Conception & Développement d'Applications et Services Web/Serveur

TD7 : authentification, token JWT

L'objectif du TD est de construire un mécanisme d'authentification auprès du service de fidélisation en utilisant un token JWT.

contexte : lors du paiement d'une commande, le client peut choisir, s'il le souhaite, de participer au programme de fidélisation, c'est à dire inscrire le montant de la commande sur sa carte de fidélité et éventuellement bénéficier d'une remise.
Le service de fidélisation et de gestion des cartes de fidélité est un service autonome, séparé du service de prise de commande. L'utilisation d'une carte de fidélité est protégée par un mot de passe, que ce service est seul à détenir.

Pour fidéliser son paiement, le client doit :

1. s'authentifier auprès du service de fidélisation pour obtenir un token JWT,

2. ce token JWT lui permet d'accéder à sa carte de fidélité pour, par exemple, connaître le montant cumulé qui y est inscrit,

3. lors du paiement d'une commande auprès du service de prise de commande, le client peut ajouter son numéro de carte et le token JWT obtenu, ce qui va permettre à ce service d'enregistrer la commande sur cette carte auprès du service de fidélisation.

Des données pour créer la base de données du service de fidélisation sont disponibles dans arche.

1. authentification et création d'un token JWT

Programmer la route permettant l'authentification d'un client auprès du service de fidélisation. On utilise une authentification HTTP Basic.

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

```
POST /cartes/015f4eb0-0a82-4290-aa4f/auth HTTP/1.1
Authorization: Basic bWljaGVsOm1pY2hlbA==
```

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 token JWT. Ce token contient le numéro de la carte pour laquelle l'authentification est accordée :

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

{
  "token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwOlwvXC9hcGkubG
JzLmxvY2FsXC9hdXRoIiwiYXVkIjoiaHR0cDpcL1wvYXBpLmxicy5sb2NhbCIsImlhdCI6MTUxMzY3MTAw
MCwiZXhwIjoxNTEzNjc0NjAwLCJjaWQiOjF9.EtE4iY2XIGf_V0Ai9g62D3XU35IFgSJr6n8ja9jOLr7JCdqxG-
oTosWwruGT028oFm_6pwQzUHwYBCtyZx4AGQ"

}
```

2. accès à une ressource de type carte

Programmer la route qui permet à un client d'accéder à sa carte de fidélité. Ceci lui permet notamment de consulter le montant cumulé d'achats qu'il a réalisé.
L'accès n'est autorisé que pour les requêtes proposant un token JWT valide pour le client auquel on accède. Le token est transporté dans le header "Authorization" en mode "Bearer"

```
GET /cartes/015f4eb0-0a82-4290-aa4f
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwOlw
vXC9hcGkubGJzLmxvY2FsXC9hdXRoIiwiYXVkIjoiaHR0cDpcL1wvYXBpLmxicy5sb2NhbCIsImlhdCI6MT
UxMzY3MTAwMCwiZXhwIjoxNTEzNjc0NjAwLCJjaWQiOjF9.EtE4iY2XIGf_V0Ai9g62D3XU35IFgSJr6n8ja
9jOLr7JCdqxG-oTosWwruGT028oFm_6pwQzUHwYBCtyZx4AGQ
```

Un code 401 est retourné si le header est absent, si le token est invalide ou s'il est valide pour une carte différente de celle demandée.

3. Payer une commande en mode fidélisé

Créer la route pour le paiement d'une commande auprès du service de prise de commande. Le client transmet l'identifiant de commande et son numéro de carte bancaire.
De manière optionnelle, il peut demander la fidélisation de son paiement en transmettant en complément son identifiant de carte de fidélité et le token JWT obtenu grâce à l'authentification auprès du service fidélisation.
Dans ce cas, le service de prise de commande enregistre le paiement et transmet les données de fidélisation (identifiant de commande, montant de la commande, identifiant de carte de fidélité, token JWT) au service de fidélisation en utilisant une route dédiée dans l'api. Le service de fidélisation enregistre la commande après vérification du token JWT.