FCTUC
UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
Departamento de Engenharia Informática

Project #3
Integração de Sistemas/
Enterprise Application Integration

2015/16 – 1st Semester
MEI
Deadline: 2015-12-18

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.

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 of the project.
- A PDF report about the implementation of the project, which must include and explain all diagram flows created.
- After submitting, you are required to register the (extra-class) effort spent solving the assignment. This step is mandatory. Please fill the effort form at:
  https://docs.google.com/forms/d/1cC5_TozubwUNGgoDOZihyF9zzNaZfvEqeWi9_oHvP70/viewform

Software

- **Mule ESB**, an open–source and very popular Enterprise Service Bus. The easiest way to start using Mule ESB is to download the **Anypoint Studio** editor, which you can find at http://www.mulesoft.org/. This editor is eclipse-based and most students will find it quite familiar.

References

- **Fundamental reference:**
  - **https://docs.mulesoft.com**

Other references you may want to look at:

- MEL reference card:
  - http://blogs.mulesoft.com/wp-content/uploads/2012/12/refcard-mel.pdf
- The mule message, property scopes, and variables:
  - http://blogs.mulesoft.com/dev/anypoint-platform-dev/mule-school-the-mulemessage-property-scopes-and-variables/
- 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.mulesoft.org/documentation/display/current/Database+Connector
- Sending Email:
  - http://www.mulesoft.org/documentation/display/current/SMTP+Transport+Reference
  - If you don't manage to use the Email connector, you are allowed to use a script component for this purpose, e.g., a Groovy script.
- Using Twitter:
  - First register for a Twitter account and then go to https://apps.twitter.com to create a new application.
  - To properly integrate with twitter you need to install the new version of the connector in Help→ install new software and then in the filter text enter "Twitter". You must be careful to use the newer version of the connector in the flow. Otherwise, it won't work! (see this: http://forums.mulesoft.com/questions/2279/not-able-to-update-in-twitter-account.html)
- Using Java Transformers:
  - http://www.mulesoft.org/documentation/display/current/Java+Transformer+Reference
- 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.
5. Now do it at a given time of the day, using the Poll scope.
6. Create a Flow that updates the status of some account in Twitter.

Project 3 (for evaluation)

In this assignment students should create a set of services, connect and orchestrate them in Mule Flows. These flows will process information of smartphones 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.

Subscription to smartphones information

Users can subscribe/unsubscribe to smartphones information coming from the Pixmania (or other) site. In practice, this means they will receive email notifications about smartphones entering the system. Subscribing/Unsubscribing can be made in the following two ways:

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

The subscription information must include:

- An email address.
- The name of the client.
- A favorite brand in which the user is interested. The user will later receive email information about this brand's smartphones (and only this one).
- A range of prices in which the user is interested. Minimum and maximum thresholds define this range.

The email address should be verified (for subscribing and unsubscribing), by sending an email to the address in cause. The subscription channel (Web or SOAP) must be stored by the system.

Adding smartphones

There are two ways of adding *smartphones* to the system:

- Adding an XML file (like the one produced in Project 1) that contains a list of smartphones to a specific directory. This action will trigger a processing flow.
- Sending an HTTP request for a SOAP web service, which accepts a String holding the list of smartphones (in XML format). Students should also create the text-based SOAP client to send the information.

From the list of smartphones, only those, which have information of screen size and screens larger than 10 cm should be further processed. All processed smartphones will be announced by email to subscribed users, according to each user's preferences. This announcement to users will occur every day at midnight (to make this practical you should choose some other period).

Statistics

Whenever one of the following statistics changes, the system should send one (or more) Tweets (to Twitter) with the following information:

- Number of updates done to the list of smartphones in the database.
- Number of smartphones in the database.
- Number of emails sent.
- Total number of tweets!

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

All of the above functionalities requiring persistent data must be supported by a database. Any exception triggered during execution, must be logged to a file.

Good Work!