TP 05 - Implement Authentication

Individuel ou Groupe de 2 - Rendu attendu

Objectif du TP

L'objectif de ce TP est d'implémenter différents mécanismes de sécurité pour votre API. Le but de ce TP est de gérer principalement l'authentification et non les différentes autorisations.

Une petite API très simple accompagne ce TP et vous sert de base de code, à vous d'implémenter les différents mécanismes de sécurité pour protéger celle-ci.

Au cours de ce TP vous devrez documenter vos choix en particulier pour la partie OAuth 2.

Rendu du TP

Le rendu de ce TP s'effectue sur votre classroom Github. N'oubliez pas de versionner votre rendu dès le début et d'ajouter un **README.md** complet documentant votre architecture et les choix que vous avez effectué.

Les fichiers Docker sont fournis avec le TP.

Pour simplifier les tests et la correction, vous trouverez 4 variables d'environnement dans le docker-compose qui correspondent aux 4 types d'authentifications. Ces variables d'environnement sont des booléens, lorsque le booléen est à *false* la méthode d'authentification correspondante doit être désactivée.

1 Basic authentication

Implémentez un mécanisme d'authentification basique avec Express. Il vous est conseillé d'utiliser le module npm correspondant.

N'oubliez pas de mentionner dans le README les credentials.

2 JWToken authentication

Réalisez une authentification custom à l'aide d'un token JWT. L'utilisateur doit poster sur votre API son nom d'utilisateur et mot de passe. Ces informations doivent être stockées dans une base de données que vous ajouterez dans le docker-compose. Créez une table/collection User contenant une adresse email et un mot de passe. Si le jeton n'est pas juste vous renverrez une erreur 40X.

Vous utiliserez une des librairies npm permettant la gestion de JWT.

3 API Key

Dans une Table/Collection “Applications” doivent être listés des applications liées à une clé API que vous avez au préalablement générée.

Votre API doit accepter la clé API en header HTTP a chaque requête. Si la clé API n’est pas juste vous renverrez une erreur 40X.

4 OpenID Provider

Vous créerez un provider OpenID et choisirez le flow adéquat pour votre cas d’utilisation. Spécifiez dans votre README quel flow vous avez choisi et pourquoi.

L’implémentation de OpenId se fait via l’utilisation d’une librairie de provider, telle que https://github.com/panva/node-oidc-provider.

Dans votre API, vous effectuerez ensuite l’implémentation cliente via passport et la stratégie passport de votre choix, telle que https://github.com/jaredhanson/passport et https://github.com/jaredhanson/passport-openid