



UNIVERSIDADE DE COIMBRA

Integração de Sistemas

Ana Inês Mesquita Fidalgo - 2013134819
Diogo Miguel Costa Goncalves - 20131140087

December 22, 2016

1 Introdução

Neste projecto era-nos pedido que implementássemos um sistema de notificações. Este sistema de notificação avisaria os estudantes, inscritos numa determinada cadeira, de mudanças realizadas nos seus materiais. Se ocorrer uma mudança nos materiais, no dia a seguir às 0h, o Mule envia um email de alerta a cada um dos alunos dessa cadeira.

Para além disto, os administrados podem ainda adicionar, remover e listar subscritores através de um web service SOAP. O Mule envia a todos os estudantes inscritos e aos respectivos subscritores um email com a lista de alterações nos materiais desse curso, guardando a lista actual de materiais num XML, de forma a poder comparar no dia seguinte, caso o system vá abaixo.

Era pedido também que se implementá-se um web service REST para produzir uma lista de estudantes inscritos em JSON, contendo todas as suas informações. E ainda um segundo REST que fornecesse toda a lista de materiais de cada curso em XML.

Finalmente, ao final de cada dia é postado um twitt com as estatísticas de envio de emails, o número total de emails enviados nesse dia, o número total desde sempre e o número total de cursos que existem.

2 Estrutura da Aplicação

2.0.1 Cliente

Começamos por implementar uma consola em java para o cliente-administrador. Esta consola tem três opções: subscrever, remover subscrição e listar subscritores de uma cadeira. Quando o cliente escolha a opção pretendida e introduz os inputs necessários é chamada uma das três funções contidas neste script e criada uma instância desse serviço, ao qual fazemos um set aos seus parâmetros de entrada, a partir da função setParams. Esta função irá retornar então um resultado ao cliente, caso, por exemplo, este tenha inserido um input não válido, um par email e curso que já exista por exemplo, é-lhe emitido uma mensagem de erro.

2.0.2 REST aplicado ao projeto 2

Responde a pedidos GET em URL's especificos retornando um ficheiro .json com os dados dos estudantes ou um ficheiro .xml com os dados dos materiais. Utiliza classes extra (semelhantes as classes originais) para formatar os dados e criar a geracao automatica de xml ou json a partir duma lista de elemntos dessa classe. Existem schemas para cada tipo de dados para ajuda no Mule ou para verificacao dos dados.

2.0.3 Flows

2.0.4 is3

Este fluxo contém o menu principal do web service SOAP. Apresenta apenas um HTTP listener e um CXF que chama o serviço MeuServico e um Choice que irá avaliar qual foi a escolha do cliente, dependendo desta escolha irá redirecionar para o serviço desejado, seja listar, adicionar ou remover subscrição.

2.0.5 add subscription

Primeiramente, quando o cliente escolhe adicionar subscrição, é chamado o fluxo soap de adicionar subscrição. Tal como foi referido anteriormente, a menos que o cliente insira o um par único cliente-curso não ocorrerá subscrição. Temos também um fluxo que tem um listener para a parte web. No fluxo soap, temos um HTTP listener para o path /soap_add_subscription e um CXF que chama o ficheiro java Subscription. Quando é realizado o set dos parâmetros inseridos pelo cliente, estes parâmetros são transformados e armazenados numa hashmap no ficheiro java HashSubscription. Guardando assim o email do subscritor no índice 0 e o curso a subscrever no índice 1. Para ser realizada esta transformação é chamado o HashSubscription numa componente Java neste flow. Depois desta trasformação é realizado um set no message.payload do indice correspondente ao email para ser utilizado na validação da subscrição. É também realizado um set para o course e finalmente chamado o subflow de adicionar subscrição.

Este subflow irá, primeiramente, realizar um insert na base de dados dos dados fornecidos pelo cliente e deixa a variável "valido" a null pois a subscrição ainda não foi confirmada. Realiza ainda um update no número de cursos existentes, necessário para a realização do twitt com as estatísticas. Caso não tenha ocorrido nenhum problema, é chamado o subflow de enviar o email de validação de subscrição. Aqui é realizado um set ao conteúdo do email a enviar, neste caso com o link de confirmação e é enviado finalmente o email para o email que foi fornecido, fazendo-se update das variáveis estatísticas relativas ao número de emails enviados, seja o total como os diários.

2.0.6 validate subscription

O HTTP está à escuta do path `/validate_subscription` que é desencadeado quando o utilizador vai ao seu email e seleciona o link para confirma a sua subscrição. É realizado um set do email a validar e do curso correspondente a partir dos parâmetros fornecidos e chamado o subflow de validar subscrição. Neste subflow é apenas realizado um update na variável "valido" presente na tabela de subscrições da base de dados, mudando o seu valor de null para 1, confirmando assim que este subscritor passará a receber emails de aviso de alterações nos materiais desse curso.

2.0.7 unsubscribe

Tal como no adicionar, é chamado o flow soap que tem um CXF que chama o ficheiro java Unsubscription. A partir dos dados de input do cliente, é mais uma vez feita a transformação dos mesmos para uma hashmap e feito um set no `message.payload` com o índice correspondente, guardado assim as variáveis e finalmente é chamado o subflow de remover subscrições. Este flow chama apenas o subflow para enviar o email de validação de cancelamento de subscrição.

No subflow de enviar emails, primeiramente começamos por fazer set do conteúdo do email a enviar, construindo assim o link para o path do HTTP listener de validação de remoção de subscrição e com os dados do utilizador, email e curso, necessários posteriormente. Depois deste set é enviado o email para o remetente para o email definido e são realizados dois updates na base de dados para actualizar as variáveis estatísticas de número de envio de emails. Neste ponto ainda não foi removida a subscrição, sendo que isto só ocorrerá caso esse par cliente-curso exista e depois de o utilizador validar a sua remoção.

2.0.8 validate unsubscription

Bastante semelhante ao flow de validação de uma subscrição, com exceção de que a query que irá alterar a base de dados nesse caso é realizada previamente. É desencadeado o fluxo web quando o utilizador seleciona o link de confirmação de remoção de subscrição no seu email e feito posteriormente um set das variáveis de email e curso a serem utilizados mais tarde na query de remoção. Estes parâmetros são obtidos através do link que os contém, e é finalmente chamado o subfluxo de validação de remoção de subscrição. Neste fluxo é apenas realizado um query que irá apagar da tabela de subscrições a subscrição que contenha aquele par email-curso.

2.0.9 listar

Muito semelhantemente ao inscrever e remover subscrição, irá a partir dos dados introduzidos pelo administrador, neste caso, apenas o curso, usar uma componente java para chamar a classe `HashMapSubscription` para transformar e guardar este dado numa hashmap e de seguida realiza um set no `message.payload` com o índice correspondente ao email (neste caso ficou lá armazenado o nome do curso, sendo que o outro parâmetro ficou vazio) e é chamado o subflow de listar.

Este subflow irá realizar uma query que irá selecionar todos os emails presentes na tabela de subscrições que estejam subscritos e validados no curso pretendido. É de seguida realizado um set onde é criado um novo `StringBuilder()` para uma variável chamada `result` e com um `for each` é realizado um `append` de todos os emails encontrados. Finalmente para apresentar o resultado ao cliente é realizado um `to string` ao `result`.

2.0.10 pollerflow

Aplicacao corre todos os dias as 00:00 GMT, acede aos servicos REST do projeto 2 (que retornam um .json com os estudantes por course e um .xml com os materiais por course), guarda o .xml em disco para comparacao no dia seguinte, cria um mapa com todos os materiais mudados e os

respetivos courses, o flow seguinte verifica que estudantes precisam de ser avisados das mudancas e que estudantes estao subscritos ao course, envia um email com as mudancas todas a cada estudante que verifica um dos casos mencionados anteriormente, atualizando as variaveis na base de dados; depois le a base de dados e envia um tweet (com um programa POJO) e atualiza a base de dados.