

# SER - Labo4

Yann Lederrey & Joel Schär

June 9, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Fonctionnement Générale</b>	<b>3</b>
2.1	WFC . . . . .	3
2.1.1	Conception . . . . .	3
2.2	PlexAdmin . . . . .	4
2.2.1	Conception . . . . .	4
2.3	PlexMedia . . . . .	5
2.3.1	Conception . . . . .	5
2.4	Utilisation . . . . .	6
<b>3</b>	<b>conclusion</b>	<b>7</b>

# 1 Introduction

Ce rapport détailles les opérations demandé au laboratoire 4 de Sérialisation. Le but était d'échanger des informations entre plusieurs applicatifs en utilisant la méthode RMI.

Premièrement il fallait permettre l'échange de l'état actuel de la base de données entre l'application WFC (contenant la base de données) et l'application générale PlexAdmin.

Ensuite il fallait permettre à l'application PlexMedia de récupérer le fichier Json contenant la liste des projections cela en le demandant à PlexAdmin.

# 2 Fonctionnement Générale

Ici je vais vous expliquer le fonctionnement générale et l'interaction entre les applications dans le cadre du laboratoire 4. Seulement des actions en rapport avec le laboratoire 4 seront montrées.

Premièrement il faut savoir qu'il est nécessaire de démarrer les applicatifs dans l'ordre suivant:

- WFC
- PlexAdmin
- PlexMedia

L'ordre doit être respecté car certaines application ont un rôle de serveur RMI ou Client ou les deux et bien-sur les serveurs doivent être démarré avant leur client respectif.

Aucun fichier Json/XML n'ont été modifiés, seulement les fichier java ont été modifiés.

## 2.1 WFC

Le seul but de WFC est d'aller récupérer des données dans une base de données se trouvant à l'HEIG-VD et de retransmettre ces informations à PlexAdmin. Afin d'effectuer au mieux la transmission d'informations. PlexAdmin est observateur de WFC. C'est à dire que lorsque WFC annonce un changement (il a récupéré de nouvelles informations provenant de la base de données) il va alors notifier PlexAdmin pour que celui-ci invoque un update des données qui transite entre les deux applications.

### 2.1.1 Conception

Les classes modifiées sur WFC sont les suivantes :

- Main.java

Le code des classes modifiées ont été rendues avec le rapport.

Si je résume ce qu'il se passe dans le main par rapport à RMI :

- Premièrement je crée un Registry sur le port 9999, le crée un service et je le bind avec le nom “RmiService”.
- Ensuite dans le méthode Run qui va tourner en fond j’ajoute les méthodes permettant de signaler une mise à jour des données via setChanged et notifyObservers.
- j’implémente les méthodes nécessaires au modèle Observer/Observé et à la mise à jour des informations.

Pour résumé lorsque PlexAdmin se connecte il communique à WFC son existence, celui-ci va alors l’ajouter dans sa liste d’observable. Ensuite lorsque WFC notifie PlexAdmin d’une mise à jour ce dernier va invoquer la méthode partagée getData afin de récupérer les données.

## 2.2 PlexAdmin

Pour ce laboratoire PlexAdmin a le rôle de récupérer les informations en provenance de WFC et de mettre à jour ça liste de films etc...

Après avoir définis une liste de projections, et généré un Json contenant cette liste, PlexAdmin doit permettre au client PlexMedia de récupérer ce fichier Json via RMI.

### 2.2.1 Conception

Les classes modifiées sur PlexAdmin sont les suivantes :

- ControleurGeneral.java
- ControleurWFC.java
- ControleurMedia.java
- WFC-Interface/rmi/IserverMediaApi

Le code des classes modifiées ont été rendues avec le rapport.

Si je résume ce qu’il se passe pour la partie WFC-PlexAdmin.

Ici PlexAdmin doit recevoir la notification de WFC lorsque celui-ci a mise à jour ses informations, récupérer les nouvelles données puis mettre à jour ses propres informations.

- Premièrement dans le ControleurGeneral je vais définir quel serveur RMI PlexAdmin doit écouter. Pour cela je définis un remoteService sur le port 9999 en localhost/RmiService.
- Ensuite dans la classe ControleurWFC je définis les méthodes nécessaires au modèle Observer/Observé ainsi que les méthodes des Interface.

Si je résume ce qu’il se passe pour la partie PlexAdmin-PlexMedia.

Ici PlexAdmin prend le rôle de serveur et PlexMedia de client. PlexAdmin doit alors fournir à PlexMedia la méthode nécessaire à récupérer un payload json de projections.

- Premièrement dans le ControleurMedia je vais définir le serveur RMI en créant une nouvelle registrie sur le port 9998. que je bind sur “Rmimedia”.
- Ensuite je définis les méthodes nécessaires au modèle Observer/Observé et les méthodes des Interface en communs permettant la récupération des informations.
- la méthode sendJSONToMedia crée le Json et notifie PlexMedia que celui-ci a été mis à jour. La méthode getData permet de récupérer le Json en format String.

Pour résumé lorsque PlexAdmin se connecte il communique à WFC son existence et démarre un thread “startCheckingThread” qui va vérifier continuellement que le serveur est joignable WFC va ensuite l’ajouter dans sa liste d’observable. Lorsque la connexion est initialisée PlexAdmin demande son ajout dans les Observable. Lorsque PlexAdmin reçoit un signal (une notification) il demande à récupérer les données et met à jour ses informations. (méthode update). Ensuite lorsque PlexAdmin a crée une liste de projections et généré le Json cela notifie PlexMedia qui peut alors mettre à jour ça liste en appelant la méthode commune getData et reçoit alors un payload Json en format String contenant les projections.

## 2.3 PlexMedia

PlexMedia a pour but de récupérer un Json contenant la liste des projections depuis PlexAdmin, Pour cela un bouton update a été crée permettant de faire cette récupération.

### 2.3.1 Conception

Les classes modifiées sur WFC sont les suivantes :

- Amorce.java
- mainController
- WFC-Interface/rmi/IserverMediaApi

Le code des classes modifiées ont été rendues avec le rapport.

Si je résume ce qu’il se passe pour PlexMedia :

- Premièrement dans l’Amorce je démarre l’Interface graphique et par la même occasion le mainController.
- Ensuite dans la méthode initialize du mainController je mets en écoute PlexMedia sur le port 9998 en localhost/Rmimedia
- Puis finalement j’implémente et utilise les méthode permettant de récupérer le fichier Json à travers RMI.

Pour résumer lorsque l’on clique sur le bouton update PlexMedia va invoquer la méthode commune provenant de l’Interface IserverMediaApi permettant de récupérer le payload Json de PlexAdmin. Puis va afficher son contenant en terminal.

## 2.4 Utilisation

- État de PlexAdmin avant la mise à jour de ses données :

id	Titre	Duree	Photo	Note	Show	Edit	Critique
2053403	Casino Royale	144		-			
2235127	GoldenEye	130		-			
2576878	Quantum of Solace	106		-			
2652003	Skyfall	143		-			
2806844	The World Is Not Enough	128		-			
2822603	Tomorrow Never Dies	119		-			

- Mise a jour des données sur WFC :

```
Select the data version to download [1/2/3/0=quit]: 3
Starting dataset 3...
Done dataset 3 [29.5 s]
Select the data version to download [1/2/3/0=quit]:
```

- État de PlexAdmin après la mise à jour des données :

id	Titre	Duree	Photo	Note	Show	Edit	Critique
2053403	Casino Royale	144		-			
2136717	Django Unchained	165		-			
2207917	Forrest Gump	142		-			
2235127	GoldenEye	130		-			
2239795	Gravity	91		-			
2304289	Inglourious Basterds	153		-			
2433059	Man of Steel	143		-			
2576878	Quantum of Solace	106		-			
2652003	Skyfall	143		-			
2728991	The Da Vinci Code	149		-			
2768695	The Monuments Men	118		-			
2806844	The World Is Not Enough	128		-			

- Création de projections sur PlexAdmin :

Projections		Films					
id	DateHeure	Film	Salle	Duree	Show	Edit	Delete
16...	09-06-2018 08:45	Casino Royale	Flon 1	144			
16...	09-06-2018 08:45	GoldenEye	Flon 1	130			
16...	09-06-2018 08:45	Man of Steel	Flon 1	143			

- Création du fichier Json sur PlexAdmin :

[JSON > Media](#)

- demande de mise à jour sur PlexMedia :

[UPDATE](#)

- nouveau payload Json pour PlexMedia :

```
{Projections:[{TitreFilm=Casino Royale, DateProjection=09-45-2018, Acteurs={1er Role=Craig, Daniel, 2eme Role=Green, Eva}},
```

### **3 conclusion**

Ce laboratoire a été intéressant et nous a fait découvrir une nouvelle technologie. Je ne suis pas sûr de souvent réutiliser RMI mais il est toujours instructif de connaître de nouvelles