

AJAX

Ángel Villalba Fdez-Paniagua

Table of Contents

| | |
|------------------------|---|
| 1. Introducción | 1 |
| 2. XMLHttpRequest..... | 2 |
| 2.1. Ejercicio 1 | 3 |

1. Introducción

AJAX (Asynchronous JavaScript and XML) es una técnica que nos permite cargar datos en nuestra página HTML sin la necesidad de refrescarla. Normalmente, los datos que se cargan se envían en formato JSON.

Al poder cargar el contenido en la página sin la necesidad de recargarla, estamos mejorando la experiencia de usuario porque este no va a tener que esperar a que toda la página HTML se cargue, sino que solo esperará a que se actualicen los datos en una parte del contenido de la página.

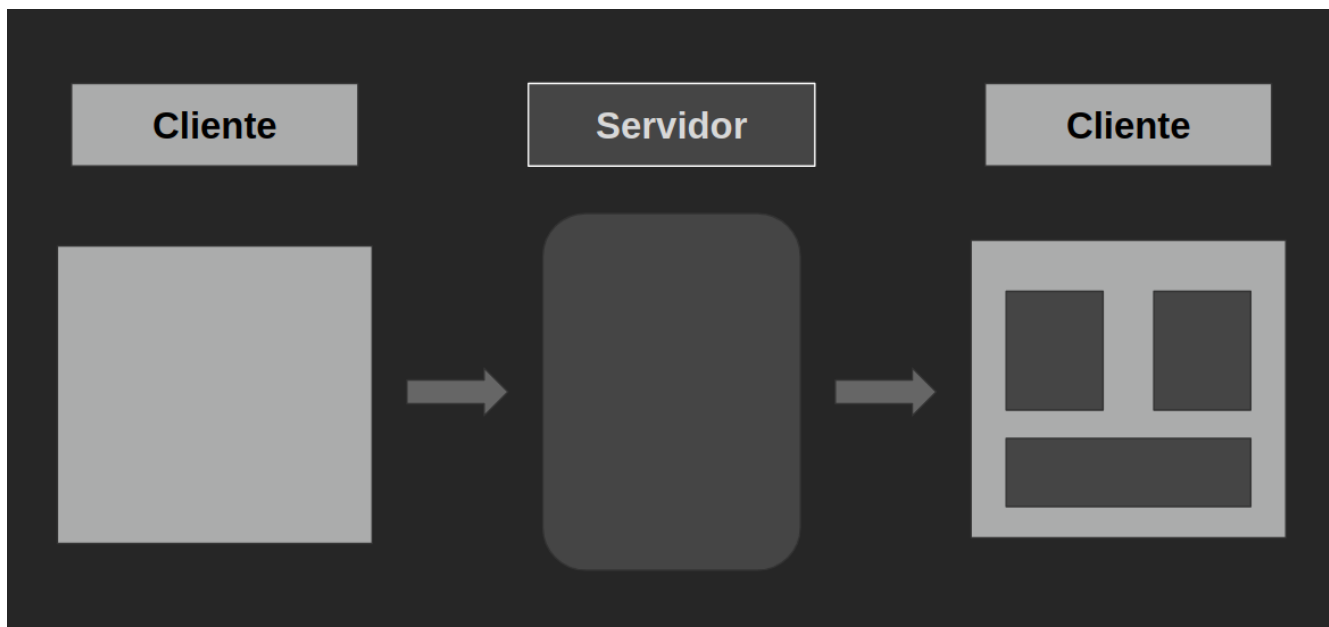
Muchos sitios de los que visitamos usan AJAX y es posible que no nos hayamos dado cuenta de ello. Por ejemplo, cuando añadimos un producto a una cesta de la compra, podemos ver como se actualiza el número de productos que aparecen en la cesta. Otro ejemplo que vemos muy a menudo, es cuando escribimos en una barra de búsqueda y esta nos empieza a mostrar las posibles búsquedas que podemos querer hacer. Otro más puede ser cuando añadimos en alguna red social una nueva publicación.

AJAX permite al usuario realizar otras acciones mientras el navegador está esperando a que los datos se carguen, y esto hace mejor la experiencia de usuario.

Cuando el navegador se encuentra con una etiqueta `<script>`, parará de procesar el resto de la página hasta que ese script haya terminado. Cuando una página que se está cargando necesita cargar datos del servidor, no deberíamos de esperar hasta obtenerlos porque no sabemos cuanto tiempo puede llevar, por lo que usando AJAX, podremos pedir esos datos, mientras el servidor los prepara y nos los envía, el navegador seguirá cargando lo necesario. Cuando el servidor nos envíe los datos que hemos pedido, se disparará un evento que llamará a la función encargada de procesar esos datos. Un ejemplo sería cuando entramos en un ecommerce, y nos aparecen todos los productos.

Una vez cargada la página, el usuario puede realizar acciones que requieran actualizar los datos de la interfaz. Aquí ocurre lo mismo que hemos visto antes. El navegador enviará la petición al servidor, y mientras el servidor prepara los datos, el usuario podrá seguir haciendo cosas en la página. Una vez que el servidor nos envía los datos, se emitirá el evento y se ejecutará la función asignada a ese evento para que muestre los datos. Un ejemplo sería cuando rellenamos un formulario, y enviamos los datos para guardarlos.

AJAX es el acrónimo de *Asynchronous JavaScript and XML* que eran las tecnologías usadas para poder realizar estas operaciones asíncronas de las que hemos hablado. Pero con el tiempo, las tecnologías han ido cambiando y ahora a lo que se refiere el termino de AJAX, es al conjunto de las tecnologías que permiten trabajar con operaciones asíncronas en el navegador.



2. XMLHttpRequest

Todas las aplicaciones AJAX tienen que usar el objeto XMLHttpRequest que es el que nos va a permitir realizar la comunicación con el servidor, en segundo plano y sin recargar la página.

A continuación se muestran los métodos y propiedades de XMLHttpRequest necesarios para realizar estas peticiones:

Table 1. Propiedades de XMLHttpRequest

| Propiedad | Descripción |
|--------------|---|
| readyState | Devuelve el estado de XMLHttpRequest: 0 (no inicializado), 1 (conexión con el servidor establecida), 2 (petición recibida), 3 (procesando la petición), 4 (petición finalizada) |
| responseText | Devuelve el contenido de la respuesta del servidor en forma de texto |
| status | Devuelve el estado de la petición HTTP: 200, 403, 404... |
| statusText | Devuelve el mensaje del estado de la petición HTTP: <i>Ok, Forbidden, Not found...</i> |

Table 2. Métodos de XMLHttpRequest

| Método | Descripción |
|-------------------|--|
| open(metodo, url) | Inicializa los parámetros necesarios para crear la petición. Estos parámetros son el método (<i>GET, POST...</i>) y la url . |
| send(datos) | Realiza la petición HTTP al servidor. En caso de la petición <i>POST</i> , se envían los datos como parámetro |

| Método | Descripción |
|--------------------|---|
| onreadystatechange | Cada vez que haya algún cambio en las propiedades del estado (readyState) de la petición, ejecuta la función que se le asigna |
| abort() | Detiene la petición actual |

Los pasos a seguir para realizar una petición son los siguientes:

- Creamos el objeto XMLHttpRequest
- Inicializar los parámetros de la petición usando el método `open()`
- Definir la función que se tiene que ejecutar cuando recibamos la respuesta, usando `onreadystatechange`
- Enviar la petición usando el método `send()`

```
function cargarContenido() {
  var xhr = new XMLHttpRequest();
  xhr.open('GET', 'datos.json');

  xhr.onreadystatechange = function() {
    if (xhr.status == 200 && xhr.readyState == 4) {
      var datos = JSON.parse(xhr.responseText);
      console.log(datos);
    }
  }

  xhr.send();
}
```

2.1. Ejercicio 1

- Usar AJAX para traer los datos de los ingresos y gastos desde Firebase
- Mostrar esos datos en una lista
 - Los ingresos tienen que mostrarse con el fondo verde
 - Los gastos tienen que mostrarse con el fondo rojo
- Crear un formulario y un boton para añadir nuevos datos
 - Al guardar un nuevo dato hay que volver a pintar los datos