

Objectifs

- Réaliser l'itération 6 de l'application Gestion Log
 - Envoi des informations de log SSH sur un serveur centralisé
 - Valider les exigences du CDC (REQ005 ET REQ006)

Contenu technique

- Utilisation d'une API REST
- Formatage Json

Durée 6h

I. Installation du Serveur centralisé de logs

Vérifier que docker et docker compose soient installés sur votre poste

```
seb@Sb-portable:~$ docker -v
Docker version 24.0.7, build afdd53b
seb@Sb-portable:~$ docker compose version
Docker Compose version v2.21.0
```

Si ce n'est pas le cas, suivre le point 1.1

I.1 Docker

I.1.1 Installation docker

<https://docs.docker.com/engine/install/debian/#install-using-the-repository>

- Utiliser la méthode « apt »

I.1.2 Installation docker compose

<https://docs.docker.com/compose/install/#scenario-two-install-the-compose-plugin>

- Utiliser « Using Docker's repository »

I.1.3 Post installation step

<https://docs.docker.com/engine/install/linux-postinstall/>



I.2 Serveur centralisé

Un container Docker est disponible à cette adresse pour simuler le serveur centralisé de logs

<https://forge.apps.education.fr/gestionlogged/srv-flash.git>

Tâche1. Suivez les instructions du README pour configurer et exécuter le container.

II. Vérification de l'API POST du serveur centralisé

Q1. Donner la liste des routes définies dans l'API

Q2. Donner la définition et le rôle d'une requête HTTP GET

Q3. Donner la définition et le rôle d'une requête HTTP POST

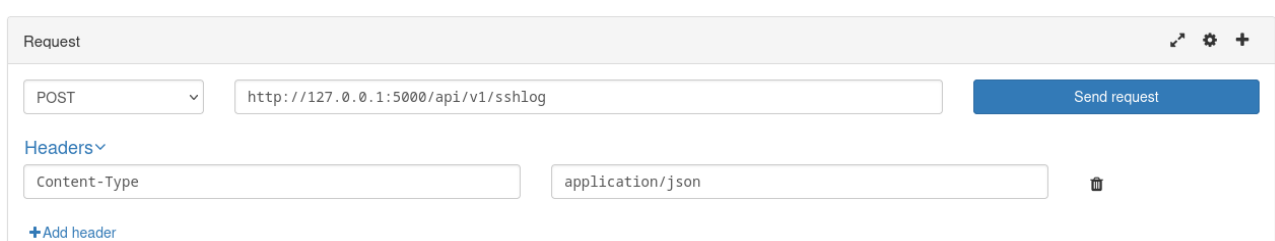
Tâche2. En utilisant Firefox, tester les différentes routes GET du serveur

.

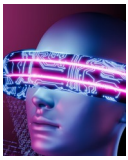
Tâche3. En utilisant le plugin [firefox RESTED](#), tester les 3 routes utilisant le protocole HTTP GET

Q4. Donner les éléments de chaque entrée de log dans votre compte-rendu (screenshot de la réponse RESTED)

Tâche4. En utilisant le plugin firefox RESTED, tester la route utilisant le protocole HTTP POST (pour celle-ci n'oubliez pas de préciser le **Content-Type application/json** dans le header)



Q5. Donner les éléments de la réponse du serveur dans votre compte-rendu (screenshot de la réponse RESTED)



III. Utiliser Poco

Q6. Rechercher et donner le rôle de la librairie C++ POCO

III.1 Dépôt PocoTest

Un exemple d'utilisation de la librairie Poco est proposée sur ce dépôt

<https://forge.apps.education.fr/gestionlogsed/pocotest.git>

Tâche5. Cloner le dépôt et ouvrir le projet PocoTest

III.1.1 Fonction PocoGet ()

Étude de la fonction PocoGet ()

Q7. Quel est son rôle ?

Q8. Donner la définition d'une URI

Q9. Que réalise la méthode `Poco::Net::HTTPRequest` ? Est-ce une requête GET ou POST ?

Q10. Que réalise la méthode `HTTPClientSession` ?

Q11. Tester la méthode pour afficher le contenu de la route `/api/v1/sshlog/all`

Q12. Tester la méthode pour afficher le contenu de la route `/api/v1/sshlog?id=x`

III.1.2 Fonction PocoPost ()

Étude de la fonction PocoPost ()

Q13. Quel est son rôle ?

Q14. Que réalise la méthode `Poco::Net::HTTPRequest` ? Est-ce une requête GET ou POST ?

Q15. Quel est le rôle de `Poco::JSON::Object obj` ?

Q16. Grâce à quelle méthode de `Poco::JSON::Object` peut-on ajouter des éléments à `Poco::JSON::Object` ?



IV. Intégration de la librairie POCO à votre projet

Après cette étude préalable, il est temps d'intégrer la librairie Poco à votre projet GestionLog

IV.1 Git

N'oubliez pas de vous replacer sur la branche **dev** pour cette itération.

IV.2 Ajout de la librairie Poco

Tâche6. Étudier le `CMakeList.txt` du projet Pocotest et intégrer la librairie Poco à votre projet

IV.3 PocoGetAll()

Tâche7. En vous référant au DDC, ajouter la fonction `PocoGetAll()` à votre projet pour l'appeler lors de la sélection de l'option du menu correspondante (3-pocoGetAllLog)

IV.4 PocoGetLogId()

Tâche8. En vous référant au DDC, ajouter la fonction `pocoGetLogId()` à votre projet pour l'appeler lors de la sélection de l'option du menu correspondante (4-pocoGetOneLog)

IV.5 PocoPostSshLog()

Objectif : La fonction `PocoPostSshLog()` est appelée via le menu pour envoyer sur le serveur les logs enregistrés dans le fichier `ssh_connexion.txt`

Tâche9. Dans un premier temps, ajouter l'appel à `PocoPostSshLog()` depuis la fonction `sshLog()`. En effet, c'est l'option la plus facile pour créer l'objet json à envoyer sur le serveur



IV.6 LireEtEnvoyerFichierConnexionSSH()

Tâche10. En vous référant au DDC, ajouter la fonction

`LireEtEnvoyerFichierConnexionSSH()` à votre projet pour l'appeler lors de la sélection de l'option du menu correspondante (5-Poco Post SSH Log)

Résultats attendus :

- ✓ La fonction `PocoGetAll()` récupère tous les logs enregistrés sur le serveur centralisé.
- ✓ La fonction `pocoGetLogId()` récupère en fonction de son id un log enregistré sur le serveur centralisé.
- ✓ La fonction `PocoPostSshLog()` envoie les logs sur le serveur centralisé.
- ✓ La fonction `LireEtEnvoyerFichierConnexionSSH()` envoie les logs du fichier `ssh_connexion.txt` sur le serveur centralisé.

IV.7 Validation

Appeler l'enseignant pour qu'il valide votre travail

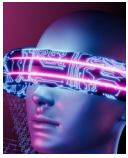
V. Améliorations

Grâce à ce projet, vous avez créé une application qui permet de lister, filtrer et enregistrer des logs importants sur votre PC et également de les transmettre à un serveur centralisé.

Cependant, cette application n'est pas parfaite.

Pour conclure ce projet, le RSSI vous demande un retour d'expérience pour améliorer son produit.

Q17. Critiquer l'application notamment en listant les pistes d'améliorations que vous avez pu relever à réalisant les différentes itérations.



VI. Dossier de validation

Vérifier que votre application répond aux exigences du cahier des charges.

Q18. Compléter le dossier de vérification Gestion_Log_DDV.odt (ESS004)

VII. Outils DevOps

Afin de finaliser votre travail, n'oubliez pas :

- D'indenter correctement votre code
- De documenter le code et les fonctions
- de fusionner votre branche dev avec master

VIII. Livrable

Sur le Moodle de la section

- Votre compte rendu de TP avec les réponses aux questions de celui-ci
- Le dossier de validation DDV complété (ESS04)

Sur le serveur GIT de la section

- Le projet Gestion_Log à jour
- Les branches dev et master
- Le tag de la version v6.0