

AJAX



**Claudio Flores
Sapiain**

Ingeniero Informático

Scrum Master

Ingeniero DevOps

2020

FORMAR

TRANSFORMAR

Ajax

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Con Ajax los datos adicionales se solicitan al servidor (JAVA, GO, PHP, C#, etc) y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

Llamada

```
$.ajax({  
    cache: false,  
    // puede ser GET, POST  
    type: "POST",  
    // Tipo de retorno  
    dataType: "html",  
    // pagina php que recibe la llamada  
    url: "logica.php?" + (new Date().getTime()),  
  
    // datos, ej: $_POST['data']  
    data: {  
        n1:nro1,  
        n2:nro2,  
        n3:nro3  
    },  
},
```

Llamada

```
beforeSend: function(){  
  
    },  
    // acciones cuando me retorna algo el PHP  
    success: function( msg){  
        console.log(msg);  
    }  
    },  
    // acciones cuando hay error en comunicacion el el php  
    error: function(xhr, msg2 ){  
  
        console.log(msg2);  
    }  
});//fin ajax
```

Estructura

cache: Establece si la petición será guardada en la cache del navegador. De forma predeterminada es true para todos los *dataType* excepto para script y jsonp. Cuando posee el valor false, se agrega una cadena de caracteres anti-cache al final de la URL de la petición.

type: De forma predeterminada su valor es GET. Otros tipos de peticiones también pueden ser utilizadas (como PUT y DELETE), sin embargo pueden no estar soportados por todos los navegadores.

dataType : Establece el tipo de información que se espera recibir como respuesta del servidor. Si no se especifica ningún valor, de forma predeterminada, jQuery revisa el tipo que posee la respuesta

Estructura

url : Establece la URL en donde se realiza la petición. La opción url es obligatoria para el método \$.ajax;

data : Establece la información que se enviará al servidor. Esta puede ser tanto un objeto como una cadena de datos (por ejemplo foo=bar&baz=bim)

beforeSend : acciones que se hacen desde que se hace la llamada hasta que retorna.

success : Establece una función a ejecutar si la petición a sido satisfactoria. Dicha función recibe como argumentos la información de la petición (convertida a objeto JavaScript en el caso que dataType sea JSON), el estatus de la misma y el objeto de la petición en crudo.

Estructura

error : Establece una función de devolución de llamada a ejecutar si resulta algún error en la petición. Dicha función recibe como argumentos el objeto de la petición en crudo y el código de estatus de la misma petición.

Try - Catch

Una cláusula catch es parte de una declaración try-catch y tiene la forma siguiente.

```
try
{ Declaraciones
catch (EXCEPCION1 e1)
{ Declaraciones
}
catch (EXCEPCION2 e2)
{ Declaraciones
```

```
finally
{ Declaraciones
}
```


Try - Catch

try es el inicio de la función, dentro de try va el código que vamos a procesar, en caso de que haya un error, la que sigue al terminar catch, atrapará la excepción y de ahí podremos mostrar los datos de la misma (de la excepción).

El código de dentro del bloque finally siempre se ejecutará después de los bloques try y catch, independientemente de que se haya lanzado una excepción o no, y antes de que la ejecución normal continúe.

Try - Catch

```
function valida() {  
    var msj, x;  
    msj = document.getElementById("msj");  
    msj.innerHTML = "";  
    x = document.getElementById("x").value;  
    try {  
        if(x == "") throw "vacío";  
        if(isNaN(x)) throw "no es un nro!";  
        x = Number(x);  
        if(x < 5) throw "menor que 5";  
        if(x > 10) throw "mayor que 5";  
    }  
    catch(err) {  
        msj.innerHTML = "Ha ocurrido un error: " + err;  
    }  
}
```

BIBLIOGRAFÍA AJAX

<http://www.tutorialsteacher.com/jquery/jquery-ajax-method>

http://librosweb.es/libro/fundamentos_jquery/capitulo_7/metodos_ajax_de_jquery.html



Preguntas...

FORMAR

TRANSFORMAR