

 <p>FCTUC UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA <i>Departamento de Engenharia Informática</i></p>	<p align="center">Project #3 Integração de Sistemas/ Enterprise Application Integration</p> <p align="center">2013/14 – 1st Semester MEI Deadline: 2012-12-16 8:00AM</p>
<p>Nota: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional. Qualquer tentativa de fraude pode levar à reprovação na disciplina tanto do facilitador como do prevaricador.</p>	

Application Integration with an Enterprise Service Bus

Objectives

- Learn how to use a real and very popular Enterprise Service Bus
-

Final Delivery

- All source code for the project.
 - A PDF report (5 pages recommended max size) about the implementation of the project, which must include and explain all diagram Flows created. The report must specify the number of hours spent per student while working on the assignment.
-

Software

- **Mule**, an open-source and very popular ESB. The easiest way to start using Mule ESB is to download the **Mule Studio** editor. Browse to <http://www.mulesoft.com/> and then to products > development > Mule Studio. This editor is eclipse-based and most of you will find it quite familiar.
-

References

- **Fundamental reference:**
 - <http://www.mulesoft.org/documentation/display/current/Home>

Other references you may want to look at:

- MEL reference card:
 - <http://blogs.mulesoft.org/wp-content/uploads/2012/12/refcard-mel.pdf>
 - Generic information about creating and using a web service:
 - <http://eai-course.blogspot.pt/2013/04/web-service-with-jax-ws-224.html>
 - Using a database:
 - <http://www.dotnetfunda.com/articles/show/2068/using-mule-studio-to-read-data-from-postgresqlinbound-and-write-it-to>
 - Sending Email:
-

- <http://soatechlab.blogspot.pt/2010/07/sending-outbound-smtp-messages-through.html>
 - Using Twitter:
 - <http://java.dzone.com/articles/mule-school-integration-social>
 - Using JAXB Transformers:
 - <http://www.mulesoft.org/documentation/display/current/JAXB+Transformers>
-

Mule ESB Training Part (doesn't count for Evaluation)

1. Create a Flow that is triggered by an HTTP request sent to *http://localhost:8081/hello* and echoes the message and its payload. Now add an HTTP parameter to the request and use Mule's *set-payload* Transformer to change the response message (use the HTTP parameter value to build the response).
 2. Build a SOAP web service that receives a String as input and returns another string. Create a client for it.
 3. Create a Flow that can read from a database.
 4. Create a Flow that is able to send an Email.
-

Project 3 (for evaluation)

In this assignment students should create a set of services and connect and orchestrate them in Mule Flows. These flows will process movie information and perform several functions such as sending emails to subscribed users or producing statistics. It is the students' responsibility to identify, define and link the building blocks to create the overall system. The following paragraphs describe the system's functionality (please read the whole description before starting).

Subscription to movies information

Users can subscribe/unsubscribe to new movies information. This means they will receive email notifications about new movies entering the system. Subscribing/Unsubscribing can be made in the following two ways:

- Using a SOAP client (should provide a simple text user interface);
- From a static web page.

The subscription information must include:

- An email address;
- One of the following preferences for receiving email:
 - a) Non-digest Email: whenever a new movie is introduced in the system, the user is notified with a single email carrying information about that new movie;
 - b) Digest Email: The user will only receive periodic emails about new movies (e.g., once per week). Thus, each email can carry information about several new movies. The periodicity is defined by the system.

The email address should be verified and the subscription channel (Web or SOAP) must also be registered by the system.

Adding new Movies

There are two ways of adding new movies to the system:

- Adding an XML file (holding a list of movies, like the one produced in project 1) to a specific directory will trigger a processing flow.
- Sending an HTTP request for a SOAP web service (which accepts an XML String holding a list of movies). Students should also create the SOAP client to send the information.

From the list of new movies, only those with a Metascore rating should be processed. For this assignment we will assume that there are two target audiences (whose interests do not overlap). As such, after receiving a list of movies, the next step is to separate the movies for children (since currently there are not many G-rated upcoming movies, you can choose PG-rated movies for this purpose) from the remaining. For each children-safe movie processed, your system will post a message to Twitter announcing the new movie. Each of the remaining movies will be announced by email to subscribed users (and according to each user's preferences).

Statistics

The system should also provide a SOAP interface to display the following statistics:

- Total number of movies processed.
- Number of movies processed for each category.
- Number of digest emails sent.
- Number of non-digest emails sent.

Students should develop a simple SOAP client to display these statistics.

Finally, exceptions occurring in the system should be logged.

Good Work!