



Documentation technique

BILEMO - API



Version : 0.1

Date de la dernière mise à jour : 12 octobre 2021

Sommaire

BILEMO - API	1
Cadre du projet	3
Résumé	3
Enjeux et objectifs	3
Spécifications techniques	4
Utilisation de l'API	4

1. Cadre du projet

1.1. Résumé

La mission est de créer un webservice exposant une API pour le nouveau client de l'entreprise BILEMO. Le client requiert des fonctionnalités qui sont les suivantes :

- consulter la liste des produits BileMo ;
- consulter les détails d'un produit BileMo ;
- consulter la liste des utilisateurs inscrits liés à un client sur le site web ;
- consulter le détail d'un utilisateur inscrit lié à un client ;
- ajouter un nouvel utilisateur lié à un client ;
- supprimer un utilisateur ajouté par un client.

1.2. Enjeux et objectifs

L'objectif du projet est donc la création d'une API fonctionnelle et sécurisée permettant à BILEMO et à ses clients de consommer l'API pour de la vente B2B (business to business).

Concernant la sécurité de l'API, celle-ci doit, pour son utilisation, requérir à une authentification via une validation par JWT (Json Web Token).

Suivant le souhait du client, l'API devra respecter les niveaux 1, 2 et 3 du modèle de Richardson, elle sera donc auto-découvrable.

Une mise en cache des données sera implémentée suivant le souhait du client.

```
{
  "token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpYXQiOiJlbnRvbmVzZXJuYW1lIjoibHVjaWUuZGVzY2hhbXBzQGxIY29tdGUub3JnIn0.bmJ5VfuuaFhDAqyJCGKftWPQCJo9G5TSOt2ISw38ucfZHcyBujB3c2oEHw4BOTvhTTt08SZd2Vrlbo0wlIwi9CnaV2o7YoIr8J4VunWnjUUMd4Sz2Byvadj6G0uyyrXLFW2U0svp3BAUe2Ksc1AOFEQQTqEKrbHLKx2EkIDCDaJpJtMZ9-Ei3Ifk6Sc1YtoXUXEpsZk53mCIURrMwBI7JAfEkzbwsMR_CXWywfXt3Nv_fubELvkbsk_MHHEtjVUARcvF-flg9TVxnBDyex05HbFH6Uck_LdYxpCSFAktlyGyDQzoL5AOOKDAcqGXbvAAWwY9oWEFDvY_rCZDAQ"
}
```

3.2. Consultation de la liste des produits BileMo (GET)

- <https://bilemo.baeteromain.fr/api/products>

Via une requête GET, et avec une “Authorization” de type Bearer + Token dans le header de la requête, vous récupérerez les téléphones sous la forme :

```
{
  "@id": "/api/products/27",
  "@type": "Product",
  "id": 27,
  "model": "Exemple de modèle",
  "brand": "/api/brands/33"
},
{...}
```

Les informations disponibles sont l'id, le modèle, et la marque du téléphone (dont l'URI est disponible vous permettant de récupérer les informations de la marque du téléphone).

Des filtres peuvent être appliqués vous permettant de trier les téléphones en réponse :

- price[between]
- price[gt]
- price[gte]
- price[lte]
- price[lte]
- order[price] : asc OU desc

Exemple de la requête avec un tri par prix décroissant :

- [https://bilemo.baeteromain.fr/api/products?order\[price\]=desc](https://bilemo.baeteromain.fr/api/products?order[price]=desc)

3.3. Consultation des détails d'un produit BileMo (GET)

- <https://bilemo.baeteromain.fr/api/products/{ID}>

Via une requête GET, et avec une “Authorization” de type Bearer + Token dans le header de la requête, vous récupérerez les détails du téléphone ayant l'ID sous la forme :

```
{
  "@context": "/api/contexts/Product",
  "@id": "/api/products/22",
  "@type": "Product",
  "id": 22,
  "model": "Exemple de modèle",
  "description": "Exemple de description.",
  "price": 192,
  "quantity": 2,
  "created_at": "2019-11-13T14:01:32+00:00",
  "updated_at": "2021-11-03T10:36:21+00:00",
  "brand": "/api/brands/32"
}
```

Les informations disponibles sont l'id, le modèle, la description du téléphone, le prix, la quantité, la date de création, la date de dernière modification, et la marque (sous forme d'URI)

3.4. Consultation de la liste des utilisateurs inscrits liés à un client sur le site web (GET)

- <https://bilemo.baeteromain.fr/api/users>

Via une requête GET, et avec une “Authorization” de type Bearer + Token dans le header de la requête, vous récupérerez les utilisateurs sous la forme :

```
{
  "@id": "/api/users/64",
  "@type": "User",
  "id": 64,
  "username": "exemple username",
  "email": "exemple@exemple.org",
  "firstname": "exemple prénom",
  "lastname": "exemple nom",
  "created_at": "2021-07-29T09:11:30+00:00",
  "zip_code": "XX XXX",
  "address": "XX, exemple adresse",
  "city": "Exemple ville",
  "phone_number": "07 XX XX XX XX",
  "country": "FRANCE"
},
{...}
```

Les utilisateurs retournés via cette requête sont seulement les utilisateurs rattachés au client actuellement authentifié.

3.5. Consultation du détail d'un utilisateur inscrit lié à un client (GET)

- <https://bilemo.baeteromain.fr/api/users/{ID}>

Sur le même principe que l'exemple précédent, via une requête GET, et avec une “Authorization” de type Bearer + Token dans le header de la requête, vous récupérerez les utilisateurs sous la forme :

```
{
  "@context": "/api/contexts/User",
  "@id": "/api/users/64",
  "@type": "User",
  "id": 64,
  "username": "exemple username",
  "email": "exemple@exemple.org",
  "firstname": "exemple prénom",
  "lastname": "exemple nom",
  "created_at": "2021-07-29T09:11:30+00:00",
  "zip_code": "XX XXX",
  "address": "XX, exemple adresse",
  "city": "Exemple ville",
  "phone_number": "07 XX XX XX XX",
  "country": "FRANCE"
}
```

Si le client fait une recherche d'un utilisateur qui ne lui est pas rattaché, l'API renverra une erreur avec un code **403 Forbidden** contenant un message de description lui indiquant qu'il ne peut voir seulement ses utilisateurs.

3.6. Ajout d'un nouvel utilisateur lié à un client (POST)

- <https://bilemo.baeteromain.fr/api/users>

Via une requête POST, et avec une "Authorization" de type Bearer + Token dans le header de la requête, un client peut ajouter un utilisateur (qui sera automatique rattaché à lui même) en fournissant dans le body de la requête sous format JSON les informations suivantes :

```
{
  "username": "string",
  "password": "string",
  "email": "string",
  "firstname": "string",
  "lastname": "string",
  "zip_code": "string",
  "address": "string",
  "city": "string",
  "phone_number": "string",
  "country": "string"
}
```

3.7. Suppression d'un utilisateur ajouté par un client (DELETE)

- <https://bilemo.baeteromain.fr/api/users/{ID}>

Via une requête DELETE, et avec une "Authorization" de type Bearer + Token dans le header de la requête, un client peut supprimer un de ses utilisateurs en fournissant via son ID. Une réponse code 204 sera retourné validant la suppression de l'utilisateur.

Un client ne peut supprimer que ses propres utilisateurs, dans le cas où il fournit l'ID d'un utilisateur qui n'est pas le sien, l'API renverra une erreur avec un code **403 Forbidden** contenant un message de description lui indiquant qu'il ne peut supprimer que ses propres utilisateurs.