

1 Práctica 1 XMLHttpRequest ? GET + PHP ? MySQL

Se trata de crear una pequeña web donde trabajamos con una base de datos MySQL y hacemos consultas en ella con PHP. Las peticiones al servidor las haremos con el objeto XMLHttpRequest de JavaScript utilizando el método GET.

Creamos una base de datos **marvel** con un usuario **marvel** que tenga sobre ella control total (contraseña **abc123**). Esta base de datos tendrá una única tabla que podemos importar con el siguiente archivo:

```
-- phpMyAdmin SQL Dump
-- version 4.9.5deb2
-- https://www.phpmyadmin.net/
--
-- Servidor: localhost:3306
-- Tiempo de generación: 24-01-2021 a las 16:14:02
-- Versión del servidor: 8.0.22-0ubuntu0.20.04.3
-- Versión de PHP: 7.4.3

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de datos: `marvel`
--

-----

--
-- Estructura de tabla para la tabla `marvels`
--

CREATE TABLE `marvels` (
  `ID` int NOT NULL,
  `name` text,
  `alignment` text,
  `gender` text,
  `hometown` text,
  `fighting_Skills` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Volcado de datos para la tabla `marvels`
--

INSERT INTO `marvels` (`ID`, `name`, `alignment`, `gender`, `hometown`, `fighting_Skills`) VALUES
(1, 'Spider Man', 'Good', 'Male', 'USA', 4),
(2, 'Iron Man', 'Neutral', 'Male', 'USA', 4),
(3, 'Hulk', 'Neutral', 'Male', 'USA', 4),
(4, 'Wolverine', 'Good', 'Male', 'Canada', 7),
(5, 'Thor', 'Good', 'Male', 'Asgard', 4),
(6, 'Green Goblin', 'Bad', 'Male', 'USA', 3),
(7, 'Magneto', 'Neutral', 'Male', 'Germany', 4),
(8, 'Thanos', 'Bad', 'Male', 'Titan', 4),
(9, 'Loki', 'Bad', 'Male', 'Jotunheim', 3),
(10, 'Doctor Doom', 'Bad', 'Male', 'Latveria', 4),
(11, 'Jean Grey', 'Good', 'Female', 'USA', 4),
(12, 'Rogue', 'Good', 'Female', 'USA', 7),
(13, 'Storm', 'Good', 'Female', 'Kenya', 4),
(14, 'Nightcrawler', 'Good', 'Male', 'Germany', 3),
(15, 'Gambit', 'Good', 'Male', 'EUA', 4),
(16, 'Captain America', 'Good', 'Male', 'EUA', 6),
(17, 'Cyclops', 'Good', 'Male', 'EUA', 4),
(18, 'Emma Frost', 'Neutral', 'Female', 'EUA', 3),
(19, 'Kitty Pryde', 'Good', 'Female', 'EUA', 5),
```

```

(20, 'Daredevil', 'Good', 'Male', 'EUA', 5),
(21, 'Punisher', 'Neutral', 'Male', 'EUA', 6),
(22, 'Silver Surfer', 'Good', 'Male', 'Zenn-La', 2),
(23, 'Ghost Rider', 'Good', 'Male', 'EUA', 2),
(24, 'Venon', 'Neutral', 'Male', 'EUA', 4),
(25, 'Juggernaut', 'Neutral', 'Male', 'EUA', 4),
(26, 'Professor X', 'Good', 'Male', 'EUA', 3);

--
-- Índices para tablas volcadas
--

--
-- Indices de la tabla `marvels`
--
ALTER TABLE `marvels`
  ADD PRIMARY KEY (`ID`);

--
-- AUTO_INCREMENT de las tablas volcadas
--

--
-- AUTO_INCREMENT de la tabla `marvels`
--
ALTER TABLE `marvels`
  MODIFY `ID` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=29;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

Crearemos también un archivo PHP para poder acceder a los datos dado el campo 'ID'. Si no se pasa el 'ID' del registro que se busca, el archivo PHP devolverá toda la tabla **marvels**.

```

<?php
// Cabecera de resultado en JSON.
header('Content-Type: application/json');

// Constantes de configuración de la aplicación.

define('DB_SERVIDOR', 'localhost');
define('DB_PUERTO', '3306');
define('DB_BASEDATOS', 'marvel');
define('DB_USUARIO', 'marvel');
define('DB_PASSWORD', 'abc123');

try {
    $cadenaConexion = "mysql:host=" . DB_SERVIDOR . ";port=" . DB_PUERTO . ";dbname=" . DB_BASEDATOS . ";charset=utf8";
    $pdo = new PDO($cadenaConexion, DB_USUARIO, DB_PASSWORD);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $pdo->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    die("Error conectando a servidor de base de datos: " . $e->getMessage());
}

try {
    // Creamos un objeto para componer el resultado luego en JSON
    $objeto = new stdClass();

    if (isset($_GET['id']) && $_GET['id'] != '') {
        // Preparamos la consulta de SELECT
        $stmt = $pdo->prepare('select * from marvels where ID=?');
        // Vinculamos los parámetros.
        $stmt->bindParam(1, $_GET['id']);
        // Ejecutamos la consulta
        $stmt->execute();

        // Si encontró el registro de la consulta inicial
        if ($stmt->rowCount() == 1) {
            $objeto->total = $stmt->rowCount();
            $objeto->resultados = $stmt->fetchAll();
        } else {

```

```

        $objeto->total = 0;
        $objeto->resultados = [];
    }

    echo json_encode($objeto);
} else {
    // No hay ningún registro con ese id o no se envía el ID.
    // Preparamos la consulta de SELECT
    $stmt = $pdo->prepare('select * from marvels');
    // Ejecutamos la consulta
    $stmt->execute();
    $objeto->total = $stmt->rowCount();
    $objeto->resultados = $stmt->fetchAll();

    echo json_encode($objeto);
}
} catch (PDOException $error) {
    echo "Error: " . $error->getMessage();
}
}

```

Ahora podemos crear un archivo html con un formulario que tenga un objeto 'select', donde cargaremos al iniciar la página el 'name' de todos los 'marvels' y un botón que nos permita pedir los datos del usuario seleccionado y escribirlos en una tabla los datos: "Name, Alignment, Hometown, Gender, Fighting Skills".

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="estilos.css">
    <title>Web Marvel</title>
</head>
<body>
    <h1>Web Marvel</h1>
    <form action="#" id="formulario">
        <select name="personajes" id="personajes"></select>
        <input type="submit" value="Pedir datos">
    </form>

    <table id="tabla">
        <thead>
            <th>Name</th>
            <th>Alignment</th>
            <th>Hometown</th>
            <th>Gender</th>
            <th>Fighting Skills</th>
        </thead>
    </table>

    <script src="scripts.js"></script>

</body>
</html>

```

Con un CSS sencillo para dar formato a la tabla.

```

table {
    margin-top: 2em;
    border-collapse: collapse;
}

th, td, tr {
    border: 1px solid black;
    text-align: center;
}

```

El archivo 'scripts.js' tendrá el siguiente código.

```

//Vamos a crear una función que rellene los campos del objeto select

```

```

function getData(id) {

    let xhr = new XMLHttpRequest();
    const numAleatorio = (new Date()).getTime();
    // console.log(numAleatorio);

    //Si llamamos a getData() sin pasar ningún 'id' para rellenar el select
    if (id == undefined) {
        let url = `marvel.php?r=${numAleatorio}`
        // console.log(url);
        xhr.open('GET',url,true);

        xhr.addEventListener('load', function(datos) {
            //Creamos un json con todos los resultados
            const jsonDatos = JSON.parse(datos.target.response);
            // console.log(jsonDatos.resultados);

            //Rellenamos el select
            //<option value="1">Spider Man</option>
            for (const heroes of jsonDatos.resultados) {
                // console.log(heroes);
                const option = document.createElement('option');
                option.setAttribute('value', heroes.ID);
                option.textContent = heroes.name;
                personajes.appendChild(option);
            }
        });

        xhr.send();

        //Si llamamos a getData(id) con un id de la tabla
    } else {
        // console.log(id);
        let url=`marvel.php?id=${id}&r=${numAleatorio}`
        console.log(url);
        xhr.open('GET',url,true);

        xhr.addEventListener('load', function(datos) {
            const jsonDatos = JSON.parse(datos.target.response);
            // console.log(jsonDatos);

            //Creamos la fila de datos
            const aCampos = ['name',
                'alignment',
                'hometown',
                'gender',
                'fighting_Skills'];

            const tr = document.createElement('tr');
            for (const campo of aCampos) {
                // console.log(campo);
                // console.log(jsonDatos.resultados[0][campo]);
                const td = document.createElement('td');
                td.textContent = jsonDatos.resultados[0][campo];
                tr.appendChild(td);
            }
            //Borramos el tr existente antes de poner el del elemento seleccionado
            if (tabla.children[1]) tabla.removeChild(tabla.children[1]);
            //Añadimos el elemento seleccionado a la tabla
            tabla.appendChild(tr);
        });

        xhr.send();
    }

}

//Seleccionamos los elementos html que vamos a utilizar
const tabla = document.getElementById('tabla');
const personajes = document.getElementById('personajes');

```

```

const formulario = document.getElementById('formulario');

//Añadimos un escuchador al submit del formulario
formulario.addEventListener('submit', function(e) {
  e.preventDefault();
  // console.log(personajes.selectedIndex);
  // console.log(personajes.children);
  const IDSelect = personajes.children[personajes.selectedIndex].value;
  // console.log(IDSelect);
  getData(IDSelect);
  }, false);

//Llamamos a getData() sin 'id' para rellenar el 'select'
getData();

```

Una mejora interesante sería:

- Añadir a la web un *radio button* que selecciones los súper héroes por *alignment* (*Bad, Good, Neutral*), también ofrecer una opción de "Todos". Dependiendo de cual sea el *radio button* seleccionado, el elemento *select* sólo se llenará con los súper héroes correspondientes.

The screenshot shows a web form titled "Web Marvel". Under the heading "Elige Alignment", there are four radio buttons: "Todos", "Bad", "Neutral" (which is selected), and "Good". Below the radio buttons is a dropdown menu with a checkmark next to "Iron Man". Other options in the dropdown include "Hulk", "Magneto", "Emma Frost", "Punisher", "Venon", and "Juggernaut". To the right of the dropdown is a button labeled "Pedir datos". Below the form is a table with columns: "Name", "Hometown", "Gender", and "Fighting Skills".

Web con la mejora pedida 1

- Permitir añadir nuevos elementos a la Base de datos.
 - ◊ 'Alignment' : Mejor un select con 'Bad', 'Good' y 'Neutral'
 - ◊ 'Gender' : Mejor un select con 'Male' y 'Female'

The screenshot shows the updated "Web Marvel" form. The "Elige Alignment" section now has radio buttons for "Todos", "Bad", "Neutral", and "Good". Below this, there are two buttons: "Pedir datos" and "Añadir datos". At the bottom, there is a table with the following structure:

Name	Alignment	Hometown	Gender	Fighting Skills
Spider Man				

Web con la mejora pedida 2

- También permitir editar y eliminar elementos.

-> Solución a la mejora de la "Práctica 1 XMLHttpRequest ? GET + PHP ? MySQL"

Volver