

Conception & Développement d'Applications et Services Web/Serveur**TD7 : authentification, token JWT**

L'objectif du TD est de construire un service d'authentification et de le mettre en place pour contrôler les accès à l'application de backoffice LBS.

contexte : lors de l'accès au backoffice LBS, par exemple pour utiliser l'API de suivi de fabrication, il est nécessaire de contrôler que le client émettant la requête possède les droits pour accéder aux ressources.

Pour cela, on met en place un service d'authentification basé sur l'utilisation de token JWT. Ce service est capable de délivrer des tokens sur la base de credentials qui lui sont transmis (authenticate) , et de vérifier la validité d'un token JWT qui lui est soumis (check).

Ce service est une application indépendante qui expose sa propre API et utilise sa propre base de données. Cette application est la seule à détenir les mots de passe des utilisateurs.

Sur un plan pratique, ce service est implanté dans deux conteneurs docker : un conteneur docker correspondant à l'API, et un conteneur hébergeant la base de données. Il est donc nécessaire de compléter la configuration docker-compose.yml pour ajouter ce service.

1. Mise en place du service, authentification et création d'un token JWT

Mettre en place le service d'authentification dans la composition docker. Le service API doit répondre sur l'url `api.auth.local` et utiliser sa propre base de données. La structure et le contenu de cette base vous est fourni, elle ne contient qu'une seule table `User`.

Programmer la route permettant l'authentification auprès de ce service de d'authentification. On utilise une authentification HTTP Basic. A titre d'exemple, un contrôleur proposant une méthode d'authentification vous est fourni, vous pourrez l'adapter à votre contexte.

Ainsi, la requête d'authentification aura la forme suivante :

```
POST /auth HTTP/1.1
Authorization: Basic bWljaGVsOm1pY2hlaA==
```

En cas d'absence du header `Authorization` ou de credentials invalides, la réponse contient un code 401 :

```
HTTP/1.1 401 Unauthorized

Content-Type: application/json

{
  "type": "error",
  "error": 401,
  "message": "no authorization header present"
}
```

En cas de réussite de l'authentification, la réponse contient un access token JWT et un refresh token opaque. Le token JWT contient le profil de l'utilisateur pour lequel l'authentification est accordée :

```
HTTP/1.1 200 OK
Content-Type: application/json

{
```

```

"access-token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwOlwvXC9hcGkubGJzLmxvY2FsXC9hdXRoIiwiaXVkiOiJoiaHR0cDpcL1wvYXBpLmxicy5sb2NhbmCIslhdCI6MTUxMzY3MTAwMCwiZXhwIjoxNTEzNjc0NjAwLCJjaWQiOiJF9.EtE4iY2XIGf_V0Ai9g62D3XU35IFgSJr6n8ja9j0Lr7JCdqxG-oTosWwruGT028oFm_6pwQzUHwYBCtyZx4AGQ" ,
"refreh-token": "A675E34FA7B43109FEC43218FEDD56"
}

```

2. contrôle de la validité d'un token JWT

Programmer la route qui permet de vérifier la validité d'un token JWT. Le token est transporté dans le header "Authorization" en mode "Bearer"

```

GET /check
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwOlwvXC9hcGkubGJzLmxvY2FsXC9hdXRoIiwiaXVkiOiJoiaHR0cDpcL1wvYXBpLmxicy5sb2NhbmCIslhdCI6MTUxMzY3MTAwMCwiZXhwIjoxNTEzNjc0NjAwLCJjaWQiOiJF9.EtE4iY2XIGf_V0Ai9g62D3XU35IFgSJr6n8ja9j0Lr7JCdqxG-oTosWwruGT028oFm_6pwQzUHwYBCtyZx4AGQ

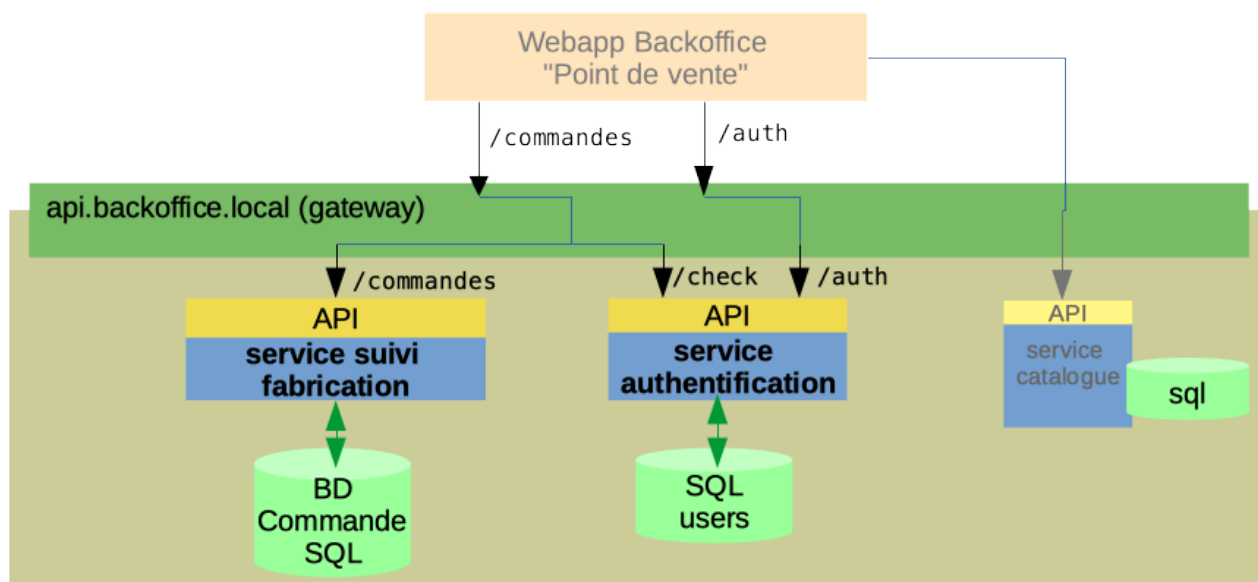
```

Un code 401 est retourné si le header est absent, si le token est invalide ou expiré, accompagné d'un message d'erreur indiquant la raison de la non validation.

Si le token est valide, un code 200 est retourné, accompagné du profil de l'utilisateur authentifié (email, username, level).

3. contrôle d'accès au backoffice LBS

Le backoffice LBS est réalisé par une API passerelle (gateway) dédié qui regroupe et unifie l'accès à l'ensemble des micro-services de gestion de l'application LBS. Cette API expose des routes pour accéder aux ressources du backoffice (suivi et fabrication des commandes, gestion du catalogue), et assure l'authentification et le contrôle d'accès :



Fonctionnement de l'api backoffice :

- lors de la réception d'une demande d'authentification (/auth), la requête est acheminée vers le service d'authentification, et la réponse du service d'authentification est retournée au client (token JWT ou message d'erreur),
- lors de la réception d'une requête d'accès ou de modification d'une ressource, par exemple lister les commandes à préparer, l'api vérifie la présence d'un access token et demande sa vérification au service d'authentification (check).
- Si le token est valide et les droits associés au profil de l'utilisateur suffisant, la requête d'accès est adressée au service de suivi de fabrication, et la réponse est retournée au client. Dans le cas contraire un message d'erreur est retourné au client.

Il est demandé de réaliser l'api gateway du backoffice en implantant uniquement 2 routes :

- la route permettant au client de s'authentifier,
- la route d'accès à la liste des commandes