

Integração de Sistemas

Projeto 1

Introdução

Este trabalho teve como objetivo a criação de quatro aplicações. A primeira um Web Crawler que recolhe informação sobre medalhas do web site dos jogos olímpicos (<https://www.rio2016.com/en/medal-count-country>), extrai os dados relevantes para XML, e envia-os para o Java Message Service Topic. Este tópico serve duas outras aplicações que processam os dados. Uma das aplicações, o Medal Keeper que mantém informação sobre quem ganhou cada medalha. O Medal Requester pede informações do Medal Keeper. A outra aplicação, o HTML Summary Creator que resume a informação e cria ficheiros HTML para posterior visualização. A figura 1 ilustra este cenário.

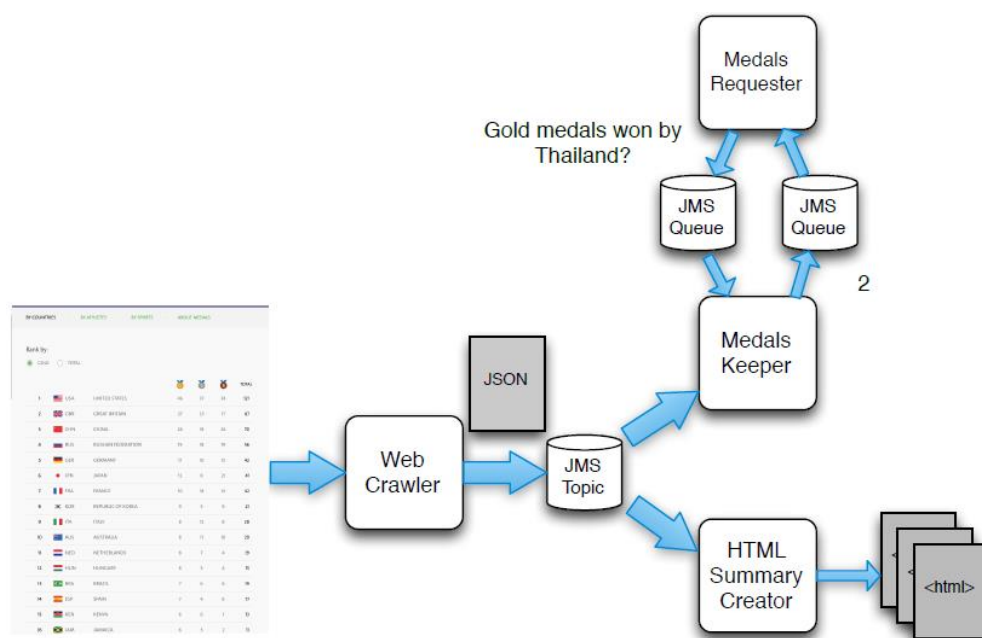


Figura 1 – O fluxo de informação

Este relatório está dividido em cinco partes. A primeira explica o modelo de dados utilizado no documento XML e o respetivo esquema XSD, e as restantes dedicam-se a cada uma das quatro aplicações: Web Crawler, HTML Summary Creator, Medal Keeper, Medal Requester.

Modelo de dados XML e esquema de validação XSD

Na figura abaixo encontra-se o modelo de dados escolhido para o documento XML (utilizando um dos países obtidos como exemplo) e as opções tomadas para o respetivo esquema de validação XSD.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jogosOlimpicos>
  ...
  <pais nome="Slovenia">
    <ranking>45</ranking>
    <abreviatura>SLO</abreviatura>
    <medalhasOuro>1</medalhasOuro>
    <medalhasPrata>2</medalhasPrata>
    <medalhasBronze>1</medalhasBronze>
    <totalMedalhas>4</totalMedalhas>
    <ouro>
      <atleta>
        <modalidade>Judo</modalidade>
        <categoria>Women -63 kg</categoria>
        <nome>TRSTENJAK Tina</nome>
      </atleta>
    </ouro>
    <prata>
      <atleta>
        <modalidade>Canoe Slalom</modalidade>
        <categoria>Kayak (K1) Men</categoria>
        <nome>KAUZER Peter</nome>
      </atleta>
      <atleta>
        <modalidade>Sailing</modalidade>
        <categoria>Finn Men</categoria>
        <nome>ZBOGAR Vasilij</nome>
      </atleta>
    </prata>
    <bronze>
      <atleta>
        <modalidade>Judo</modalidade>
        <categoria>Women -78 kg</categoria>
        <nome>VELENSEK Anamari</nome>
      </atleta>
    </bronze>
  </pais>
  ...
</jogosOlimpicos>

```

Figura 1 – Documento XML

Todos os países são elementos filhos de um elemento pai de nome jogosOlimpicos. Os elementos país, por seu turno, apresentam como atributo um identificador único (nome) e diversos elementos filhos com as características do país. Para os elementos do país com múltipla informação (como os atletas de ouro, prata e bronze) foram considerados elementos do tipo complextype de forma a separar a informação em elementos individuais filhos do elemento principal.

O esquema XSD foi gerado com o freeformatter.com e modificado manualmente para responder a algumas restrições. De forma breve, limitámo-nos a assegurar que o type de cada elemento/atributo correspondia ao tipo correto (por exemplo o elemento ranking como positiveInteger e os elementos medalhasOuro, medalhasPrata, medalhasBronze, totalMedalhas como nonNegativeInteger) e que a ocorrência ou ausência de valor nos elementos estava prevista no XSD. Como exemplo podemos apresentar o atributo ranking (<xs:attribute type="xs:positiveInteger" name="nome" use="required"/>) de presença obrigatória para a validação do documento.

Web Crawler

Esta aplicação é responsável por extrair a informação das medalhas da página web dos Jogos Olímpicos, pela criação do documento XML e pela publicação do XML no tópico do JMS. Para obter os dados a partir da página web recorreremos ao Jsoup (parser de HTML).

Para extrair a informação das medalhas começámos por obter o DOM da página web, de seguida fez-se o parse do documento obtido através de algumas seleções dos elementos HTML. A tabela com toda a informação necessária é obtida através da seleção do “`doc.select(div[class=table-count])`”, de seguida esta tabela é percorrida em ciclo, e se estivermos perante uma linha com sete colunas instanciamos um objeto País com o ranking do país, abreviatura, nome, número de medalhas de ouro, número de medalhas de prata, número de medalhas de bronze e número de medalhas ganhas. No caso de ser uma linha da tabela com quatro colunas instanciamos um objeto Atleta com a modalidade, categoria e nome e adicionamo-lo ao objeto País criado previamente. Ou seja, durante o processo de parsing do documento vão sendo criados e preenchidos os objetos Java que carregam a informação sobre as medalhas para memória.

Para a criação do documento XML fizemos o marshall para converter o objeto Jogos Olímpicos que contem toda a informação das medalhas numa string.

Por último a string contendo o documento XML é publicada no tópico (Topic) do JMS, sendo o Web Crawler um JMSProducer. Em caso de falha na publicação da string XML, o Web Crawler cai numa exceção e tenta reenviar novamente após 10 segundos, se voltar a não conseguir aguarda 20 segundos, e assim por diante, até que consiga publicar a mensagem.

HTML Summary Creator

Esta aplicação é responsável pela criação de uma página HTML utilizando os dados XML enviados pelo Web Crawler para o JMS Topic. Para transformar os dados XML numa página HTML recorreremos a um ficheiro XSL (extensible stylesheet layout) criado para o efeito e à biblioteca xalan.jar externa do apache.

Uma vez que o Web Crawler pode publicar várias mensagens para o tópico, é importante que o HTML Summary Creator consiga ler as mensagens publicadas mesmo que, por alguma razão, não esteja em execução no momento em que estas são enviadas. Como tal, a classe HTMLSummaryCreator (que implementa o `MessageListener` do JMS) cria um consumidor durável de id único que recebe as mensagens publicadas no tópico de forma assíncrona. Deste modo o tópico irá manter as mensagens para posterior recuperação. Por cada leitura do Web Crawler é gerado e mantido um novo ficheiro HTML.

Na geração do ficheiro HTML a partir do ficheiro XSL, recorreremos ao `<xsl:apply-templates/>` para aplicar um template que itera sobre cada um dos países que tenha conquistado medalhas nos jogos olímpicos. Para cada país utilizámos três `<xsl:for-each select="??">` para iterar sobre cada um dos grupos de atletas de ouro, prata e bronze de forma a extrair os dados relativos a cada atleta de forma organizada.

Na figura abaixo encontra-se um screenshot da página HTML gerada, com a listagem dos países (cabeçalho verde) ordenados por ordem crescente de ranking e as suas medalhas de ouro (cabeçalho de ouro), prata (cabeçalho de prata) e bronze (cabeçalho de bronze) adquiridas nos jogos olímpicos.

Jogos Olímpicos

45	SLO	Slovenia	Ouro: 1	Prata: 2	Bronze: 1	Total: 4
Modalidade	Categoria		Nome			
Judo	Women -63 kg		TRSTENJAK Tina			
Modalidade	Categoria		Nome			
Canoe Slalom	Kayak (K1) Men		KAUZER Peter			
Sailing	Finn Men		ZBOGAR Vasilij			
Modalidade	Categoria		Nome			
Judo	Women -78 kg		VELENSEK Anamari			

46	INA	Indonesia	Ouro: 1	Prata: 2	Bronze: 0	Total: 3
Modalidade	Categoria		Nome			
Badminton	Mixed Doubles		Natsir / Ahmad			
Modalidade	Categoria		Nome			
Weightlifting	Women's 48kg		AGUSTIANI Sri Wahyuni			
Weightlifting	Men's 62kg		IRAWAN Eko Yuli			

Figura 2 – Página HTML gerada

Medal Keeper

Esta aplicação é responsável por receber informação atualizada das medalhas que é publicada no tópico pelo Web Crawler, e por receber e responder aos pedidos recebidos do Medal Requester.

De modo a escutar dois destinos diferentes, foi criada uma thread adicional TopicReader para receber a informação atualizada das medalhas que é publicada no tópico (Topic) pelo Web Crawler. Foi criado um consumidor durável de id único, para garantir que em caso de falha ou se por alguma razão a aplicação não estiver em execução, o tópico mantenha as mensagens para posterior recuperação. Após a receção de uma mensagem, esta é validada contra o esquema XSD para verificar se cumpre a especificação. De seguida recorrendo ao unmarshall o conteúdo XML recebido é convertido num objeto.

A thread principal cria um consumidor e fica em ciclo a escutar a fila (Queue) destinada à receção dos pedidos do Medal Requester(s). Quando recebe um pedido, é feita uma procura nos dados que estão em memória. Caso a resposta ao pedido seja encontrada, é criada um MessageProducer sobre a fila temporária (obtida da mensagem recebida pelo MedalRequester) que envia a resposta ao pedido.

De forma a evitar problemas de concorrência entre as duas threads, a atualização do objeto compartilhado que contém todos os dados das medalhas é feita de forma sincronizada, deste modo asseguramos que não haja problemas quando ocorre uma pesquisa e atualização simultânea nesse objeto.

Medal Requester

Esta aplicação é responsável por receber pedidos do utilizador (como listar as medalhas ganhas por um país, listar os atletas medalhados de uma modalidade, listar as medalhas ganhas por um atleta) e devolver a resposta aos seus pedidos. Os dados introduzidos pelo utilizador são concatenados numa string para serem enviados ao Medal Keeper através de uma fila (Queue) que será lida pelo Medal Keeper. A mensagem que é enviada ao Medal Keeper inclui também uma fila temporária (anexada através do método `setJMSReplyTo(tempQueue)`) para que a resposta ao pedido seja feita através dessa fila temporária. Isso permite que vários Medal Requesters executem ao mesmo tempo.