P\_APP-183



Rotzetter Ethan – MID2A

Lausanne, Venne

24 P

Gaël Sonney

Table des matières

[1. conceptualisation (schéma) 2](#_Toc192239290)

[2. explications sur le code 3](#_Toc192239291)

[1.1 Profil du client 3](#_Toc192239292)

[1.2 HTTPS 3](#_Toc192239293)

[1.3 Authentification par mot de passe 5](#_Toc192239294)

[1.4 Vérification du token JWT 7](#_Toc192239295)

[1.5 Administration 7](#_Toc192239296)

[1.6 Protection contre les injections SQL 8](#_Toc192239297)

[1.7 Utilisation de bcrypt 8](#_Toc192239298)

[1.8 Versioning 9](#_Toc192239299)

[3. conclusion sur le travail fourni et sur l’attitude face au projet 9](#_Toc192239300)

[4. Webographie 9](#_Toc192239301)

# conceptualisation (schéma)

Une image contenant capture d’écran, conception

Description générée automatiquement

Figure Schéma

# explications sur le code

## Profil du client

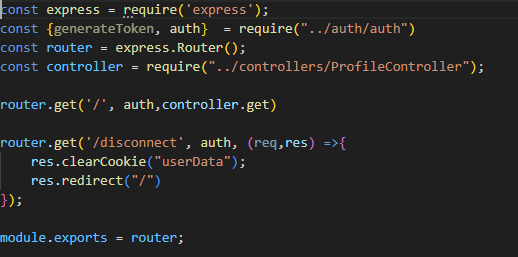


Figure Le controleur de profile



Figure Le code du Router profile

## 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

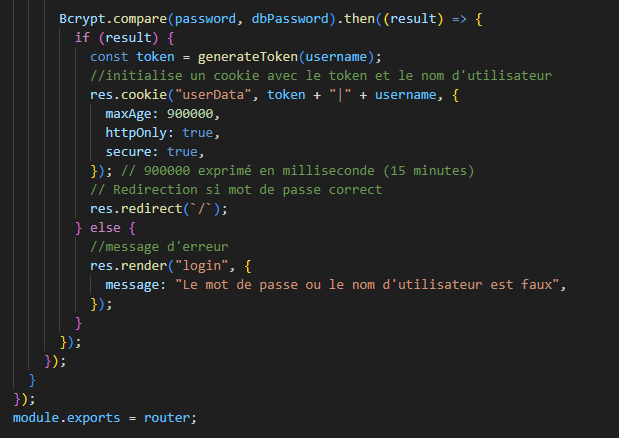
signer cette demande pour produire un certificat auto-signé :

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

## Authentification par mot de passe







## Vérification du token JWT

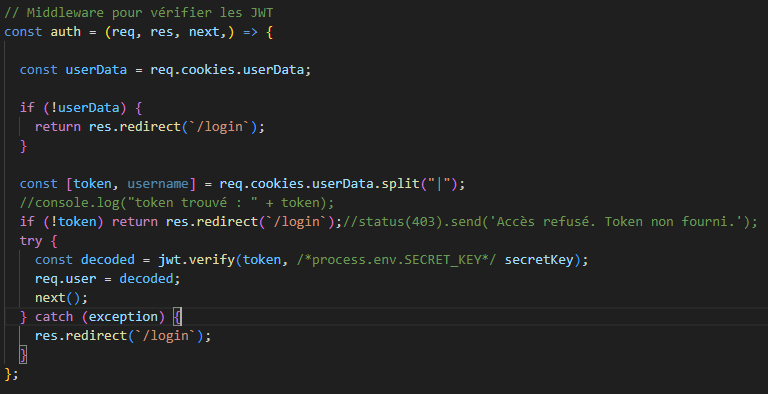


Figure Middleware "auth"

## Administration

## Protection contre les injections SQL

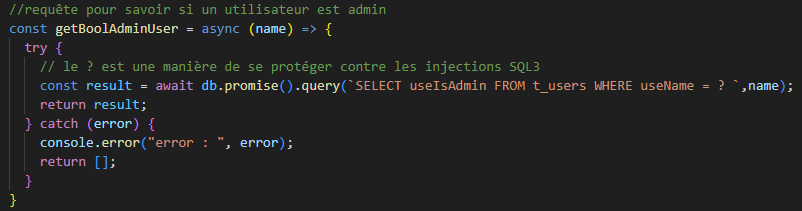


Figure Exemple de protection

## Utilisation de bcrypt

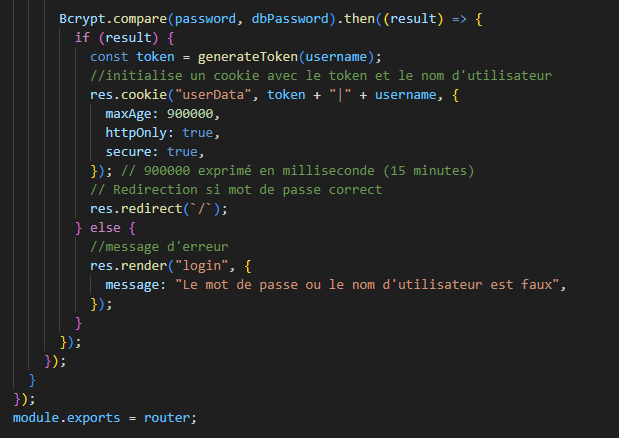


Figure Bcrypt pour le login

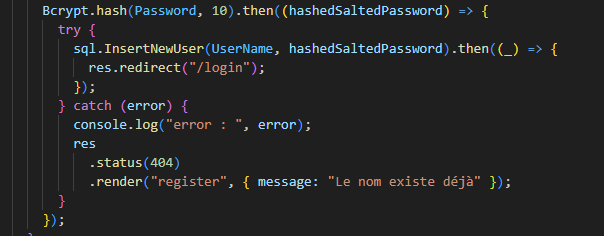


Figure Bcrypt pour le registre

## Versioning

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

# Webographie

Voici une liste approximative des sites utilisés pour le projet :

(Je mentionne des sites dans le code)

* [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/)