



Recuperación de datos del servidor con JS y JSON

JUAN CARLOS CONDE RAMÍREZ

WEB-TECHNOLOGIES

Ejemplo de comunicación asíncrona

- Ahora sí, con el fin de demostrar cómo recuperar datos del servidor, de forma asíncrona, en formato JSON (en lugar TXT o XML) implementaremos el siguiente ejemplo:
 - Se diseñará un sitio que permita ingresar el número de *INE* de una persona y nos retorne su *apellido*, *nombre* y *dirección* donde debe votar.
 - Para reducir el tamaño del problema, y concentrarnos en la forma de transmisión de datos y su posterior recuperación en el navegador, NO se ha implementado una base de datos en el servidor.
 - Sin embargo, se sabe que en la vida real los datos de los votantes se encuentran en una BD.

index.html

- El archivo **index.html** contiene algunos detalles importantes:
 - El archivo HTML sólo tiene un formulario simple.
 - Se define el `div` con `id="resultados"` para mostrar posteriormente los datos devueltos por el servidor.
 - El archivo **funciones.js** contiene los detalles del archivo JavaScript.

funciones.js, I

- Cuando se presiona el botón enviar se ejecuta la función:

```
function presionBoton(e)
{
    var ob = document.getElementById( 'ine' );
    recuperarDatos( ob.value );
}
```

- En esta función recuperamos el *documento* ingresado por el operador y llamamos a la función `recuperarDatos` pasando como parámetro el número de `ine` de la persona.

funciones.js, II

- La función sería la siguiente:

```
function recuperarDatos(ine)
{
    conexion1 = crearXMLHttpRequest();
    conexion1.onreadystatechange = procesarEventos;
    conexion1.open( 'GET', 'datos_ine.php?ine=' + ine, true );
    conexion1.send( null );
}
```

- Se crea un objeto de la clase XMLHttpRequest que realiza una **comunicación asíncrona** con el servidor, usando un método GET y pasando como parámetro el ine ingresado por el visitante.

funciones.js, III

- Después, la función `procesarEventos` se ejecuta para cada uno de los estados de la petición:

```
function procesarEventos()
{
    var resultados = document.getElementById( "resultados" );
    if( conexion1.readyState == 4 )
    {
        var datos = eval( "(" + conexion1.responseText + ")" );
        var salida = "Apellido:" + datos.apellido + "<br>";
        salida = salida + "Nombre:" + datos.nombre + "<br>";
        salida = salida + "Dirección donde debe votar:" + datos.direccion;
        resultados.innerHTML = salida;
    }
    else
    {
        resultados.innerHTML = "Cargando...";
    }
}
```

funciones.js, IV

- Cuando la propiedad `readyState` tiene almacenado un 4 procedemos a recuperar el *string* devuelto por la conexión.

Valor	Estado	Descripción
0	UNINITIALIZED	todavía no se llamó a <code>open()</code> .
1	LOADING	todavía no se llamó a <code>send()</code> .
2	LOADED	<code>send()</code> ya fue invocado, y los encabezados y el estado están disponibles.
3	INTERACTIVE	Descargando; <code>responseText</code> contiene información parcial.
4	COMPLETED	La operación está terminada.

- Así que esta es apenas una introducción al uso del objeto `XMLHttpRequest`, que es el núcleo del **funcionamiento de AJAX**.

funciones.js, V

- Como se puede observar más abajo en el código, lo que sigue es obtener la propiedad `conexion1.responseText`; un *string* que almacena los datos en formato JSON.
- Así que procedemos a generar un objeto en JavaScript, mediante la función `eval`, y posteriormente mostramos los datos accediendo a los atributos de dicho objeto.
- Como podemos observar es muy fácil acceder luego a la información devuelta por el servidor.

datos_ine.php, I

- Por último, se analiza el archivo **pagina1.php**
- Como se menciono anteriormente, para concentrarnos en JSON no extraemos la información de una base de datos.
- Por medio de un `switch`, verificamos de que número de INE se trata y procedemos a inicializar tres variables.

datos_ine.php, II

```
switch ($_REQUEST['ine'] ) {
    case '1':
        $nombre = 'Juan';
        $apellido = 'Rodríguez';
        $direccion = '15 Oriente 123';
        break;
    case '2':
        $nombre = 'Ana Luisa';
        $apellido = 'Maldonado';
        $direccion = '12 Sur 245';
        break;
    case '3':
        $nombre = 'Laura';
        $apellido = 'Puebla';
        $direccion = 'Carmen Serdán 785';
        break;
    default:
        $nombre = 'No encontrado';
        $apellido = 'No encontrado';
        $direccion = 'No encontrado';
}
```

datos_ine.php, III

- Lo más interesante es como procedemos a generar la salida con formato JSON:

```
echo "{  
    'nombre': '$nombre',  
    'apellido': '$apellido',  
    'dirección': '$dirección'  
}";
```

- Recordemos que todo objeto JSON debe ir entre llaves y a cada atributo le debe seguir el carácter dos puntos y el valor de dicho atributo.

[Ver archivos adjuntos...](#)