



---

INF00503 - Compte Rendu projet 3

Cyril Rabat - Jean-Charles Boisson

# Les usines du Père Noël

## GIGOUT Thomas - DAUNIQUE Wilfried

---



### Introduction

Nous souhaitons développer une application qui aura pour but d'aider le Père Noël à gérer ses usines et les commandes des enfants, car, à cause de l'accroissement de la population humaine et la diversification des cadeaux, il a lui a fallu créer plusieurs usines réparties dans le monde entier, et non une unique usine comme on peut le croire. Cette application distribuée sera constituée d'applications Java et PHP. Tout échange entre des applications Java sera chiffré et déchiffré à l'aide d'une clé symétrique (AES) initialement créée à l'aide du chiffrement RSA.

---

---

## Fonctionnement

### Portail PHP

Le portail PHP se compose de deux pages. Une de connexion/inscription, et une d'accueil après avoir validé la précédente. Une fois connecté, on récupère la liste des usines stockées dans le serveur BackOffice et on peut en sélectionner une. On voit alors la liste des articles créés dans l'usine. On peut ensuite sélectionner chaque article pour voir le stock actuel, lancer la création de nouveaux articles etc. On utilise Ajax pour changer dynamiquement les selecteurs.

### Serveur d'authentification

Le serveur d'authentification est écrit en Java et gère les clients avec leurs droits : vérification du couple login / mot de passe, création de compte, récupération d'informations du compte etc. Il attend des requêtes HTTP.

### Serveur BackOffice

Le serveur BackOffice permet de faire le lien entre le portail PHP et les usines. Chaque usine, à son démarrage, s'enregistre sur le backoffice qui possède ainsi la liste des usines en mémoire. Il sert également de lien lorsque le portail envoie des requêtes comme la récupération des produits d'une usine, le lancement d'une commande etc. Il reçoit lui aussi des requêtes HTTP.

### Les usines

Les usines reçoivent également des requêtes HTTP. Le programme, identique pour chaque usine, récupère les produits, stocks et commandes depuis des fichiers JSON.

### L'autorité tierce

Le Père Noël, de par la criticité de ses activités, a besoin de sa propre autorité. Elle utilise le protocole TCP.

### Conditions

Toutes les applications Java doivent avoir un certificat. Toute communication entre deux applications Java doit être sécurisée : récupération et vérification du certificat de son interlocuteur. Les communications entre le BackOffice et les Usines sont chiffrées en utilisant AES. Toutes les communications utilisent le format JSON, sauf la phase de connexion entre le portail et le navigateur. Toutes les applications Java utilisent des fichiers de configuration, et toutes ces informations (configs, articles, etc.) sont stockées dans des fichiers JSON.

---

## Travaux réalisés

### Application PHP

Nous avons mis en place une page **index.php** qui gère la connexion ou l'inscription d'une personne. Si vous êtes loggé en tant que login : Santa, mdp : SuperSanta, vous avez accès au reste du site qui n'est autre qu'une page d'accueil contenant des selects censés être dynamique grâce à Ajax. Nous avons pour cela mis en place une session lors de la connexion suite à une vérification grâce à un LoginHandler.java et un GestionnaireUtilisateurs. Tous les utilisateurs sont stockés dans un fichier json "users.json".

### Serveur d'authentification

Nous avons pu développer et tester le bon fonctionnement de notre serveur d'authentification, qui se compose d'un serveur et de deux handlers (LoginHandler, CreateHandler) ainsi que d'un GestionnaireUtilisateurs.

### Serveur BackOffice

Nous avons pu créer le BackOffice avec deux handlers aussi (BackOfficeHandler et RecupUsineHandler). BackOfficeHandler servira à enregistrer les informations de l'usine sur le serveur BackOffice (ArrayList contenant les usines). Nous n'arrivons cependant pas à créer une usine à partir des informations que nous recevons (*erreur : Unexpected end of file from server*) Nous pensons que cela vient de l'objet Produit contenu dans la Commande en JSON. Du coup, la partie récupération des usines pour l'Ajex n'est pas fonctionnelle, mais il ne manque pas grand chose pour que ça le soit.

### Sécurité : clé et certificats

Les clés et certificats sont non fonctionnels. Nous avons tout de même tenté de développer certaines parties des clés et certificats, nous avons donc un fichier GestionClesRSA qui permet de créer, enregistrer et lire la clé publique et privée. Nous avons un fichier Chiffrement qui est en fait le chiffrement AES. Nous avons aussi une ébauche de ce que pourrait être le certificat.java.

### Usines

Nous pouvons créer des usines, malheureusement nous nous sommes aperçu trop tard que nous ne créons pas les informations de chaque usine. Pour tester l'application, il faut donc rentrer en dur les informations dans usines/nom\_usine.json...

---

## Format des fichiers JSON

### 1. Configuration des usines

```
{  
  "port": 5001,  
  "adresse": "127.0.0.1",  
  "nom": "usine1"  
}
```

### 2. Utilisateurs (users.json)

```
{  
  "users": [{  
    "mdp": "SuperSanta",  
    "id": 1000,  
    "pseudo": "Santa",  
    "email": "supersanta@northpole.com"  
  }]  
}
```

### 3. Informations des usines

```
{  
  "nom": "usine1",  
  "produits": [{  
    "id": 1,  
    "nom": "Lego",  
    "stock": 2  
  }, {  
    "id": 2,  
    "nom": "Peluche",  
    "stock": 3  
  }  
],  
  "commandes": [{  
    "id": 1,  
    "produit": {  
      "id": 1,  
      "nom": "Lego",  
      "stock": 1  
    },  
    "qte": 10  
  }]  
}
```

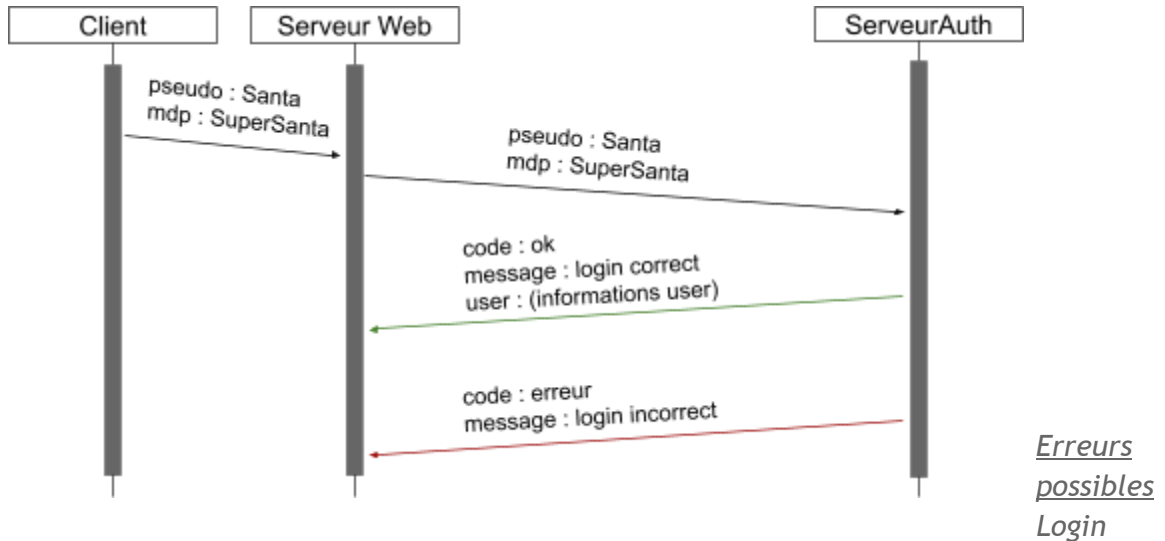
---

## Arborescence des fichiers

```
|_ Noel_project/
  |_ docs/
    |_ rapport503.pdf
    |_ rapport504.pdf
  |_ java/
    |_ Classes/
    |_ CleClient/
    |_ lib/
    |_ src/
      |_ BackOffice.java
      |_ BackOfficeHandler.java
      |_ Certificat.java
      |_ ChallengeHandler.java
      |_ Chiffrement.java
      |_ Commande.java
      |_ Config.java
      |_ CreateHandler.java
      |_ GestionClesRSA.java
      |_ GestionnaireProduits.java
      |_ GestionnaireUsines.java
      |_ GestionnaireUtilisateurs.java
      |_ LoginHandler.java
      |_ Produit.java
      |_ RecupUsineHandler.java
      |_ ServeurAuth.java
      |_ Usine.java
      |_ Utilisateur.java
  |_ json/
    |_ Config/
      |_ usine_nom.json
    |_ users/
      |_ users.json
    |_ usines/
      |_ usine_nom.json
  |_ php/
    |_ accueil.php
    |_ connexion.php
    |_ index.php
    |_ js/
    |_ css/
    |_ img/
```

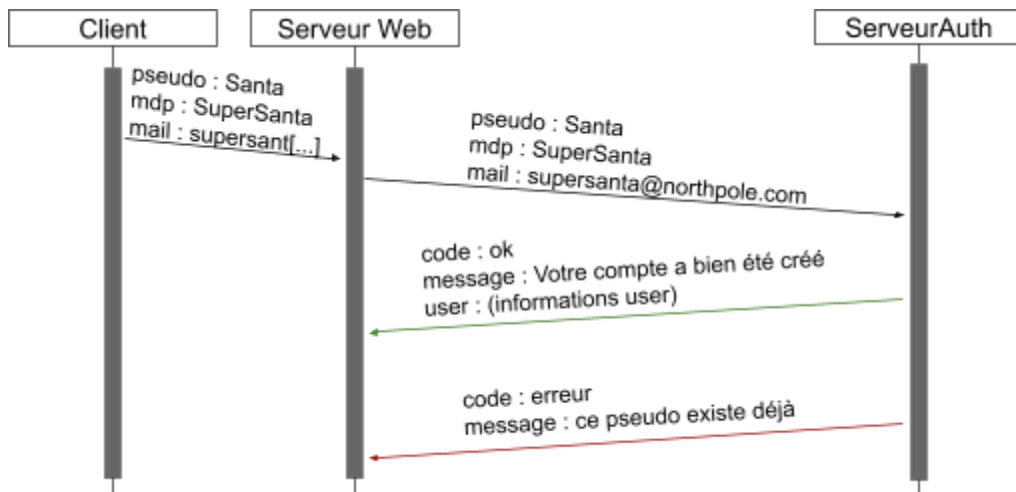
## Chronogrammes d'échanges

### Connexion



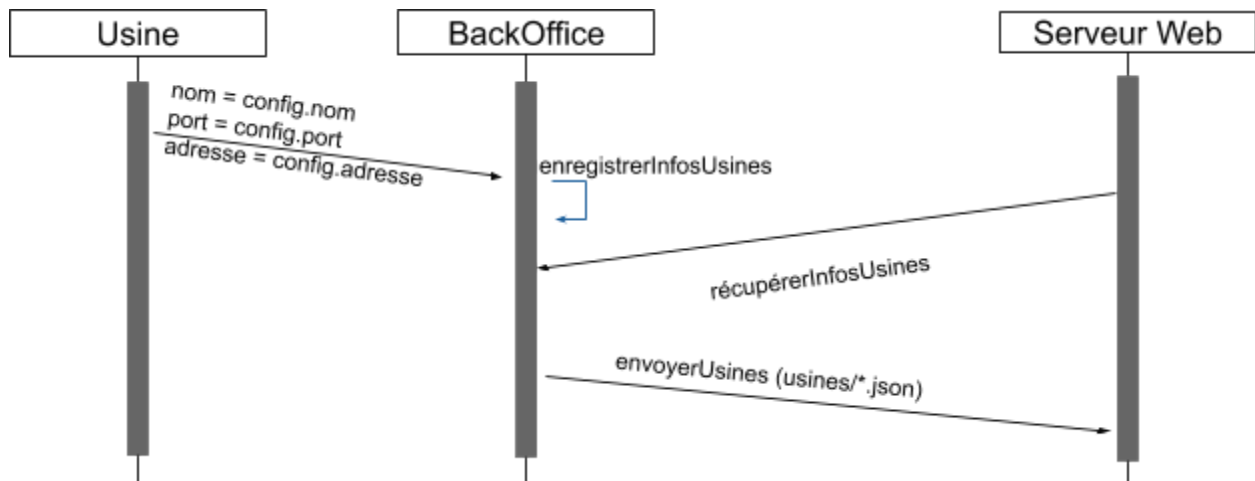
:  
incorrect (selon le fichier users.json), mot de passe incorrect (de même) , serveur d'authentification éteint.

### Inscription



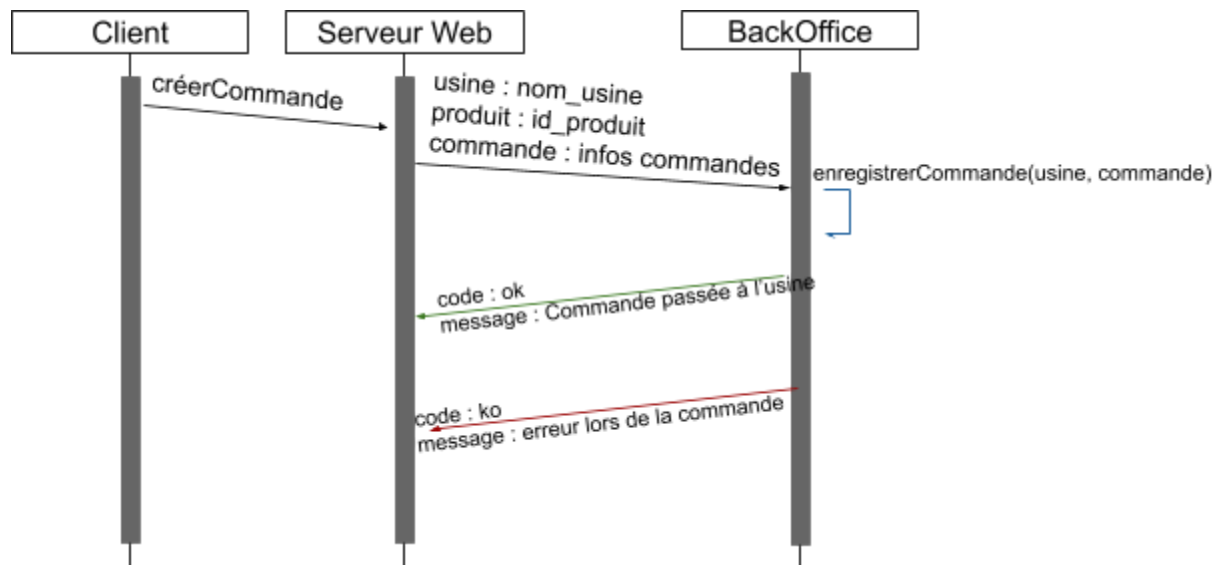
*Erreurs possibles* : Serveur d'authentification éteint, login déjà existant.

## Récupération des usines par le BackOffice



Erreurs possibles : Serveur BackOffice éteint.

## Création d'une commande



Erreurs possibles : Serveur BackOffice éteint, mauvais chemin concernant les usines.json,

## Échange de certificat



Erreurs possibles : Certificat invalide.

## Exécution

Se placer dans le répertoire `java/` du projet, puis compiler avec :

```
javac -d .\classes\ -sourcepath .\src\ -cp .\lib\json-20190722.jar .\src\*.java
```

### 1. Lancer le serveur d'authentification

```
java -cp ".\classes\;.\lib\json-20190722.jar" ServeurAuth
```

### 2. Lancer le serveur BackOffice

```
java -cp ".\classes\;.\lib\json-20190722.jar" BackOffice
```

### 3. Lancer l'usine

```
java -cp ".\classes\;.\lib\json-20190722.jar" Usine
Nom de l'usine ? : (il faut entrer "usine1" pour tester convenablement)
usine1
```

Vous pourrez voir que l'on reçoit les bonnes informations dans les différents serveur, mais la fonctionnalité n'est pas fonctionnelle à 100%...

Par contre vous pourrez, dès le moment où le serveur d'authentification est lancé, vous rendre sur le localhost de l'application, et vous inscrire et vous connecter.

Si vous souhaitez vous connecter en tant que Père Noël

Login : Santa | Mot de passe : SuperSanta