utilisateur
- Ajouter un utilisateur
- Supprimer un utilisateur

La sécurité

La sécurité sera gérée en plusieurs étapes:

Administrateur client.

Une fois un client enregistré par BileMo, un utilisateur administrateur sera créé pour ce client. Seul l'administrateur client pourra manipuler et avoir accès à la ressource **utilisateurs**.

Authentification (JWT)

Le système d'authentification se fera via Json Web Token.
A la première connexion, l'utilisateur soumettra ses identifiants et se verra retourner un token de sécurité. Celui-ci devra ensuite être impérativement passé en en tête de chaque requête afin de pouvoir accéder et manipuler les différentes ressources de l'API mises à sa disposition.

Roles

Il y aura 2 type d'utilisateurs pour chaque client:

- **Admin**: Les administrateurs pourront accéder à la ressource **produits** en lecture seule, et manipuler la ressource **utilisateurs** (lister, voir, ajouter et supprimer)
- **User**: Les utilisateurs auront seulement accès à la ressource **produits** en lecture également.

Documentation

Une documentation de l'API sera mise à la disposition des clients afin de faciliter la manipulation des ressources.

Spécifications techniques

Environnement

PHP 7.4
SGBD: MySQL 5.7
Symfony LTS: 4.4
ApiPlatform 2.5

Entités

Customer

Attribute	Description	Info
name	Company name	String (3 - 50) - Unique - required

User

Attribute	Description	Info
username		String (3 - 50) - Unique - required
email		String (5 - 80) - Unique - required
password		String (255) - required - hashed
role		Array (ROLE_USER | ROLE_ADMIN)
customer	User customer	Relation (Customer) - ManyToOne

Product

Attribute	Description	Info
name	Product name	String (3 - 255) - Unique - required
description	Product description	Text - required
color	Product color	String (3 - 20)
price	Customer status	Smallint

Ressources de l’API

Product

Nom de la ressource : **product**.
Entité **Product**

Endpoints

GET /products

Roles	ROLE_USER
Options	
page	Default: 10 per page
search	Partial - optional
Responses	
200	Ressources trouvées
204	Aucune ressource dans la base
404	Aucune ressource trouvée (Si option **Search** utilisée)

GET /product/{id}

Roles	ROLE_USER
Responses	
200	Ressource trouvée
404	Ressource non trouvée

User

Nom de la ressource: **user**
Entité **User**

Entrypoints

GET /users

Roles	ROLE_ADMIN
Options	
page	Defaut: 10 par page
search	Partial - optionnel
Responses	
200	Ressources trouvées
204	Aucune ressource dans la base
404	Aucune ressource trouvée (Si option **Search** utilisée)

POST /users

Roles	ROLE_ADMIN
Responses	
201	Ressource créée
400	Entrée invalide

GET /users/{id}

Roles	ROLE_ADMIN
Responses	
200	Ressource trouvée
404	Aucune ressource trouvée (Si option **Search** utilisée)

PUT /users/{id}

Roles	ROLE_ADMIN
Responses	
200	Ressource modifiée
400	Entrée invalide
404	Aucune ressource trouvée (Si option **Search** utilisée)

DELETE /users/{id}

Roles	ROLE_ADMIN
Responses	
204	Ressource supprimée
404	Aucune ressource trouvée (Si option **Search** utilisée)

La sécurité

Résumé

Sécurité implémentée via JWT.
Librairie **lexik/LexikJWTAuthenticationBundle**.

Token de sécurité (JWT Authentication)

A la première connexion, l'utilisateur soumettra ses identifiants via l'endpoint **/login**, et se verra retourner un token de sécurité. Celui-ci devra ensuite lui être impérativement passé en en tête de chaque requête via l'en tête **Authorization: Bearer TOKEN** afin de pouvoir accéder et manipuler les différentes ressources mises à sa disposition.

Roles

ROLE_ADMIN

Seul l'admin aura accès à la ressource **utilisateurs.**

ROLE_USER

L'utilisateur aura seulement accès à la ressource **produits**.

Utilisateurs

Deux types d'utilisateurs:

Admin
Un admin sera créé à chaque enregistrement d'un nouveau client.
Il héritera des rôles: ROLE_USER et ROLE_ADMIN.
Il aura donc la possibilité de manipuler la ressource utilisateurs du client auquel il est affilié.
Il aura également accès à la ressource **utilisateurs**.

Utilisateur
Il aura seulement le rôle ROLE_USER.
Il n'aura donc la possibilité d'accéder qu'à la ressource **produits.**

Format de données

Les données seront retournées au format **json**.
Note: Elles seront également disponibles au format JSON-LD pour le cas de l'utilisation d'un client gérant ce format.