

COMPTE RENDU
DECOUTY Hugo / JUBERT Baptiste
TEA Systèmes répartis

Partie de DECOUTY Hugo

Cas d'utilisation de Server_Launcher.jar

Pour faire fonctionner l'exécutable, il suffit de démarrer depuis le terminal, le .jar avec deux numéros de ports associé au serveur en argument (facultatif).

Le premier numéro de port correspond au port du serveur HTTP et le deuxième numéro de port correspond au port du serveur HTTPS, comme ceci par exemple :

```
java -jar Server_Launcher 8080 3500.
```

Si vous ne spécifiez pas de numéro de port, le port 2323 sera utilisé par défaut dans le cas du HTTP et 2324 dans le cas du HTTPS.

Une fois le serveur lancé, pour afficher une page, il suffit d'utiliser un navigateur, en utilisant une URL telle que «localhost:<n° de port>/<nom de la page >», exemple :

« http://localhost:3490/get_form.html »

Cela permet d'envoyer des requêtes HTTP sur le port 3490 et d'afficher la page get_form.html.

Pour utiliser HTTPS il faut modifier « http:// » par « https:// » et modifier le numéro de port par le port associé aux requêtes HTTPS. Si le code est exécuté dans un terminal, il sera possible de consulter les informations transmises lors des requêtes.

Travail réalisé et justification des choix techniques :

Le dossier que nous vous avons envoyé contient un exécutable : Server_Launcher, le certificat pour que le site web soit vérifié, le .jks pour HTTPS, les trois fichiers html qui permettent de tester le navigateur et le code source dans le dossier src.

Notre code source réalise la création d'un navigateur avec un protocole HTTP/HTTPS, avec la récupération des résultats de formulaire GET et POST et prend en compte le multithreading.

Au démarrage du serveur, celui-ci crée deux threads : un thread pour l'écoute de requêtes HTTP et l'autre pour l'écoute de requêtes HTTPS.

Pour vérifier que les requêtes soient bien effectuées, il faut lancer l'application sur un terminal.

Vous pourrez ainsi voir en temps réel, les informations des requêtes GET et POST et le contenu des formulaires.

En ce qui concerne mon implication personnelle au sein du projet, j'ai réalisé entièrement la partie HTTPS (certificat, insertions du code), la partie exécutable, la partie GET et la conception des trois différentes pages web pour tester le navigateur.

Les trois différents fichiers html permettent de tester l'étendue des fonctionnalités de notre code source :

- index.html : Page d'accueil. Elle servira aussi de page de redirection lors des formulaires.
- get_form.html : Page du formulaire utilisant la méthode GET.
- post_form.html : Page du formulaire utilisant la méthode POST.

Pour la conception des certificats, il a fallu générer un fichier keystoreFile.jks à l'aide de la commande ci-dessous :

```
hdecouty@DESKTOP-BIKAKJR:~/c:/users/hugod/desktop/CM_Info/L1_Info/SC/SC Systemes repartit/TLA systemes repartit$ keytool -genkeypair -alias keystoreFile -ext san=dns:localhost,dns:127.0.0.1,ip:127.0.0.1,ip:::
1 -keyalg RSA -keypass hugobaptiste -storepass hugobaptiste -keystore keystoreFile.jks
What is your first and last name?
[Unknown]: Hugo Decouty
What is the name of your organizational unit?
[Unknown]: ULR
What is the name of your organization?
[Unknown]: ULR
What is the name of your City or Locality?
[Unknown]: La Rochelle
What is the name of your State or Province?
[Unknown]: France
What is the two-letter country code for this unit?
[Unknown]: FR
Is CN=Hugo Decouty, OU=ULR, O=ULR, L=La Rochelle, ST=France, C=FR correct?
[no]: yes
```

À partir de là, afin de générer le certificat, il a fallu installer un logiciel KeyStore Explorer qui va permettre la réalisation d'un certificat .cerf (celui dans le dossier) à partir de la paire de clés conçue à l'aide de ce même logiciel.

Dans le navigateur Chrome, il faut donc aller dans les paramètres de sécurité et de confidentialité du navigateur, modifier les certificats racine de confiance et ajouter le certificat généré.

Nous avons utilisé un lien externe pour réaliser la partie HTTPS du serveur, le lien du code est le suivant : <https://stackoverflow.com/questions/18787419/ssl-socket-connection>

La conception de l'exécutable a été réalisée à l'aide de l'IDE IntelliJ IDEA. Le dossier META-INF est essentiel pour que le .jar fonctionne. Cependant, le .jar peut très bien être lancé sans avoir besoin d'IDE.