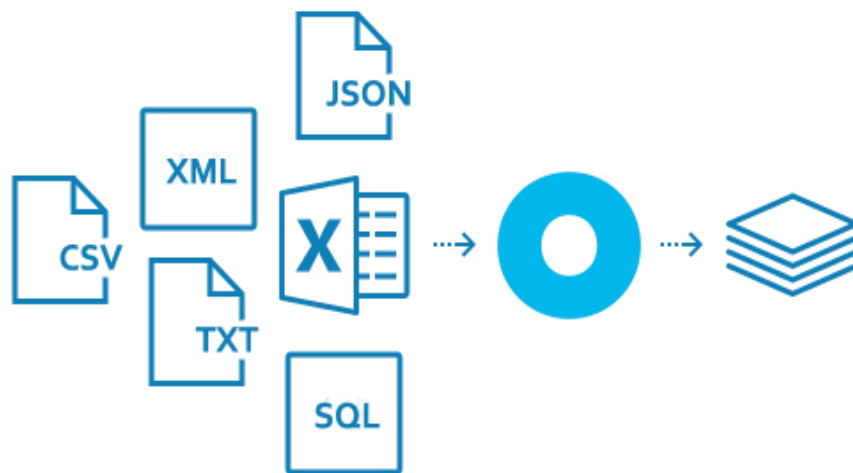


Integração de Dados com XML

Relatório Trabalho Prático



Índice

1.	Introdução	3
2.	Estrutura do Trabalho	4
3.	Implementação dos wrappers	5
3.1	Wrappers	5
4.	Criação do ficheiro XML	6
5.	Validar DTD e XSD	6
5.1	Validar DTD	6
5.2	Validar XSD	6
6.	Interface	7
7.	Conclusão	9

1. Introdução

O objetivo deste trabalho consiste na criação de uma aplicação integradora usando linguagem java que seja composto por Wrappers e que apresenta uma vista unificada de informação relativa a diversas cidades e países.

Esta informação encontra-se nas seguintes fontes de dados: <https://pt.wikipedia.org/wiki/> e <https://pt.db-city.com/>, para que o utilizador possa ver, pesquisar, acrescentar dados que respeitem os esquemas adotados e gerar ficheiros com informação selecionada.

Usando estas fontes de dados, é suposto construir um modelo global G composto por um ficheiro XML que agregue a informação de forma organizada e coerente.

O ficheiro tem por nome idades.xml e vai conter a informação de uma cidade obtida nos sites acima descritos.

Depois do processo de integração de dados estar realizado, e ser validado usando DTD E XSD, será possível ao utilizador fazer pesquisas e edições sobre a vista unificada.

2. Estrutura do Trabalho

O trabalho encontra-se dividido em diferentes ficheiros, são eles:

- Cidade.java;
- Frame.java;
- HttpRequestFunctions.java;
- JDOMFunctionsValidate.java
- JDOMFunctionsXSLT.java;
- SaxonFunctionsXQuery.java;
- TP.java;
- Wrappers.java;
- XMLFunctions.java;
- XMLJDomFunctions.java;
- XPathFunctions.java;

Além disso, adicionamos 4 ficheiros às Libraries:

- jdom-2.0.6.jar;
- saxon9-s9api.jar;
- saxon9.jar;
- JDK 17 (Default).

3. Implementação dos wrappers

3.1 Wrappers

- **Procurar link do DBCity**

Em `public static String encontrarLinkDBCityPais (String pais, String cidade)` permite encontrar o link da cidade que pretendemos. Usámos 3 expressões regulares, no entanto, não conseguimos abranger todas as cidades existentes no website.

- **Procurar link da Wikipedia**

Em `public static String Wikipedia(String cidade)` temos a possibilidade de encontrar o link da cidade no website da Wikipedia.

Restantes Wrappers:

- **`procuraCP(String link);`**
- **`procuraNumeroHabitantes(String link);`**
- **`procuraClima(String link);`**
- **`procuraCapitais(String pais, String cidade);`**
- **`procuraAltitude(String link);`**
- **`procuraLongitude(String link);`**
- **`procuraLatitude(String link);`**
- **`procuraFUSO(String link);`**
- **`procuraArea(String link);`**
- **`procuraPresidente(String link);`**
- **`procuraDensidadePopulacional(String link);`**
- **`procuraCidadesGeminadas(String link);`**
- **`procuraWebsite(String link);`**
- **`procuraBandeiraPais(String link);`**
- **`procuraBandeiraCidade(String cidade);`**
- **`procuraLinguaOficial(String pais);`**
- **`procuraMonumentos(String cidade);`**

4. Criação do ficheiro XML

Neste trabalho é gerado um ficheiro XML, denominado de “cidades.xml”. A função `public static Document adicionaCidade (Cidade cidade, Document doc)` permite adicionar cidades e de seguida a função `public static void escreverDocumentoParaFicheiro(Document doc, String caminhoFicheiro)` permite criar o ficheiro xml.

5. Validar DTD e XSD

5.1 Validar DTD

- `public static int validarDocumentoDTD(String xmlFile, String DTDFile)`

Esta função verifica se o ficheiro `xmlFile` existe em disco e se existir, associa esse ficheiro ao ficheiro DTD (se este também existir) de forma que as funções de validação fornecidas funcionem corretamente. No final, valida o ficheiro.

5.2 Validar XSD

- `public static int validarDocumentoXSD(String xmlFile, String XSDFile)`

Esta função verifica se o ficheiro `xmlFile` existe em disco e se existir, associa esse ficheiro ao ficheiro XSD (se este também existir) de forma que as funções de validação fornecidas funcionem corretamente. No final, valida o ficheiro.

6. Interface

A interface deste trabalho permite ver o conteúdo do ficheiro XML, alterar dados como eliminar ou adicionar uma cidade e também alterar alguns atributos de uma cidade. Ainda é possível efetuar pesquisas XPATH através dos atributos disponíveis e, por fim, gerar outups.

- **Ver Conteúdo XML:** Nesta opção, o utilizador consegue aceder a toda a informação, caso exista, de cidades que estejam guardados no ficheiro “cidades.xml”. Se o ficheiro não existir, mostra a mensagem de “Ficheiro não existe”.
- **Alterar Dados XML:** Nesta opção, é apresentado outras 3 opções, “Eliminar cidade”, “Adicionar cidade” e “Alterar atributos”.
 - A opção de “Eliminar cidade” foi criada a partir da função “public static Document eliminaCidade (String procura, Document doc)” (no ficheiro “XMLFuncoes.java”) e tem a finalidade de eliminar uma cidade existente no ficheiro xml.
 - A opção de “Adicionar cidade” foi criada a partir da função “public static Document adicionaCidade(Cidade cidad, Document doc)” (no ficheiro “XMLFuncoes.java”) e tem a finalidade de adicionar uma cidade que ainda não existe no ficheiro xml.
 - Na opção de “Alterar atributos” será apresentado uma nova janela que contém a informação que podemos alterar na cidade. Podemos alterar o presidente, o número de habitantes e a área da cidade.
 - No altera presidente, foi criada a partir da função “public static Document alteraPresidente (String cidade, String novoPresidente, Document doc)” (no ficheiro “XMLFuncoes.java”).

- No altera número de habitantes, foi criada a partir da função `public static Document alteraNHabitantes (String cidade, String novoHabitantes, Document doc)` (no ficheiro `XMLFuncoes.java`).

- No altera área da cidade, foi criada a partir da função `public static Document alteraArea (String cidade, String novaArea, Document doc)` (no ficheiro `XMLFuncoes.java`).

- **Efetuar Pesquisas XPATH:** Nesta opção é nos permitida encontrar as cidades que existem no ficheiro `“cidades.xml”`, através dos seus atributos, nomeadamente, o nome, o país, o número de habitantes, o clima, a capital, o presidente, o idioma e pelo código postal.
- **Gerar Outputs:** Nesta opção é nos dado outras hipóteses para fazer transformações com XSLT e XQuery. Usámos XQuery para obter um ficheiro HTML de fotos das bandeiras das cidades, outra XQuery para obter um ficheiro TXT com as cidades de um determinado país, outra XQuery para obter um ficheiro XML das 5 cidades mais populosas e uma XQuery para obter um ficheiro HTML com os monumento de uma determinada cidade, e mias 3 XSLT usadas para manipular o ficheiro original (`cidades.xml`).

7. Conclusão

Com este trabalho, conseguimos aperfeiçoar os nossos conhecimentos de Integração de Dados, nomeadamente no que diz respeito à utilização de wrappers, gerar ficheiros XML e validações com DTD e XSD.