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](#_Toc188273472)

[2. explications sur le code 2](#_Toc188273473)

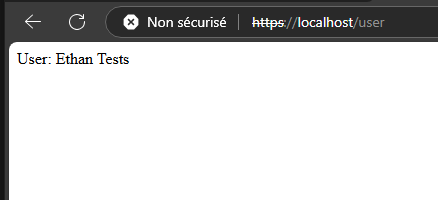
[3. conclusion sur le travail fourni et sur l’attitude face au projet 2](#_Toc188273474)

[4. Webographie 2](#_Toc188273475)

# conceptualisation (schéma)

# explications sur le code

Implémentation de https réussi



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

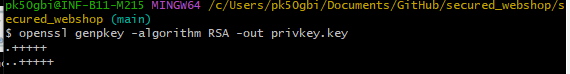
# Webographie

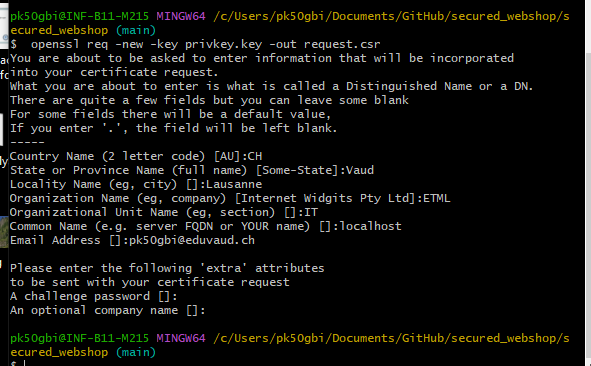
Voici une liste des sites utilisés pour le projet :

* [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/)

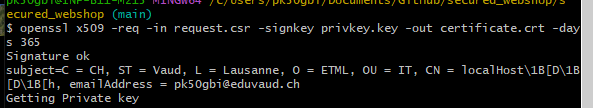
Commande SSL :

openssl genpkey -algorithm RSA -out privkey.key

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

openssl x50

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



**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