

# **TEMA 11**

## **ACCESO A BASE**

### **ABIERTA CON PHP**

## ACCESO A UNA BASE DE DATOS ABIERTA CON PHP

Aunque acabamos de ver cómo acceder directamente a la información guardada en JSON, lo normal es que el acceso a la información de una base de datos se realice mediante php, veamos y expliquemos los pasos que debemos dar para acceder a la información de esta forma.

Como siempre partiremos de un archivo html donde tendremos una cabecera, un cuadro de selección, un botón de comando y una tabla. De la siguiente forma:

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8" />
5    <title> </title>
6    <link rel="stylesheet" type="text/css" href="estilos.css" media="screen" />
7  </head>
8  <body>
9    <h1 id="title">ACCESO A LA INFORMACIÓN CON PHP</h1>
10   <form action="#" id="form">
11     <select name="Nombre" id="listado"></select>
12     <input type="submit" value="Get Data">
13   </form>
14
15   <table id="table">
16     <thead>
17       <th>Nombre</th>
18       <th>Fecha de alta</th>
19       <th>NIF</th>
20     </thead>
21   </table>
22
23   <script src="scripts.js"></script>
24 </body>
25 </html>
```

Lo primero que tenemos que hacer es crear el fichero .php que nos permitirá conectar con la base de datos a la que queremos acceder. En nuestro caso, la conexión la haremos a través de un servidor local a través de xampp, pero se puede hacer igualmente sobre un servidor a través de su dirección URL.

El programa que queremos hacer recogerá toda la información que tenemos en una tabla de la base de datos para añadir a un cuadro de selección la información que se recoge en el nombre del vendedor.

```
1 <?php
2     $conn = NULL;
3     try{
4         //Hacemos la conexión con la base de datos test que ya tenemos creada y lanzada
5         //en nuestro servidor, por eso el localhost, en este caso el usuario de acceso
6         //en el servidor es root y como podemos ver no tiene contraseña
7         $con = new PDO("mysql:host=localhost; dbname=test; charset=utf8", 'root', '');
8         //La primera vez que entramos queremos seleccionar todos los vendedores para añadirlos
9         //al cuadro de lista, ya que acabamos de acceder
10        $sql = "SELECT * FROM vendedores";
11
12        //Le decimos que sobre la base de datos queremos ejecutar la instrucción sql
13        //sobre la que estamos trabajando
14        $stm=$con->prepare($sql);
15        //La ejecutamos
16        $stm->execute();
17        //Guardamos la información resultante en un variable, fijaros que estamos en
18        //php por lo que las variables deben empezar por el símbolo $
19        $resultSet = $stm->fetchAll(PDO::FETCH_ASSOC);
20        //Convertimos a formato JSON lo obtenido en la ejecución
21        echo json_encode($resultSet);
22    }
23    }catch (PDOException $e){ //esta se ejecutará si existe alguna excepción anterior
24        echo "Error ". $e->getMessage();
25    }
26    ?>
```

Veamos cómo debemos acceder ahora desde JavaScript a este fichero que nos permitirá acceder a la información.

```
1 const form = document.getElementById('form')
2 const listado = document.getElementById('listado')
3 const table = document.getElementById('table')
4 //El id será el valor que usaremos para acceder cuando seleccionemos un valor
5 const getData = () => {
6     //Creamos el objeto que nos va a servir para acceder a la base
7     let xhr = new XMLHttpRequest()
8     //Accedemos a la base de datos mediante el archivo .php
9     xhr.open('GET', 'acceso.php')
10    //Especificamos lo que queremos que se haga cuando se cargue el contenido
11    xhr.addEventListener('load', (data) => {
12        //Creamos el vector con la información que pasa
13        const dataJSON = JSON.parse(data.target.response)
14        //Mostramos en consola esta información para comprobar si ha funcionado
15        console.log(dataJSON)
16        const fragment = document.createDocumentFragment()
17        for (const vendedor of dataJSON)
18        {
19            const option = document.createElement('option');
20            option.setAttribute('value',vendedor.A)
21            option.textContent = vendedor.B;
22            fragment.append(option)
23        }
24        listado.appendChild(fragment)
25    }) |
26    //Enviamos la solicitud al servidor
27    xhr.send()
28    };
29    getData();
30
```

Como podemos ver en el ejercicio, tras hacer la conexión GET y a la dirección URL “acceso.php”, si la conexión se ha generado correctamente, el siguiente paso es mediante un objeto de tipo fragment, añadimos toda la información que tenemos en el campo B donde se guarda el nombre de los vendedores, al cuadro de selección. Fijaros que la option que utilizamos para identificar los elementos es el campo A, que es un identificador autonumérico.

### 1. COMO PASAR PARÁMETROS A acceso.php A TRAVÉS DE GET

Vamos a modificar el ejercicio anterior, donde vamos a hacer lo siguiente, la primera vez que entramos a la aplicación, se añadirá el nombre de los vendedores en el cuadro de selección, pero si elegimos uno de esos vendedores y pulsamos el botón, lo que hará nuestro ejercicio es mostrar en la tabla del documento .html, la información asociada a ese vendedor.

Veamos los cambios que debemos hacer para ello, primero a la función `getData`, le pasaremos un parámetro, que será el identificador del vendedor si pulsamos el botón. Esta función, entonces se modificará, de la siguiente manera, el código que tenemos se aplicará si el identificador no está definido.

```
5  const getData = (id) => {
6    //Creamos el objeto que nos va a servir para acceder a la base
7    let xhr = new XMLHttpRequest()
8    //Si no hemos pasado ningún identificador cargamos el contenido
9    if (id == undefined) {
10     //Accedemos a la base de datos mediante el archivo .php
11     xhr.open('GET', 'acceso.php')
12     //Especificamos lo que queremos que se haga cuando se cargue el contenido
13     xhr.addEventListener('load', (data) => {
14       //Creamos el vector con la información que pasa
15       const dataJSON = JSON.parse(data.target.response)
16       //Mostramos en consola esta información para comprobar si ha funcionado
17       console.log(dataJSON)
18       const fragment = document.createDocumentFragment()
19       for (const vendedor of dataJSON)
20       {
21         const option = document.createElement('option');
22         option.setAttribute('value',vendedor.A)
23         option.textContent = vendedor.B;
24         fragment.appendChild(option)
25       }
26       listado.appendChild(fragment)
27     }) }else{
```

Veamos qué debemos hacer en el else, que será cuando sí que exista un elemento seleccionado en el cuadro y pulsemos el botón. Lo primero que debemos hacer es pasar a `acceso.php` el identificador, para pasar un parámetro a través de GET lo que tenemos que hacer es añadir a la dirección una interrogación `?` y luego especificaremos el elemento y el valor que queremos pasar.

```
xhr.open("GET", `acceso.php?id=${id}`);
xhr.addEventListener('load', (data) => {
```

También debemos modificar el for, ya que en este caso lo que vamos a hacer es añadir información a la tabla que tenemos creada en nuestra página.

```

41     }) }else{
42         xhr.open("GET", 'acceso.php?id='+id);
43         xhr.addEventListener('load', (data) => {
44             //Creamos el vector con la información que pasa
45             const dataJSON = JSON.parse(data.target.response)
46             //Mostramos en consola esta información para comprobar si ha funcionado
47             console.log(dataJSON)
48             const fragment = document.createDocumentFragment()
49             for (const vendedor of dataJSON)
50             {
51                 //creamos un elemento tr asociado a la fila que queremos añadir
52                 const row = document.createElement('tr');
53                 const dataName = document.createElement('td');
54                 const dataFecha = document.createElement('td');
55                 const dataNif = document.createElement('td');
56                 dataName.textContent = vendedor.B;
57                 dataFecha.textContent = vendedor.C;
58                 dataNif.textContent = vendedor.D;
59                 row.append(dataName);
60                 row.append(dataFecha);
61                 row.append(dataNif);
62                 fragment.append(row);
63             }
64             table.append(fragment)
65         })
66     }

```

Como esta parte asociada al else queremos que se ejecute cuando pulsemos el botón y nos venga un identificador asociado, lo que debemos hacer también es poner a la escucha el formulario donde tenemos ese botón, de la siguiente forma:

```

4  form.addEventListener("submit", (e)=>{
5      //Eliminamos la acción por defecto de los eventos
6      e.preventDefault()
7      //selectedIndex nos devuelve el índice que ocupa en la lista
8      //el elemento seleccionado, pero a nosotros nos interesa el value
9      console.log(listado.selectedIndex);
10     //Para acceder al value haremos lo siguiente, acceder al
11     //hijo que ocupa la posición seleccionada
12     console.log(listado.children[listado.selectedIndex]);
13     // Y para acceder al value, lo añadimos
14     console.log(listado.children[listado.selectedIndex].value);
15
16 })

```

Con el valor obtenido en la última opción es con el que tenemos que llamar a la función getData. Por tanto el código nos quedará así:

```

4  form.addEventListener("submit",(e)=>{
5      //Eliminamos la acción por defecto de los eventos
6      e.preventDefault()
7      //selectedIndex nos devuelve el índice que ocupa en la lista
8      //el elemento seleccionado, pero a nosotros nos interesa el value
9      //console.log(listado.selectedIndex);
10     //Para acceder al value haremos lo siguiente, acceder al
11     //hijo que ocupa la posición seleccionada
12     // console.log(listado.children[listado.selectedIndex]);
13     // Y para acceder al value, lo añadimos
14     //console.log(listado.children[listado.selectedIndex].value);
15     getData(listado.children[listado.selectedIndex].value);
16
17 })

```

Ahora necesitamos modificar también el archivo acceso.php, ya que la sentencia SQL que queremos aplicar dependerá de si el identificador está definido o no. Tendremos pues que modificar también este archivo con un if, en función de si recibe algo o no.

```

try{
    //Hacemos la conexión con la base de datos marvel que ya tenemos creada y lanzada
    //en nuestro servidor, por eso el localhost
    $con = new PDO("mysql:host=localhost; dbname=test; charset=utf8", 'root', '');

    if(isset($_GET['id'])){
        $id = $_GET['id'];
        //Si seleccionamos un elemento del cuadro de lista, lo que haremos será seleccionar
        // solo el elemento que coincide con el elemento que hemos pasado
        $sql = "SELECT * FROM vendedores WHERE A=$id";
    }else{
        //La primera vez que entramos queremos seleccionar todos los vendedores para añadirlos
        //al cuadro de lista, ya que acabamos de acceder
        $sql = "SELECT * FROM vendedores";
    }

    //Le decimos que sobre la base de datos queremos ejecutar la instrucción sql
    //sobre la que estamos trabajando
    $stm=$con->prepare($sql);
    //La ejecutamos
    $stm->execute();
    //Guardamos la información resultante en un variable, fijaros que estamos en
    //php por lo que las variables deben empezar por el símbolo $
    $resultSet = $stm->fetchAll(PDO::FETCH_ASSOC);
    //Convertimos a formato JSON lo obtenido en la ejecución
    echo json_encode($resultSet);

}catch (PDOException $e){
    echo "Error ". $e->getMessage();
}

```

Con el código así, si nosotros seguimos seleccionando elementos y pulsando el botón, nos incluirá cada uno de los valores que seleccionemos. Pero qué tenemos que hacer si queremos que se añada el elemento seleccionado, pero eliminar el que ya estaba incluido, para ello utilizaremos la función `removeChild` que ya vimos en el DOM. Veamos dónde y cómo queda el código.

```

61         row.append(dataNif);
62         fragment.append(row);
63     }
64
65     if(table.children[1])
66     table.removeChild(table.children[1]);
67     //Seleccionamos el 1 porque el 0 corresponde con la cabecera
68     table.append(fragment)
69 }

```

## 2. ENVIO DE DATOS A TRAVÉS DE POST

Para enviar información a través de POST a una base de datos, usaremos el objeto FormData que puede recibir como parámetros datos. Partiremos de este documento .html.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11      <h1 id = " title " > AJAX III </h1 >
12      <form action="#" id = "formulario" >
13          <label > Nombre</label >
14          <input type="text" name="nombre" />
15
16          <label > Fecha alta </label >
17          <input type="text" name="fechaAlta" />
18
19          <label > NIF </label >
20          <input type="text" name="nif" />
21
22          <label > Fecha Nacimiento </label>
23          <input type="text" name="fechaNacimiento">
24
25          <input type="submit" value="enviar"></input>
26      </form>
27
28      <script src="scripts.js "> </script>
29
30
31  </body>
32  </html>

```

Ahora nos vamos al documento .js, donde creamos el conector para acceder al acceso.php y desde allí se enviará la información para guardarla.

```
1  const form = document.getElementById('formulario')
2
3
4  form.addEventListener('submit', (e) => {
5    e.preventDefault(); // Evitamos que se recarge el formulario
6    sendData(form) // Le enviamos el propio formulario
7  })
8
9  const sendData = (data) => {
10    let xhr = new XMLHttpRequest ( )
11    xhr.open('POST', 'acceso.php')
12    // Creamos este objeto que puede contener datos
13    // en nuestro caso le enviamos todo lo que recibe
14    const formData = new FormData(data)
15    xhr.send(formData)
16  }
```

Y el documento acceso.php donde solo enviamos la información porque para poder acceder a la base la información que enviamos debe ser correcta.

```
1  <?php
2
3  echo json_encode($_POST);
4  ?>
```