UTILIDADES DE FACTURAE

manual de integrador de transformadores

SUBDIRECCION GENERAL DE TECNOLOGÍAS DE LA INFORMACIÓN Y DE LAS COMUNICACIONES

Nº Versión: 1.0.1

Fecha: 5/04/16

CONTROL DOCUMENTAL:

|  |  |
| --- | --- |
| Proyecto: | Utilidades de Facturae |
| Titulo: | Utilidades Facturae: manual de integrador de transformadores |
| Versión: | 1.01 |
| Fecha creación: | 05/04/2016 |
| Autor(es): | ADSI |
| Resumen: | Manual de Integrador de los Transformadores de Facturae |

DERECHOS DE USO:

La presente documentación es propiedad del Ministerio de Industria, Energía y Turismo.

CONTROL DE MODIFICACIONES POR CAMBIO DE VERSIONES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VERSIÓN | AUTOR | CAMBIOS | DESCRIPCIÓN | FECHA |
| 1.01 | ADSI |  | Primera versión | 05/04/2016 |
|  |  |  |  |  |

[1. INTRODUCCIÓN 5](#_Toc452968258)

[1.1. Propósito 5](#_Toc452968259)

[1.2. Referencias 6](#_Toc452968260)

[2. DESCRIPCIÓN GENERAL 7](#_Toc452968261)

[2.1. Transformación de facturas 7](#_Toc452968262)

# INTRODUCCIÓN

# Propósito

La obligatoriedad del uso de la factura electrónica en las relaciones contractuales con las Administraciones públicas se establece en la Ley 25/2013, de 27 de diciembre, de impulso de la factura electrónica y creación del registro contable de facturas en el Sector Público, que forma parte del Plan del Gobierno para la erradicación de la morosidad en el sector público.

Las facturas deberán haber sido expedidas en el formato que dicha Ley establece en su disposición adicional segunda y estar firmadas con firma electrónica avanzada basada en un certificado (formato Facturae).

La obligación de los Ministerios de Industria, Energía y Turismo, y de Hacienda y Administraciones Públicas de facilitar instrumentos gratuitos para facilitar la emisión de estas facturas electrónicas se establece en la Disposición final cuarta del Real Decreto Legislativo 3/2011, de 14 de noviembre, por el que se aprueba el texto refundido de la Ley de Contratos del Sector Público. Esta disposición se fundamenta en los principios de igualdad y no discriminación en el uso de medios electrónicos, informáticos y telemáticos en el ámbito de la contratación pública establecido en la Disposición adicional decimosexta del citado texto refundido de la Ley de Contratos del Sector Público.

La factura electrónica es un factor fundamental para el incremento de la productividad y por tanto de la competitividad de la economía española.

Para facilitar la adopción de la factura electrónica, sobre todo por parte de las empresas pequeñas y los autónomos, se han implementado diversas medidas de apoyo, entre las cuales destacan las utilidades de validación y visualización de facturas que se encuentran accesibles desde la página web www.facturae.gob.es:

* Validación de facturas electrónicas.
* Visualización de facturas electrónicas.
* Conversión entre versiones del formato Facturae.

Estas utilidades deben adaptarse a la actual normativa y a las necesidades de los diversos escenarios y usos en los que deben proporcionar servicio.

Los usuarios del sistema serán pequeñas y medianas empresas que quieran validar la corrección de sus desarrollos relativos a Facturación Electrónica.

# DESCRIPCIÓN GENERAL

El API XSLT (eXtensible Stylesheet Languate Transformations) puede utilizarse con múltiples propósitos permitiendo generar distintos formatos de salida a partir de un XML con datos de entrada y un transformador XSLT.

Lo más habitual es generar otro XML y más concretamente HTML.

Existen diferentes APIs que permiten realizar estas transformaciones, como por ejemplo JAXP de Oracle.

# Transformación de facturas

A continuación se muestra un ejemplo de cómo convertir un XML en otro aplicando una transformación con un XSLT.

La aplicación recibe dos argumentos. El primero es el path del XML que se quiere transformar y el segundo el path del transformador XSLT.

El fichero generado como resultado de la conversión será un fichero cuyo nombre será la fecha actual en milisegundos y su extensión xml.

**package** es.mityc.facturae.conversor.test;

**import** java.io.File;

**import** java.io.IOException;

**import** java.io.InputStream;

**import** javax.xml.parsers.DocumentBuilder;

**import** javax.xml.parsers.DocumentBuilderFactory;

**import** javax.xml.parsers.ParserConfigurationException;

**import** javax.xml.transform.Templates;

**import** javax.xml.transform.Transformer;

**import** javax.xml.transform.TransformerConfigurationException;

**import** javax.xml.transform.TransformerException;

**import** javax.xml.transform.TransformerFactory;

**import** javax.xml.transform.dom.DOMSource;

**import** javax.xml.transform.stream.StreamResult;

**import** org.apache.commons.io.FileUtils;

**import** org.w3c.dom.Document;

**import** org.xml.sax.SAXException;

**public** **class** Conversor {

**public** **static** **void** main(String[] args) {

//Comprobamos el número de parámetros

**if** ( args.length < 2 )

System.***out***.println("java Conversor.class pathFicheroFactura pathFicheroTransformador");

// Obtenemos la ruta de la factura a convertir

String pathFicheroFactura = args[0];

File fileFactura = **new** File(pathFicheroFactura);

InputStream facturaStream = **null**;

**try** {

facturaStream = FileUtils.*openInputStream*(fileFactura);

} **catch** (IOException e2) {

// **TODO** Auto-generated catch block

e2.printStackTrace();

}

// Obtenemos la versión a la que se quiere convertir la factura

String pathTransformador = args[1];

File fileXSLT = **new** File(pathTransformador);

InputStream xslStream=**null**;

**try** {

xslStream = FileUtils.*openInputStream*(fileXSLT);

} **catch** (IOException e2) {

// **TODO** Auto-generated catch block

e2.printStackTrace();

}

// Generacion de DocumentBuilder y TransformerFactory

DocumentBuilderFactory dbf = DocumentBuilderFactory.*newInstance*();

dbf.setNamespaceAware(**true**);

DocumentBuilder docBuilder=**null**;

**try** {

docBuilder = dbf.newDocumentBuilder();

} **catch** (ParserConfigurationException e2) {

// **TODO** Auto-generated catch block

e2.printStackTrace();

}

TransformerFactory tfactory = TransformerFactory.*newInstance*();

//Generacion de Document y DOMSource para XSL

**if** (xslStream != **null**)

{

Document xslDocument = **null**;

**try** {

xslDocument = docBuilder.parse(xslStream);

} **catch** (SAXException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

DOMSource xslSource = **new** DOMSource(xslDocument);

xslSource.setSystemId(xslStream.toString());

Document xmlDoc = **null**;

//Parseando la factura XML a un Document

**try** {

xmlDoc = docBuilder.parse(facturaStream);

} **catch** (SAXException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

//Generacion de DOMSource para XML

DOMSource xmlSource = **new** DOMSource(xmlDoc);

xmlSource.setSystemId(xslStream.toString());

//Generacion e inicializacion del transformador

**try** {

Templates templates = tfactory.newTemplates(xslSource);

Transformer transformer = templates.newTransformer();

transformer.setParameter("invoiceType", "Invoice");

//Visualizando la factura con formato - generando codigo HTML");

String nombreFicheroResultado = **new** Long(System.*currentTimeMillis*()).toString() + ".xml";

StreamResult result = **new** StreamResult(**new** File(nombreFicheroResultado));

transformer.transform(xmlSource,result);

}

**catch** (TransformerConfigurationException e1)

{

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**catch** (TransformerException e)

{

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

}