

Enterprise Application Integration

Practical Assignment 3

João Ferreira & João Silva

Departamento de Engenharia Informática

Universidade de Coimbra

`jpbat@student.dei.uc.pt` | `jfmsilva@student.dei.uc.pt`

2009113274 | 2008111448

December 2013

Contents

1	Introduction	3
2	Data Model	3
3	Flows	4
3.1	Movie Subscription	4
3.2	Movie Manage	7
3.3	Movie Stats	9
4	Division of Work	10
5	Conclusion	11

1 Introduction

In this practical assignment it was requested that we reused the first one (just the crawler) to generate a xml file. Only one adjustment had to be made, so that the rate of the movie was also fetched.

We also needed to send emails to subscription clients and post some tweets. The twitter account used can be found in @IMDbCrawler.

We'll now briefly explain some decisions that were taken and how we solved some issues that appeared along the way.

2 Data Model

As it is obvious not all the information that is obtained by the crawler is needed, so we decided to change our database desing.

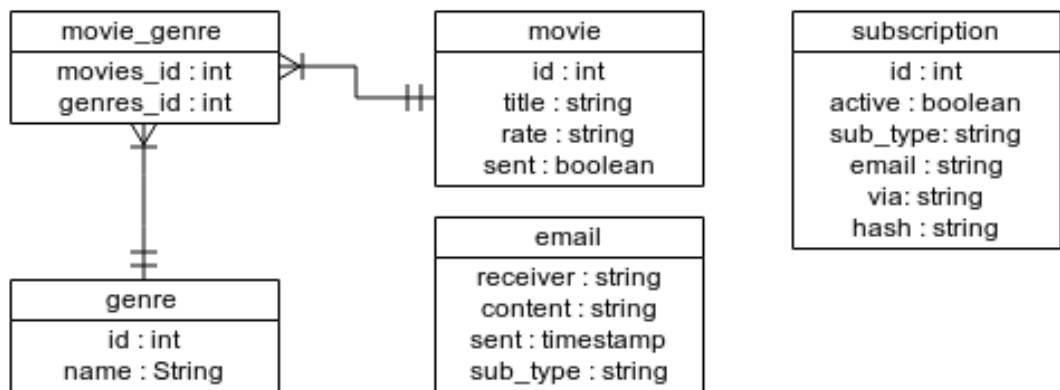


Figure 1: ER Model.

3 Flows

We created three separated flows files, being the first to movie subscription, the second to movie management and the last retrieves the stats.

3.1 Movie Subscription

In this flow we added the actions to subscribe/unsubscribe to the service. Either by SOAP or via Web, the first step is always to save the data into Session Variables so that we can develop a more dynamic application.

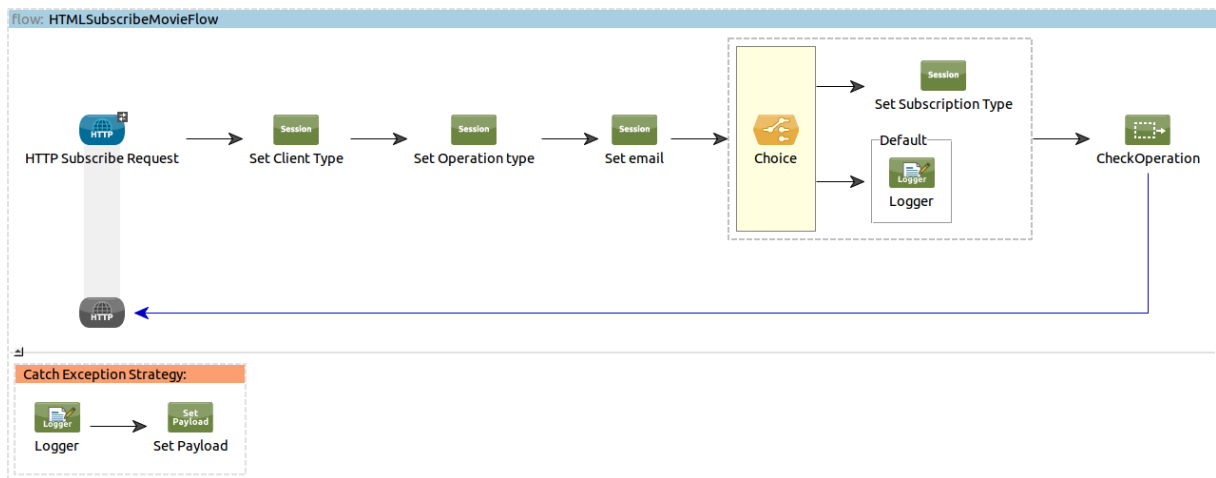


Figure 2: Data extraction on WEB

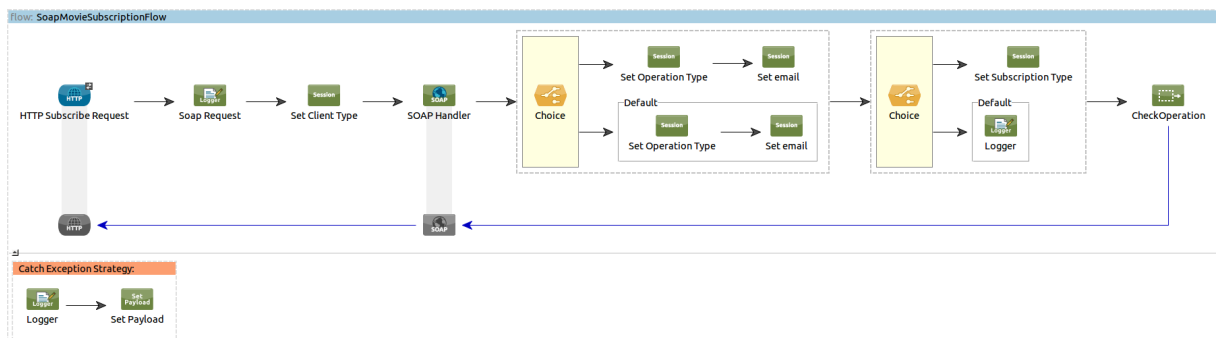


Figure 3: Data extraction on SOAP

After this the application chooses if it should perform a subscription or an unsubscription.

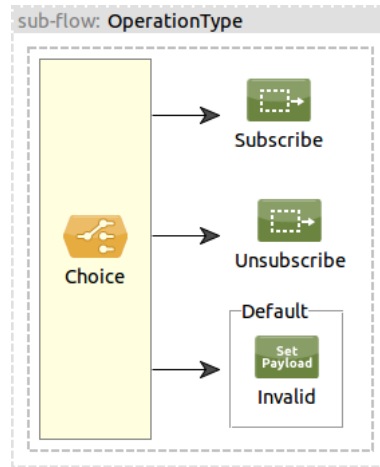


Figure 4: Choosing the type of operation

Depending on the operation the application will perform one of the two following flows.

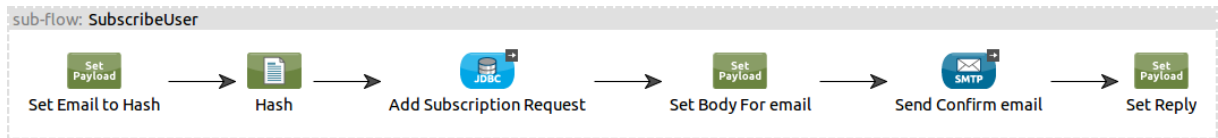


Figure 5: Subscribing a user

On the first we start by calculating an hash that is going to be used to activate the subscription. after that we insert all that information into the database. Finnnaly we send an email with the link to the activation of said subscription.

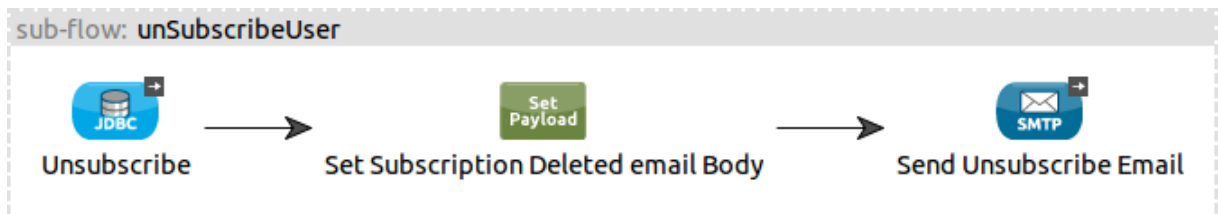


Figure 6: Unsubscribing a user

On the second we delete from the database the subscription which contained the provided email. After that we send an email confirming the deactivation of the subscription.

By last we have a flow that changes the status of the subscription when the user follows the link provided in the email that was sent.

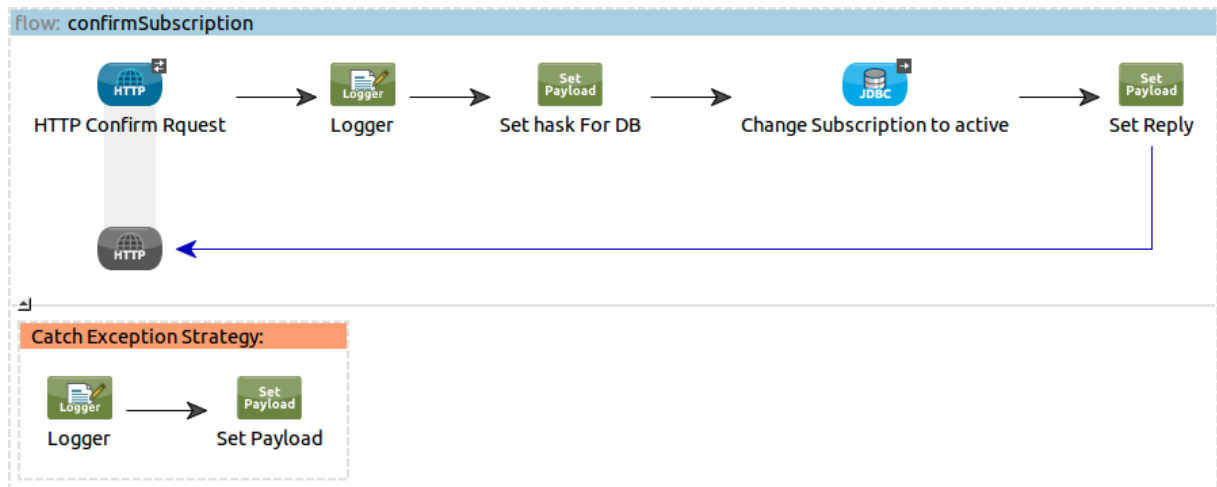


Figure 7: Confirming Subscription

3.2 Movie Manage

In this flow we have three main flows.

The first two are to add movies either via putting a file in the trigger folder, or by using the SOAP client which is integrated in the crawler from the first project.

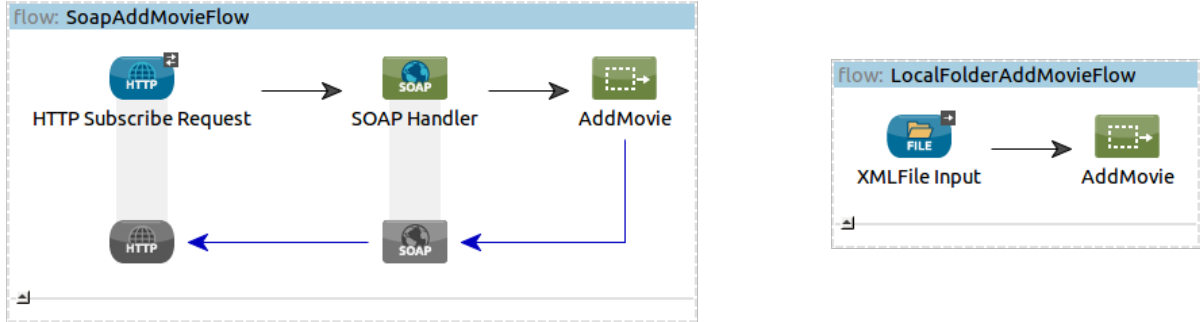


Figure 8: Both flows to add movies

After this we jump into a subflow that is responsible for storing the movie into the database, sending tweets and also sending the digest email as well as storing the information about the sent email into the database so that we can produce stats. The conversion of the XML string into object is done by using JAXB generated classes from the first project.

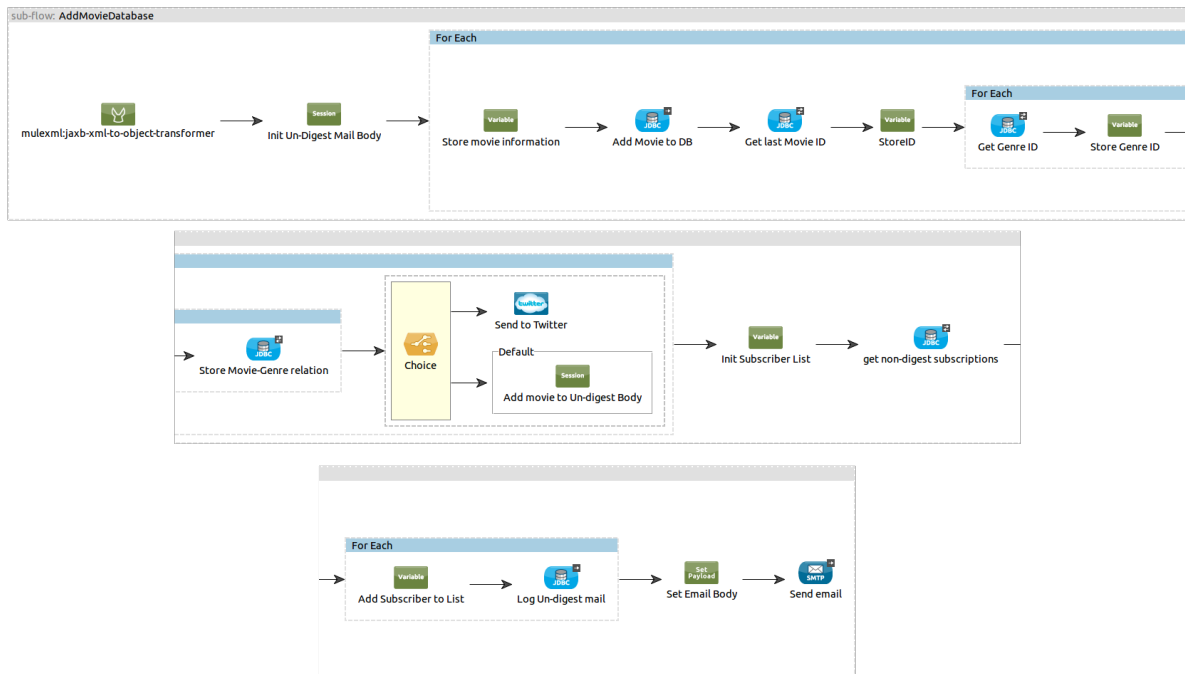


Figure 9: Add movie to database

The third flow is the one that is responsible by sending the digest emails, based on a quartz event dispatcher.

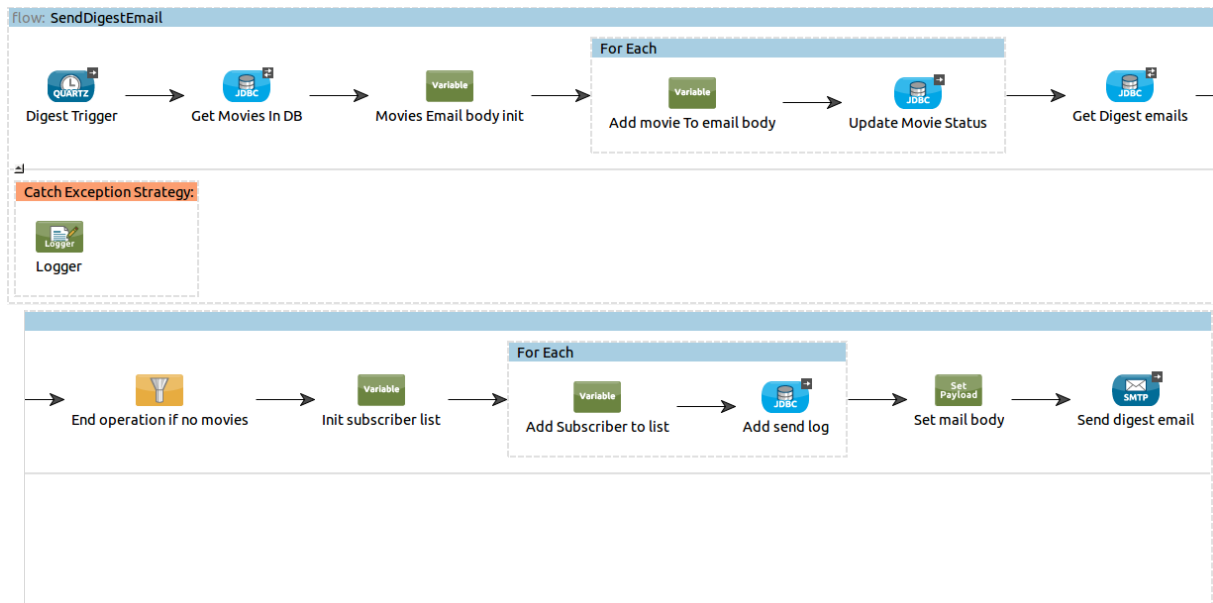


Figure 10: Send digest emails

3.3 Movie Stats

This is by far the simplest flow, once all it does is queries to the database to retrieve the information based on the selected option.

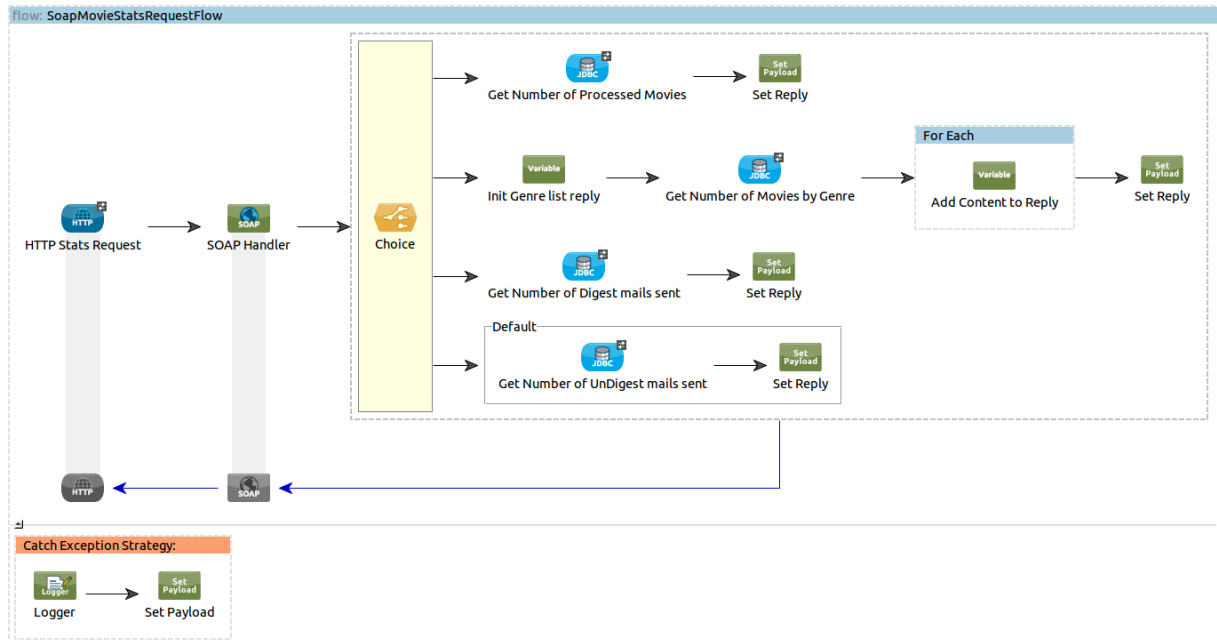


Figure 11: Stats Producer

4 Division of Work

The work was, in what appeared to us and as far as possible, equally divided.

Unfortunately we were unable to make a clear separation of the work. This happened because we couldn't develop at the same time, once the Mule Studio kept giving build issues. We even tried to add to our git repository the .gitignore that they recommended, which turned out to be a lost of time.

Also if we divided the project too much, one of us could easily lose track of the development state.

Ence this was the best division we could make.

João Ferreira Mule Flows, Database modeling. (50h)

João Silva Mule Flows, SOAP clients. (50h)

5 Conclusion

With this project we learned how to use a SOAP client/server.

Besides that we don't think that Mule Studio wasn't a good tool to use, once we crossed some bugs on it. For instance the returning value on an insert query is not supported, therefore obligating us to do yet another query, hence overloading the system. We also had some Twitter issues, to which the solution was remove the connector and add it again, we even tried to make the diff of the xml after that step and it was equal.

Maybe in the future the Mule Studio will be a good solution, for now we think its mediocre, once there are some bugs as it was referred and there could be more and better documentation.

Although all of this we consider that the execution of this assignment was a success, for which both of us learned some important aspects of the importance of using tools to integrate several services.