**DOM Parser: Schema Validator in Java**

## 1. Approach of Validation of XML data

Let us consider and example of XML content as:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **<?xml** version="1.0" encoding="utf-8"**?>**  **<html>**  **<head>**This is heading**</head>**  **<body>**  **<strong>**My first HTML document in XML format.**</strong>**  **</body>**  **</html>** |

If anything other than this means not valid XML. Means, only

|  |
| --- |
| **<head>**,  **<body>**  **<strong>** |

element in side the XML, other than this is considered as invalid.

## 2. Write XSD for XML

The following XSD class is corresponding to XML structure.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **<?xml** version = "1.0" encoding = "utf-8"**?>**  **<xs:schema** xmlns:xs="http://www.w3.org/2001/XMLSchema"  elementFormDefault="qualified" attributeFormDefault="unqualified"**>**  **<xs:element** name="html" type="htmlType" **/>**    **<xs:complexType** name="htmlType"**>**  **<xs:sequence>**  **<xs:element** name="head" type="xs:string" **/>**  **<xs:element** name="body" type="myBody" **/>**  **</xs:sequence>**  **</xs:complexType>**    **<xs:complexType** name="myBody"**>**  **<xs:sequence>**  **<xs:element** name="strong" type="xs:string" **/>**  **</xs:sequence>**  **</xs:complexType>**    **</xs:schema>** |

## 3. Write Validator class

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78 | **package** com.mysoftkey.dom.validator;    **import** java.io.File;    **import** javax.xml.XMLConstants;  **import** javax.xml.parsers.DocumentBuilder;  **import** javax.xml.parsers.DocumentBuilderFactory;  **import** javax.xml.transform.dom.DOMSource;  **import** javax.xml.validation.Schema;  **import** javax.xml.validation.SchemaFactory;  **import** javax.xml.validation.Validator;    **import** org.w3c.dom.Document;    **public** **class** XSDSchemaDomValidatorExample {    **public** **static** **void** main(String[] args) {  *// actual location where .xsd and .xml reside*  String schemaName = "./src/com/mysoftkey/dom/validator/myHtml.xsd";  String xmlName = "./src/com/mysoftkey/dom/validator/myHtml.xml";    Schema schema = loadSchema(schemaName);  Document document = parseXmlDom(xmlName);    validateXml(schema, document);  }    **public** **static** **void** validateXml(Schema schema, Document document) {  **try** {  *// 3. Get a validator from the schema.*  Validator validator = schema.newValidator();  System.out.println("Validator Class: " + validator.getClass().getName());    *// validating the document against the schema*  validator.validate(**new** DOMSource(document));  System.out.println("Validation passed.");    } **catch** (Exception e) {  *// catching all validation exceptions*  System.out.println(e.toString());  }  }    **public** **static** Schema loadSchema(String schemaFileName) {  Schema schema = **null**;  **try** {  *//// 1. Lookup a factory for the W3C XML Schema language*  String language = XMLConstants.W3C\_XML\_SCHEMA\_NS\_URI;  SchemaFactory factory = SchemaFactory.newInstance(language);    */\**  *\* 2. Compile the schema. Here the schema is loaded from a java.io.File, but*  *\* you could use a java.net.URL or a javax.xml.transform.Source instead.*  *\*/*  schema = factory.newSchema(**new** File(schemaFileName));  } **catch** (Exception e) {  System.out.println(e.toString());  }  **return** schema;  }    **public** **static** Document parseXmlDom(String xmlName) {  Document document = **null**;  **try** {  DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  DocumentBuilder builder = factory.newDocumentBuilder();  document = builder.parse(**new** File(xmlName));  } **catch** (Exception e) {  System.out.println(e.toString());  }  **return** document;  } |

**Output with valid xml and XSD**:

|  |  |
| --- | --- |
| 1  2 | Validator Class: com.sun.org.apache.xerces.internal.jaxp.validation.ValidatorImpl  Validation passed. |

**Change in XML as follows:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <?xml version="1.0" encoding="utf-8"?>  <html>  <head>**This** is heading</head>  <body>  <strong>My first HTML document in XML format.</strong>  <div>My div container</div>  </body>  </html> |

At line no 6 added div element tag so its not valid as per xsd .

**Output with Invalid xml or XSD**:

|  |  |
| --- | --- |
| 1  2  3 | Validator Class: com.sun.org.apache.xerces.internal.jaxp.validation.ValidatorImpl  org.xml.sax.SAXParseException; cvc-complex-type.2.4.d: Invalid content was found starting with element 'div'.  No child element is expected at this point. |