AJAX

XML Y JSON

Índice

- Introducción
- El lenguaje XML
- Un documento XML
- XML y AJAX
- JSON
- Literales en JavaScript
- Ejemplo de JSON

- ¿Qué es XML?
  - del inglés extensible markup language.
  - Un lenguaje de marcado extensible que puede usarse para almacenar datos en un formato estructurado, basado en texto y definido por el usuario.
  - La tecnología XML busca dar solución al problema de expresar información estructurada de la manera más abstracta y reutilizable posible.

- Partes de un documento XML
  - Cabecera, donde daremos información acerca de la versión XML en la que esta escrita el documento y la codificación del mismo:
    - <?xml version="1.0" encoding="ISO-8859-1" ?>
  - Cuerpo, parte fundamental del documento, que siempre debe de llevar al menos un elemento raiz:
    - <raiz><alumno><nombre>Luis</nombre></alumno></raiz>

- Ejemplo de documento XML:

```
<?xml version="1.0" encoding="utf-8"?>
<raiz>
    <libro isbn="0000000001">
        <titulo>Don Quijote de la Mancha</titulo>
        <autor>Cervantes</autor>
        <imagen><![CDATA[imagen1.jpg]]></imagen>
    </libro>
    <libro isbn="0000000002">
        <titulo>Episodios Nacionales</titulo>
        <autor>B. Perez Galdos</autor>
        <imagen><![CDATA[imagen2.jpg]]></imagen>
    </libro>
    <libro isbn="0000000003">
        <titulo>Yo, Robot</titulo>
        <autor>Isaac Asimov</autor>
        <imagen><![CDATA[imagen3.jpg]]></imagen>
    </libro>
</raiz>
```

- ¿Porque usar XML con AJAX?
  - Ventajas con respecto a utilizar texto plano en el retorno de información del servidor:
    - La información se devuelve estructurada pudiendo recogerla de manera ordenada y sin necesidad de devolver código HTML del servidor.
    - Es un lenguaje fácil de leer para los humanos.
    - XML lleva mucho tiempo en el mundo y los desarrolladores ya se han acostumbrado a él. Es muy común oir: "Nos gustaría que tu servidor nos devolviese un XML".

- Desventajas de utilizar XML en Ajax:
  - La información devuelta no nos sirve directamente. Es necesario llevar a cabo un proceso de lectura/recorrido del documento XML.
  - El sistema de etiquetado que emplea XML duplica información en lo documentos y los hace pesados.
    - Mas adelante veremos como otras formas de tratar la información a enviar/recibir como JSON son mucho mas eficaces.

- Recorriendo/leyendo un documento XML en JavaScript:
  - Cada navegador incorpora una serie de herramientas y librerías diferentes para el tratamiento de ficheros XML.
  - Navegar por un documento DOM de XML es muy parecido a navegar por un documento DOM de HTML: se trata de una estructura jerárquica de nodos.

- Propiedades DOM de XML
  - attributes
  - childNodes
  - firstChild / lastChild
  - nextSibling / previousSibling
  - parentNode
  - nodeValue
  - getElementsByTagName()
  - getAttribute()

- Código de ejemplo de uso Ajax con XML:

```
<html>
	<head>
		<script type="text/JavaScript" src="jquery.js"></script>
		<script>

		function enviar()
		{
			$.get("libros.xml",null,leerDatos);
		}

		function leerDatos(originalRequest)
		{

			html = "";

			datos = originalRequest;
			libro = datos.getElementsByTagName("libro");
			libros = libro.length;
			titulo = datos.getElementsByTagName("titulo");
			autor = datos.getElementsByTagName("autor");

			html += "<table><tr><td width='100'>ISBN</td><td width='150'>TITULO</td><td
width='150'>AUTOR</td></tr>";
```

- Código de ejemplo de uso Ajax con XML:

```
for(var i=0; i < libros; i++)
{
nombreTitulo = titulo[i].firstChild.nodeValue;
nombreAutor = autor[i].firstChild.nodeValue;
isbn = libro[i].getAttribute("isbn");

html += "<tr>"
html += "<td>"+isbn+"</td>"+"<td>"+nombreTitulo+"</td>"+"<td>"+nombreAutor+"</td>";
html += "</tr>"

}

html += "</table>";

$('datosLibros').html(html);

}
</script>
</head>

<body onLoad="enviar()">

<div id="datosLibros">
<table width="400">
<tr><td align="center"><img src="indicator.gif"></td></tr>
</table>
</div>

</body>
</html>
```

- ¿Que es JSON?
  - JSON, acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. Igual que XML, es legible e independiente de la plataforma.
  - JSON esta basado en la notación de literales de matriz y de objeto de JavaScript, ofreciendo la posibilidad de serialización de objetos de JavaScript, y en algunos casos de objetos de lenguajes servidor, como PHP.

- Literales de matriz y de objeto en JS:

```
//Literal de matriz
var literal = ["uno", "dos", "tres"];
alert(literal[1]); // devuelve dos

//Literal de objeto
var oLibro = {
"isbn" : "0000000001",
"titulo": "Don Quijote",
"autor" : "Cervantes"
}
alert(oLibro.titulo); // devuelve Cervantes

//Literal mixto
var oLibroVector = [{
"isbn" : "0000000001",
"titulo": "Don Quijote",
"autor" : "Cervantes"
},{
"isbn" : "0000000000",
"titulo": "La Biblia",
"autor" : "varios"
}];
alert(oLibroVector[1].autor); //devuelve varios
```

- Sintaxis JSON
  - La sintaxis JSON no es mas que la adopción de los literales tanto de matriz como de objeto de JavaScript. JSON solo representa datos, asi que no incluye variables, los datos que enviaremos o recibiéramos del servidor para un libro serian:

```
{
"isbn" : "0000000001",
"titulo": "Don Quijote",
"autor" : "Cervantes"
}
```

- Evaluar cadenas Json:
  - Para poder evaluar y convertir un String Json a un objeto en Javascript utilizaremos la función **eval().**

```
<html>
   <head>
      <script>
         var sJSON = '[{"isbn":"670894788","titulo":"Anna Karenina","autor":"Leo Tolstoy"}]';
         var oLibro = eval(sJSON); //decodificamos JSON y creamos la variable oLibro
         alert(oLibro[0].titulo);
      </script>
   </head>
   <body>
   </body>
</html>
```

- Herramientas de JSON en el Servidor:
  - Teniendo en cuenta que AJAX se utiliza para la comunicación entre cliente y servidor, no tendría sentido utilizar JSON si no estuviera soportado en el lenguaje Servidor que utilicemos.
  - En este curso vamos a estudiar una solución implementada en PHP, JSON-PHP. JSON-PHP facilita la codificación y decodificación de código JSON en PHP.

- Instalando JSON-PHP:
  - Para poder utilizar JSON-PHP debemos instalar la extensión escrita en C para PHP, o bajarnos directamente la clase JSON incluida en el fichero JSON.php.
  - La pagina del proyecto JSON-PHP es: http://mike.teczno.com/json.html

- Utilizando JSON-PHP
  - Con los métodos decode y enconde de la Clase Services_JSON podemos serializar y rescatar objetos PHP:

```
<?php
require_once("JSON.php");
$oJSON = new Services_JSON();

class Libro{
            var $isbn;
            var $titulo;
            var $autor;
            function Libro($isbn,$titulo,$autor){
                        $this->isbn = $isbn;
                        $this->titulo = $titulo;
                        $this->autor = $autor;}
      }
$libro = new Libro("0000000000","La bliblia","varios");
$sJSON = $oJSON->encode($libro);
print($sJSON); //imprimir cadena JSON
$libro2 = $oJSON->decode($sJSON);
print("titulo:".$libro2->titulo); //imprimir el valor de un atributo de libro
?>
```

- Código ejemplo JSON - AJAX:

```
<html>
<head>
<script type="text/JavaScript" src="jquery.js"></script>
<script>
function enviar()
{

 $.getJSON("jsonRequestBd.php",null,leerDatos);

}

function leerDatos(originalRequest)
{
html = "";
oLibros = originalRequest;
html += "<table><tr><td width='100'>ISBN</td><td width='150'>TITULO</td><td width='150'>AUTOR</td></tr>";

for(var i=0; i < oLibros.length; i++)
{html += "<tr>"
html += "<td>"+oLibros[i].isbn+"</td>"+"<td>"+oLibros[i].titulo+"</td>"+"<td>"+oLibros[i].autor+"</td>";
html += "</tr>"}

html += "</table>";
$('datosLibros').html(html);
}
</script>
</head>
```

- Código ejemplo JSON - AJAX:

```
<body onLoad="enviar()">

            <div id="datosLibros">
                  <table width="400">
                        <tr><td align="center"><img src="indicator.gif"></td></tr>
                  </table>
            </div>
      </body>
</html>
```

```
<?php
require_once("JSON.php");
$oJSON = new Services_JSON();
class Libro{
            var $isbn;
            var $titulo;
            var $autor;
            function Libro($isbn,$titulo,$autor){
                        $this->isbn = $isbn;
                        $this->titulo = $titulo;
                        $this->autor = $autor;}
      }
$libros = array();
$libros[0] = new Libro("0000000000","La bliblia","varios");
$libros[1] = new Libro("0000000001","Don Quijote de la Mancha","Cervantes");
$libros[2] = new Libro("0000000002","Yo, Robot","Isaac Asimov");
$sJSON = $oJSON->encode($libros);
print($sJSON);?>
```