P\_APP-183



(Une image originale représentant le projet)

Rotzetter Ethan – MID2A

Lausanne, Venne

24 P

Gaël Sonney

Table des matières

[1. conceptualisation (schéma) 2](#_Toc190096657)

[2. explications sur le code 2](#_Toc190096658)

[1.1 Profil du client 2](#_Toc190096659)

[1.2 HTTPS 2](#_Toc190096660)

[1.3 Authentification par mot de passe 3](#_Toc190096661)

[1.4 Vérification du token JWT 3](#_Toc190096662)

[1.5 Administration 4](#_Toc190096663)

[1.6 Protection contre les injections SQL 4](#_Toc190096664)

[1.7 Utilisation de bcrypt 4](#_Toc190096665)

[1.8 Versioning 4](#_Toc190096666)

[3. conclusion sur le travail fourni et sur l’attitude face au projet 4](#_Toc190096667)

[4. Webographie 4](#_Toc190096668)

# conceptualisation (schéma)

# explications sur le code

## Profil du client

## HTTPS

Voici les étapes de réalisation pour avoir HTTPS :

Je commence par faire cette commande :

openssl genpkey -algorithm RSA -out privkey.key

Une image contenant texte, capture d’écran, Police

Le contenu généré par l’IA peut être incorrect.

Puis celle là:

openssl req -new -key privkey.key -out request.csr

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Le contenu généré par l’IA peut être incorrect.

Et pour finir :

openssl x509 -req -in request.csr -signkey privkey.key -out certificate.crt -days 365

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

Le contenu généré par l’IA peut être incorrect.

**Résumé des commandes:**

générer une clé privée :

- openssl genpkey -algorithm RSA -out privkey.key

créer une demande de signature de certificat (CSR) :

- openssl req -new -key privkey.key -out request.csr

signer cette demande pour produire un certificat auto-signé :

- openssl x509 -req -in request.csr -signkey privkey.key -out certificate.crt -days 365

## Authentification par mot de passe

## Une image contenant capture d’écran Le contenu généré par l’IA peut être incorrect.Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia Le contenu généré par l’IA peut être incorrect.

## Vérification du token JWT

## Administration

## Protection contre les injections SQL

## Utilisation de bcrypt

## Versioning

# conclusion sur le travail fourni et sur l’attitude face au projet

# Webographie

Voici une liste des sites utilisés pour le projet :

* [Créer un serveur HTTPS | NodeJS](https://node-js.fr/server/ssl.html)
* [Node JS fs.readFileSync() Method - GeeksforGeeks](https://www.geeksforgeeks.org/node-js-fs-readfilesync-method/)