

21 - De PHP a JSON (utilizando la función json_encode)

En el concepto anterior habíamos visto como generar un archivo con formato JSON en el servidor y enviárselo al cliente (navegador). Esta metodología de generar el string con formato JSON puede ser muy engorroso cuando las estructuras comienzan a ser más complejas.

Se cuenta en PHP con una función que nos facilita dicha actividad llamada json_encode.

Confeccionaremos un problema que rescate un conjunto de registros de una tabla MySQL y seguidamente los transforme en formato JSON. En el navegador mediante el método parse del objeto JSON generaremos un objeto literal JavaScript y procederemos a mostrarlo.

El problema que resolveremos es el siguiente:

Se tiene una tabla llamada "perifericos" donde almacenamos el código, descripción y precio de distintos periféricos de computadoras. Generar un archivo JSON en el servidor y proceder a mostrar los datos de los periféricos en el navegador.

pagina1.html:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Problema</title>
  <script src="funciones.js"></script>
</head>
<body>
  <h2>Recuperar datos del servidor almacenados en una base de datos en formato JSON utilizando
  <br>
  <input type="button" value="Recuperar" id="boton1">
  <div id="resultados"></div>
</body>
</html>
```

Este archivo no tiene nada nuevo.

Lo más interesante y nuevo se presenta en el archivo pagina1.php:

```

<?php
header('Content-Type: text/txt; charset=utf-8');
$conexion=mysqli_connect("localhost","root","","bdajax") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,descripcion,precio from perifericos") or
    die("Problemas en el select".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    $vec[]=$reg;
}

$cad=json_encode($vec);
echo $cad;
?>

```

Lo primero que hacemos es guardar todos los registros de la tabla perifericos en el vector llamado \$vec, esto mediante el ciclo:

```

while ($reg=mysqli_fetch_array($registros))
{
    $vec[]=$reg;
}

```

Luego llamamos a la función json_encode:

```
$cad=json_encode($vec);
```

Ahora ya tenemos en la variable \$cad el contenido del vector en formato JSON.

Procedemos seguidamente a la salida de esta cadena:

```
echo $cad;
```

Luego el archivo JavaScript es:

```

addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
    var ob=document.getElementById('boton1');
    ob.addEventListener('click',presionBoton,false);
}

var conexion1;
function presionBoton(e)
{
    conexion1=new XMLHttpRequest();
    conexion1.onreadystatechange = procesarEventos;
    conexion1.open('GET','pagina1.php', true);
    conexion1.send();
}

function procesarEventos()
{
    var resultados = document.getElementById("resultados");
    if(conexion1.readyState == 4)
    {
        alert('Cadena en formato JSON: '+conexion1.responseText);

        var datos=JSON.parse(conexion1.responseText);
        var salida="";
        for(var f=0;f<datos.length;f++)
        {
            salida += 'Codigo:'+datos[f].codigo+"<br>";
            salida += 'Descripcion:'+datos[f].descripcion+"<br>";
            salida += 'Precio:'+datos[f].precio+"<br><br>";
        }
        resultados.innerHTML = salida;
    }
    else
    {
        resultados.innerHTML = "Cargando...";
    }
}

```

La función procesarEventos rescata los datos enviados por el servidor y los muestra inicialmente tal como llegan en una ventana mediante el comando alert:

```
alert('Cadena en formato JSON: '+conexion1.responseText);
```

Podemos ver que se trata de un string JSON correctamente formado.

Ahora sí mediante el método parse del objeto JSON transformamos el string:

```

var datos=JSON.parse(conexion1.responseText);
var salida="";
for(var f=0;f<datos.length;f++)

```

```
{
    salida += 'Codigo:'+datos[f].codigo+"<br>";
    salida += 'Descripcion:'+datos[f].descripcion+"<br>";
    salida += 'Precio:'+datos[f].precio+"<br><br>";
}
resultados.innerHTML = salida;
```

Para este problema posiblemente es más fácil generar un trozo de HTML en el servidor y en el navegador solo mostrarlo, pero hay muchas situaciones que necesitamos recuperar una estructura de datos del servidor y proceder a su procesamiento en el navegador.

Problema resuelto.

Se tiene una tabla llamada "perifericos" donde almacenamos el código, descripción y precio de distintos periféricos de computadoras. Generar un archivo JSON en el servidor y proceder a mostrar los datos de los periféricos en el navegador. Utilizar la librería JSON.php para facilitar la generación del archivo JSON en el servidor.

pagina1.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Problema</title>
    <script src="funciones.js"></script>
</head>
<body>
    <h2>Recuperar datos del servidor almacenados en una base de datos en formato JSON utilizando la función
    json_encode</h2>
    <br>
    <input type="button" value="Recuperar" id="boton1">
    <div id="resultados"></div>
</body>
</html>
```

[Ejecutar ejemplo](#)

funciones.js

```
addEventListener('load',inicializarEventos,false);

function inicializarEventos()
{
    var ob=document.getElementById('boton1');
    ob.addEventListener('click',presionBoton,false);
}

var conexion1;
function presionBoton(e)
{
    conexion1=new XMLHttpRequest();
    conexion1.onreadystatechange = procesarEventos;
    conexion1.open('GET','pagina1.php', true);
    conexion1.send();
}
```

pagina1.php

```
<?php
header('Content-Type: text/txt; charset=utf-8');
$conexion=mysqli_connect("localhost","root","","bdajax") or
    die("Problemas con la conexión");

$registros=mysqli_query($conexion,"select codigo,descripcion,precio from perifericos") or
    die("Problemas en el select".mysqli_error($conexion));

while ($reg=mysqli_fetch_array($registros))
{
    $vec[]=$reg;
}

$cad=json_encode ($vec);
echo $cad;
```

Problema propuesto.

Modificar el problema planteado para permitir consultar los datos de un periférico ingresando el código para su búsqueda.

[Ver solución](#)

[Retornar](#)

