



Cómo cargar data con JSON, JQuery, PHP y MySQL usando PDO

La extensión Objetos de Datos de PHP (PDO) es una interfaz ligera, para acceder a bases de datos con PHP, es decir, es una capa de abstracción de acceso a datos. Por default WampServer PHP habilita la extensión php_PDO_MySQL, si no se encuentra incluida puede ser añadida directamente en php.ini.

Primero se crea el archivo PHP (get_clientes.php) que se conectará a la base de datos y cargará los datos. Para el siguiente ejemplo se cargan los clientes del grupo de clientes definido por el parámetro tomado de \$_GET.

```
<?
$dsn = "mysql:host=".EWRPT_CONN_HOST.";dbname=".EWRPT_CONN_DB."";
$usuario= EWRPT_CONN_USER;
$password = EWRPT_CONN_PASS;
$pdo = new PDO($dsn, $usuario, $password);
$grupocliente=$_GET["x_grupocliente"];

$rows = array();
if (isset($grupocliente)) {
    $stmt = $pdo->prepare("SELECT cliente, nomcliente FROM ti_clientes WHERE grupocliente = ? ORDER BY fec_nac
DESC");
    $stmt->execute(array($grupocliente));
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);
}
echo json_encode($rows);
?>
```

Luego de ejecutar el query se obtiene el resultado con “fetchAll” que devuelve todas las filas y columnas en un array secuencial en un solo paso. Es decir, que “\$rows” contiene un objeto iterable con todas las tuplas.

El parámetro de fetchAll (PDO::FETCH_ASSOC) permite obtener un array indexado por los nombres de las columnas del conjunto de resultados.

Luego, para crear una cadena en formato JSON en PHP se dispone de una función llamada json_encode(), que recibe el arreglo que se quiere convertir en notación JSON y devuelve una cadena de texto con el JSON producido. Para el ejemplo, la devolución será un arreglo de arreglos:

```
[{"cliente": "3", "nomcliente": "Juan Perez"}, {"cliente": "4", "nomcliente": "Martín Roberts"} ]
```

El explorador almacenará en caché el resultado para peticiones iguales si está configurado para esto. Ahora es necesario definir el archivo html/javascript que manejará los datos:

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
<title>Clientes</title>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>
<script language="javascript" type="text/javascript">
```

Para ello usamos la librería JQUERY 1.6.1.

Nótese que getJSON invoca a get_clientes.php para obtener los datos de la base de datos MySQL y define la función “grupoclientes” dentro de la cual se itera para cada grupo de clientes y se forma una lista de botones de opción que se traslada al de “grupos” que se cargó en la variable “html”.

```
$(document).ready(function() {  
  
    $.getJSON('get_clientes.php', function(grupoclientes) {  
  
        var html = "";  
  
        $.each(grupoclientes, function(index, array) {  
  
            html = html + '<input type="radio" name="grupoClientes" value="" + array['nomcliente'] + ""  
/><label>' + array['nomcliente'] + '</label><br> '  
  
            });  
  
            $('#grupos').html(html);  
  
        });  
  
    });  
</script>  
</head>  
<form>  
  
    <div>  
  
        Grupo de clientes:  
  
        <span id="grupos"></span>  
  
    </div>  
</form>
```