Documentation pour l'installation de l'application:

Langage: Java
IDE: Netbeans (Télécharger et Installer la version avec le serveur Tomcat)
Lien pour télécharger: https://netbeans.org/downloads/8.2/ (version Java EE)

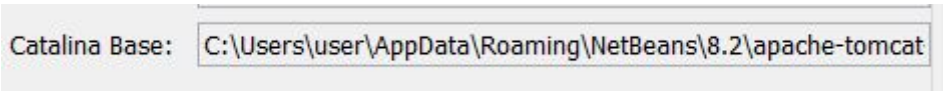
- Créer une base de données de Java DB dans onglet Services:
  - Database Name: tpsmart
  - User Name: tpsmart
  - Password: tpsmart
- Pour créer les tables, exécuter le fichier “createTableSQL.java” dans le répertoire “init”.
- Créer le serveur Tomcat.
- Se connecter à Tomcat.
- Se connecter à la base de données tpsmart.
- Reconstruire la solution avec “Build and Clean”.
- Choisissez le navigateur web avec lequel vous souhaitez ouvrir l'application. Les navigateurs “Google Chrome” et “Firefox” sont à privilégier.
- Exécuter la solution.
- Vous avez maintenant accès à l'application sur votre ordinateur.
- Pour pouvoir accéder à l'application à partir d'un autre appareil il est nécessaire de se connecter en wifi au réseau de l'ordinateur hôte.
- Il sera demandé ensuite l'acceptation de la poursuite de la navigation (problème de certificat) , affichez les informations supplémentaires et choisissez de continuer la navigation.
- Vous avez enfin accès à l'application.
- Quand le navigateur vous demandera le partage de connexion et de la caméra, acceptez pour pouvoir profiter de l'ensemble des fonctionnalités.

Configuration du serveur Apache

- Pour pouvoir accéder à l'appareil caméra d'un portable depuis un navigateur Web, il faut utiliser la méthode MediaDevices.getUserMedia() proposée et le protocole HTTPS.
- Du coup, vous devez configurer le serveur Apache de sorte qu'il écoute des requêtes HTTPS sur un port.
- Commencez en allant dans le dossier bin du java jdk, ouvriez un console puis créez un fichier de certification SSL avec la commande suivant :

```
keytool -genkeypair -alias PldSmartCert -keyalg RSA -keystore "D:\PldSmartCert.cert"
```

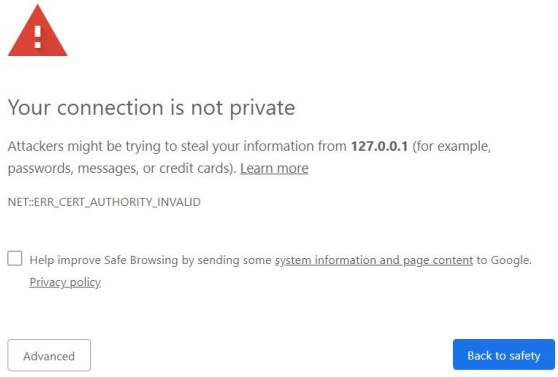
  Vous pouvez remplacer D:\PldSmartCert.cert par un autre chemin si vous préférez
- Ensuite, allez dans la base Catalina du serveur Apache (que le chemin peut être retrouvé avec la méthode ci-dessous) puis ouvriez le fichier server.xml dans le dossier conf.



- Ajoutez un connecteur dans le fichier server.xml

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" />
-->
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
           clientAuth="false" sslProtocol="TLS" keystoreFile="D:\PldSmartCert.cert" keystorePass="tpsmart"/>
```

- Tester la connection HTTPS en allant sur le lien https://localhost:8443, vous devez voir la page suivant :

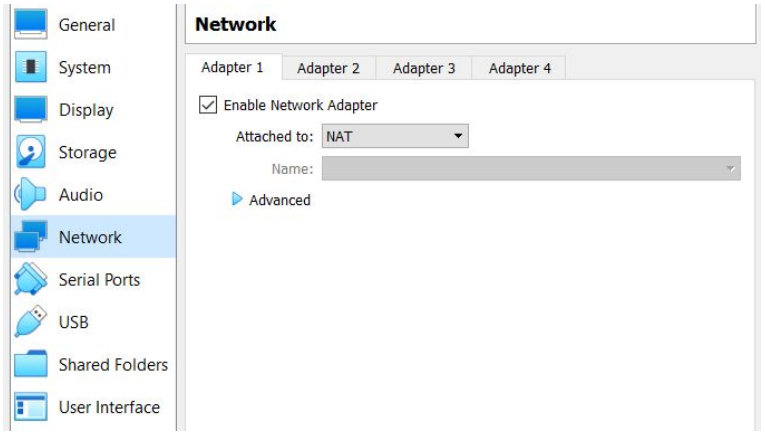


- Valider le certificat en cliquant sur Advanced puis Proceed to localhost (unsafe)
- Vous devez désormais avoir accès au serveur Apache avec le protocol HTTPS

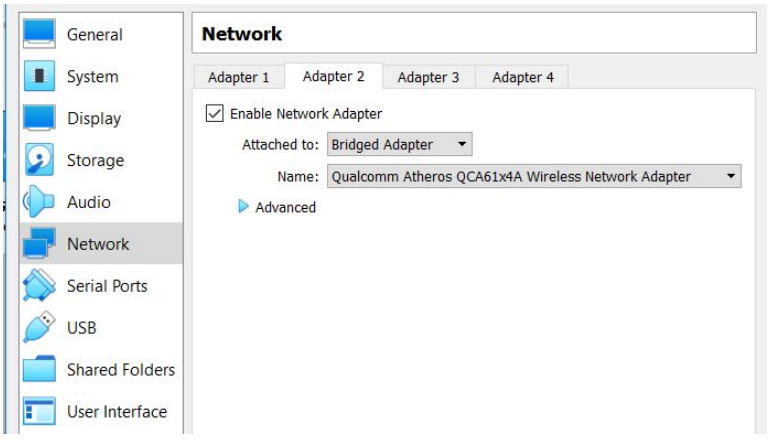
Configuration du TURN Server

Outil utilisé: Oracle VM VirtualBox

- Créer une machine virtuelle ubuntu dans le VirtualBox.
- Et puis configurer ses cartes réseaux comme ci-dessous:
  Une carte NAT:

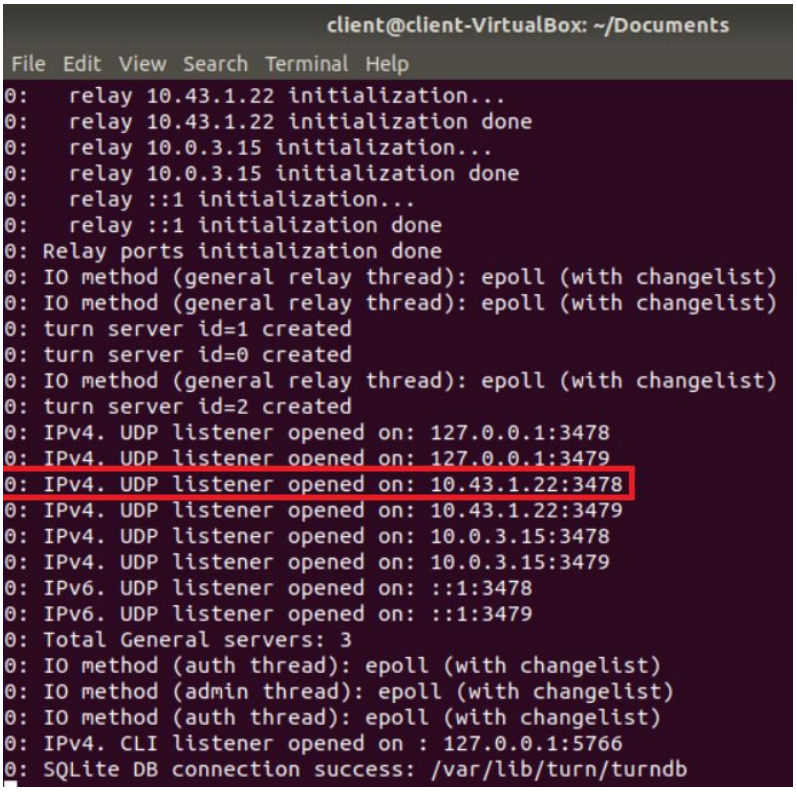


  Une carte Accès à pont:



- Lancer la machine virtuelle, et exécuter la commande ci-dessous pour obtenir l'outil coturn:
  `sudo apt-get install coturn`

- Et puis lancer la commande ci-dessous:
  sudo turnserver -a -v -n --no-dtls --no-tls --no-tcp -u test:test -r "someRealm" --cli-password 123
- Le résultat va être:



L'adresse IP du TURN server entourée par le rectangle rouge est celle qui doit être utilisé dans la **Note**.

Note:
Attention: Ne pas modifier les ports utilisés et les protocoles utilisés
Dans le fichier caller.js:

- Pour la variable globale configuration, l'adresse IP utilisée pour le TURN server doit être l'adresse de la carte réseau "Accès à pont" affichée après lancer le TURN server dans la machine virtuelle.

```
16  const configuration = {iceServers: [{urls: 'turn:10.43.2.20:3478',credential: 'test',
17      username: 'test'}]};
18  //Adresse du serveur TURN à configurer
```

- Dans la fonction connectStart, l'adresse IP utilisée pour la connexion de webSocket doit être l'adresse IP du serveur Tomcat(ordinateur) dans le réseau hotspot ouvert par lui-même.

```
85  function connectStart(number){
86      socket = new WebSocket("wss://192.168.137.1:8443/H4114/video/"+number+"/start/" + name);
87      //Adresse IP et numéro du port du serveur Tomcat en https à configurer
```

Dans le fichier receiver.js:

- Pour la variable globale configurationListen, l'adresse IP utilisée pour le TURN server doit être l'adresse de la carte réseau "Accès à pont" affichée après lancer le TURN server dans la machine virtuelle.

```
10  const configurationListen = {iceServers: [{urls: 'turn:192.168.1.102:3478',credential: 'test',
11      username: 'test'}]};
12  //Adresse du serveur TURN à configurer
```

- Dans la fonction connectListen, l'adresse IP utilisée pour la connexion de webSocket doit être l'adresse IP du serveur Tomcat(ordinateur) dans le réseau hotspot ouvert par lui-même.

```
38      socketListen = new WebSocket("wss://192.168.137.1:8443/H4114/video/" + number + "/listen/" + name);
39      //Adresse IP et numéro du port du serveur Tomcat en https à configurer
```

❖ Important : Il faut aller sur le site https://(IP du serveur Tomcat):(port pour le protocole HTTPS)/ pour accepter la certification avant de pouvoir envoyer une requête vers le Websocket.