Une image contenant texte, Police, Graphique, calligraphie

Description générée automatiquement

**BUT 3 INFORMATIQUE**

**Parcours A FI**

**Réalisations d'applications**

**RAPPORT TP4**

**Authentification et Autorisation**

**Prénom et Nom de l’étudiant :** Akash Selvaratnam

**Groupe :** 303

**Promotion :** 2023-2024

# Sommaire

# Étape 1 – « De base… »

Au départ, j’ai pu prendre connaissance du code présent avec le server.js qui peut être lancé sur le port 3000 en localhost avec Fastify qui propose deux différentes requête GET <http://localhost:3000/dmz> et <http://localhost:3000/secu> et qui valide seulement les personnes ayant pour nom d’utilisateur Tyrion et mot de passe wine.

Pour l’étape 1, j’ai pu tester via Postman deux différentes requêtes Get <http://localhost:3000/dmz> et <http://localhost:3000/secu> et observez le résultat retourné.

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

Figure 1 : Requête <http://localhost:3000/dmz>

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Figure 2 : Requête <http://localhost:3000/secu>

Ensuite, j’ai pu une authentification pour la requête <http://localhost:3000/secu> sur postman en indiquant le nom ainsi que le mot de passe que nous avons indiqué dans la validate du fichier server.js afin que le retour « Un Lannister paye toujours ses dettes ! » puisse être envoyé car il est envoyé seulement si il respecte les authentification.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Figure 3 : Configuration de l'Authorization

Le rôle de la fonction after() est d’être exécuter lorqu’e l’ensemble des plugins ont terminé de charger, il est toujours exécuter avant la fonction ready().

Dans la fonction after(), j’ai ajouté une nouvelle route /autre du type GET mais à la différence de la route /secu cela doit être accessible sans les authentification donc sans le onRequest.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Figure 4 : Fonction after()

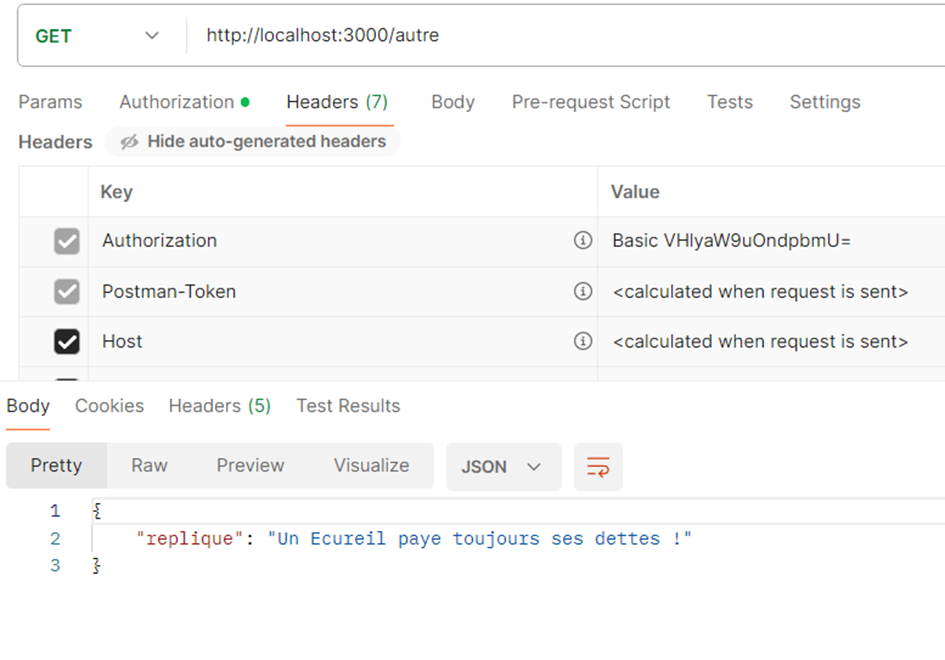


Figure 5 : Requête <http://localhost:3000/autre>

# Étape 2 – Prouves qui tu es !

Pour l’étape 2, j’ai créer une nouvelle clé RSA de 2048 bits nommé server.key avec la commande suivante openssl genrsa -out server.key 2048 .

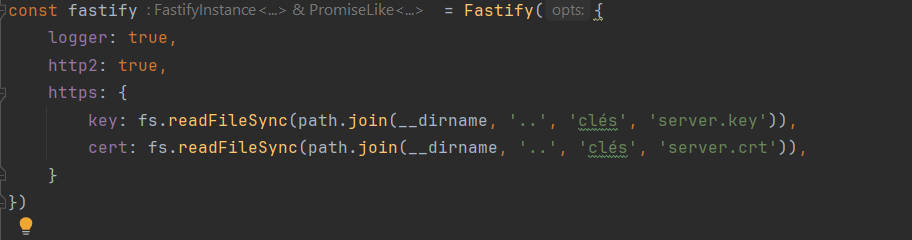
J’ai ensuite crée un nouveau Certificate Signing Request en signant mon certificat avec la clé privé créer précédemment à l’aide des commandes proposé sur les slides 35 et 36 et ensuite j’ai pu tester le certificat généré via la commande proposé dans le cours permettant de vérifier le certificat sur le port 4567 avec l’application Postman.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 6 : Test du Certificat sur Postman

J’ai ensuite configuré Fastify en HTTPS en indiquant le chemin vers ma clé privée ainsi que vers ma certification, j’ai du également importer des modules telles fs pour pouvoir lire les fichiers et le module path.



# Étape 3 – Un jeton dans la machine