MP_0373. Lenguajes de marcas y sistemas de gestión de información
UF4. Validación y transformación de documentos

4.1. Validación

Índice

Objetivos

Con esta unidad perseguimos los siguientes objetivos:

- **Recordar** las diferencias entre un documento **"bien formado"** y **"válido"**.
- **Conocer** qué métodos sirven para **validar documentos XML**.

Introducción

Conceptos

Los **errores** en los documentos **XML**implican que los datos que se incluyen y las directivas van a provocar un mal funcionamiento de las aplicaciones desarrolladas.

La especificación XML en W3C establece que un programa debe dejar de procesar un documento XML si encuentra un error. La razón es que el software XML debe ser pequeño, rápido y compatible.

Sin embargo, los navegadores de HTML mostrarán los documentos aunque tengan errores. Los navegadores HTML tienen mucho código que está definido solamente para el tratamiento de los errores de HTML.

Teniendo en cuenta que la única forma estructurada de declarar válido un documento XML sería la validación, esta debería refrendar:

- **La corrección de los datos**: aunque los datos no se validan o corrigen en su contenido, sí permite la detección de datos incorrectos por formatos nulos o valores fuera de rango.
- **La integridad de los datos**: comprobar que toda la información obligatoria está en el documento es una de las ventajas de la validación.
- **Establecimiento de un protocolo en el documento**: usando la validación se comprueba que el emisor y receptor traducen y tratan el documento de la misma manera y lo interpretan igual.

Documentos bien formados

Un documento XML con sintaxis correcta se denomina "***Well Formed***" o **bien formado**.

Como ya hemos visto en la unidad anterior, un documento XML debe estar bien formado.

Repasemos las reglas de sintaxis básicas definidas para los lenguajes SGML:

- Los documentos XML deben tener **un elemento raíz** solamente.
- Los elementos XML deben tener una **etiqueta de apertura y otra de cierre**.
- Las etiquetas de XML distinguen entre mayúsculas y minúsculas, **son *case-sensitive***.
- Los elementos XML deben anidarse dentro de la **jerarquía definida**.
- Los valores de **atributos XML deben ser indicados**.

Un documento **XML válido** se define en la especificación XML como un documento XML **bien formado**, que también se **ajusta a las reglas de una definición de tipo de documento (DTD)**. Los documentos XML bien formados simplemente marcan las páginas con etiquetas descriptivas.

Un documento **XML bien formado no necesita una DTD**, debe ajustarse a las reglas de sintaxis XML.

Si todas las etiquetas de un documento se forman correctamente y siguen las directrices XML, entonces un documento se considera como bien formado (pero puede no ser válido).

Documentos válidos

Un documento XML "bien formado" no significa que sea "válido". Un documento XML "válido" debe estar bien formado, ya que debe cumplir la sintaxis básica de formación de documentos bien formados de XML. Por lo tanto, además de estar bien formado, un documento deberá ajustarse a una definición de tipo de documento DTD o *schema*.

Según la Organización W3C, los documentos válidos son **los que validan contra una DTD**. Las reglas de validez significan que un documento **cumple con las restricciones establecidas**dentro de una DTD. Por lo tanto, las etiquetas o entidades deben estar en conformidad con las reglas y relaciones establecidas dentro de una DTD.

Sin embargo, **no hay control** sobre si una etiqueta o entidad es correcta. Así, una etiqueta de cabeza de primer nivel podría aplicarse a un objeto de cabeza de segundo nivel y ser válida, mientras que es incorrecta.

El énfasis en **documentos bien formados** se ha desarrollado dentro de la industria editorial, donde el uso de la información delimitada por el ángulo izquierdo y derecho se ha convertido en un problema. El énfasis en el documento bien formado permite definir, delimitar y anidar contenido para ser manejado dentro de programas que no son XML como tales, pero exhiben las características o potencial para estar bien formado.

Para que un XML sea válido, además de estar bien formado, el documento deberá ajustarse a una definición de tipo de documento DTD o *schema*.

Definición de tipos de documentos

Una definición de tipo de documento define las reglas y los elementos y atributos legales para un documento XML.

Existen varias definiciones de tipos de documentos diferentes que se pueden utilizar con XML:

DTD

Es la definición del tipo de documento original.

- Es el formato nativo y el más antiguo.
- Utiliza una sintaxis no-XML.
- Es el método más sencillo, pero presenta limitaciones.

XML Schema

Es una alternativa basada en XML para DTD.

- Evolución de la DTD descrita por el W3C.
- Utiliza sintaxis XML.
- Es más complejo y potente.

Relax NG

Es una nueva gramática de definición adoptada por la ISO.

- Muy intuitivo y más fácil de entender que el XSD.
- Puede utilizar sintaxis XML o una propia parecida a JSON.
- Convertido recientemente en un estándar ISO.

Schematron

Es un lenguaje de esquema estructural expresado en XML utilizando un pequeño número de elementos y XPath.

- Se basa en reglas en vez de en gramática.
- Convertido recientemente en un estándar ISO.
- Utiliza sintaxis XML, donde se definen reglas.

Una definición de tipo de documento define las reglas y los elementos y atributos legales para un documento XML.

Métodos de validación XML

Validar documentos con DTD

DTD define los elementos del documento, sus relaciones e información adicional que puede ser incluida, como atributos, entidades y/o notaciones. **Es el formato de esquema nativo**.

Este método resulta **el más sencillo de utilizar**, ya que lo que realiza es una comprobación del cumplimiento del DTD definido. El problema es que no soporta nuevas ampliaciones del XML. **Utiliza una sintaxis no-XML** para definir la estructura o modelo de contenido de un documento XML válido:

- Define todos los elementos.
- Define las relaciones entre los distintos elementos.
- Proporciona información adicional que puede ser incluida en el documento (atributos, entidades, notaciones).
- Aporta comentarios e instrucciones para su procesamiento y representación de los formatos de datos.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Recordatorio</heading>
    <body>¡No te olvides de nuestra reunión de esta semana!</body>
</note>
```

En este XML vemos como el DOCTYPE referencia al documento DTD para su validación.

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Este es el fichero DTD con las reglas para validar el documento.

Validador de documentos XML
La validación se realiza frente a cualquier esquema XML o DTD declarado dentro del documento XML. https://www.xmlvalidation.com/

Validar documentos con esquemas (schema)

Recordemos que los **esquemas XML son un lenguaje más avanzado que DTD** y permiten especificar mucho mejor el **sistema de datos**. Una vez validado es posible expresar la estructura y contenidos del documento.

Un esquema XML se utiliza para describir la estructura de un documento XML especificando los elementos válidos que pueden ocurrir en un documento, el orden en el que pueden ocurrir y expresando restricciones sobre ciertos aspectos de estos elementos.

Un esquema XML contiene: **vocabulario** (elementos y atributos), **contenido** (estructura y relaciones) y **tipos de datos.**

```
<?xml version="1.0" encoding="UTF-8"?>
<addresses xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance"
       xsi:noNamespaceSchemaLocation='test.xsd'>
  <address>
    <name>Joe Tester</name>
    <street>Baker street 5</street>
    <wrongExtraField/>
  </wrongClosingTag>
</addresses>
```

Fichero XML.

```
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
 <xs:element name="addresses">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="address" minOccurs='1' maxOccurs='unbounded'/>
 </xs:sequence>
 </xs:complexType>
</xs:element>

 <xs:element name="address">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="name" minOccurs='0' maxOccurs='1'/>
 <xs:element ref="street" minOccurs='0' maxOccurs='1'/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>

 <xs:element name="name" type='xs:string'/>
 <xs:element name="street" type='xs:string'/>
</xs:schema>
```

Fichero *schema*.

Validar XML con Relax NG

Relax NG es un lenguaje de esquema basado en la gramática y es más sencillo y fácil de entender que un esquema XML, ya que tiene un alto poder expresivo.

Las características clave de Relax NG son:

- Es **muy simple.**
- Es **fácil de aprender.**
- Tiene una sintaxis XML y una **sintaxis compacta no XML.**
- **No cambia** el conjunto de información de un documento XML.
- Soporta espacios de **nombres XML.**
- Trata atributos uniformemente con elementos en la medida de lo posible.
- Tiene un soporte sin restricciones para contenido no ordenado.
- Tiene un soporte sin restricciones para contenido mixto.
- Tiene una sólida base teórica.
- Puede asociarse con un lenguaje de datos distinto (por ejemplo, los tipos de datos del esquema W3C XML).

```
<addressBook>
  <card>
    <name>John Smith</name>
    <email>js@example.com</email>
  </card>
  <card>
    <name>Fred Bloggs</name>
    <email>fb@example.net</email>
  </card>
</addressBook>
```

Fichero XML.

```
<element name="addressBook" xmlns="http://relaxng.org/ns/structure/1.0">
  <zeroOrMore>
    <element name="card">
      <element name="name">
        <text/>
      </element>
      <element name="email">
        <text/>
      </element>
    </element>
  </zeroOrMore>
</element>
```

Relax NG.

RELAX NG Tutorial
Tutorial en inglés. http://relaxng.org/tutorial-20011203.html

Validar con Schematron

Schematron es un lenguaje basado en XML para validar documentos de instancia XML. Se utiliza para hacer afirmaciones sobre datos en un documento XML y también se utiliza para expresar las reglas operativas y de negocio. *Schematron* es una norma ISO.

Se basa en afirmaciones en vez de en gramática y utiliza expresiones de acceso.

En concreto, se basa en una serie de reglas y utiliza expresiones de acceso para definir lo que se permite en un documento XML, haciendo que si se cumplen sea un documento "válido". Este método es el más flexible en cuanto a la descripción de estructuras relacionales y se suele utilizar junto a otros lenguajes, como el anterior Relax NG.

El componente *Schematron* utiliza la implementación de ISO *Schematron*. Es una implementación basada en XSLT. Las reglas *Schematron* se ejecutan a través de cuatro *pipelines* de XSLT, lo que genera una XSLT final que se utilizará como base para ejecutar la aserción contra el documento XML.

El componente se escribe de forma que las reglas de *Schematron* se carguen al inicio del punto final (solo una vez) para minimizar la sobrecarga de instancia de un objeto de plantillas Java, que representa las reglas.

Tutorial
En este enlace encontrarás documentación sobre *Schematron*. http://schematron.com/

Despedida

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

En esta unidad hemos repasado la validación para documentos XML, las reglas y métodos que se utilizan para validar documentos XML.

Un documento bien formado no es lo mismo que un documento validado. Para que un documento XML esté bien formado debe cumplir las reglas de los lenguajes SGML:

- Solo debe contener caracteres Unicode legales correctamente codificados. Ninguno de los caracteres de sintaxis especiales como **<** y **&** deben usarse, excepto cuando se realizan sus funciones de marcado-delineación.
- Las etiquetas de "inicio", "fin" y "elemento vacío" que delimitan los elementos se deben anidar correctamente, sin que falte ninguno y sin superponerse. Las etiquetas de "inicio" y de "fin" deben coincidir exactamente.
- Los nombres de las etiquetas no pueden contener ninguno de los caracteres: **"# $% & '() * +, /; <=>? @ [\] ^` {|} ~**, ni un carácter de espacio, y no puede comenzar con -,. o un dígito numérico.

Un documento XML "válido" debe estar bien formado, ya que debe cumplir la sintaxis básica de formación de documento bien formado de XML. Pero además deberá ajustarse a una definición de tipo de documento DTD o *schema.*