

7.A. Programación AJAX en JavaScript

2.- Envío y recepción de datos de forma asíncrona.

Ahora que conocemos el objeto `XMLHttpRequest`, con sus propiedades y métodos, nos vamos a centrar en cómo se realiza la petición al servidor de forma asíncrona, y cómo se gestionan los estados y las respuestas que nos devuelve. También veremos los formatos de que disponemos para enviar datos al servidor, y en cuales se van a recibir los resultados.

Dependiendo del destino de los datos que vamos a recibir de forma asíncrona desde el servidor es recomendable el uso de uno u otro formato de los dos que recomendamos aquí:

Si los datos que recibimos los vamos a insertar directamente en el documento (a través del DOM), una opción interesante es formato XML, de esta forma el objeto recibido ya tiene la estructura de elementos del DOM y es fácil insertarlo en otro nodo del árbol.

En cambio si los datos recibidos van a ser procesados por nuestra aplicación escrita en JavaScript, es más recomendable utilizar el formato JSON. Dicho formato, utiliza la nomenclatura para definir objetos del lenguaje JavaScript y por tanto, se puede utilizar dichos resultados directamente en la aplicación JavaScript, sin tener que realizar prácticamente ningún tipo de conversiones intermedias.

En el apartado 1.5., hemos visto un ejemplo de una petición asíncrona al servidor, pero vimos que éramos incapaces de mostrar correctamente el resultado en el contenedor resultados.



```
function cargarAsync(objeto, url) {

    if (miXHR) {

        alert("Comenzamos la petición AJAX");

        //Si existe el objeto  miXHR

        miXHR.open('GET', url, true); //Abrimos la url, true=ASINCRONA

        // Hacemos la petición al servidor. Como parámetro:

        // null -> cuando usamos GET.

        // cadena con los datos -> cuando usamos POST

        miXHR.send(null);

        //Escribimos la respuesta recibida de la petición AJAX en el objeto DIV

        textoDIV(objeto, miXHR.responseText);

        alert("Terminó la petición AJAX");

    }

}
```

Si ejecutas el ejemplo del apartado 1.5, verás que no se muestra nada en el contenedor resultados, y la razón es por que, cuando accedemos a la propiedad `miXHR.responseText`, ésta no contiene nada. Eso es debido a la solicitud asíncrona. Si recuerdas, cuando hicimos el ejemplo con una solicitud síncrona, se mostró la alerta de "Comenzamos la petición AJAX", aceptaste el mensaje, y justo 2 segundos después recibimos la alerta de

“Terminó la petición AJAX”. En el modo **síncrono**, el navegador cuando hace la petición al servidor, con el método `miXHR.send()`, se queda esperando hasta que termine la solicitud (y por lo tanto nosotros no podemos hacer otra cosa más que esperar ya que el navegador está bloqueado). Cuando termina la solicitud, pasa a la siguiente línea: `textoDIV(objeto,...)`, y por tanto ya puede mostrar el contenido de `responseText`.

En el modo asíncrono, cuando aceptamos la primera alerta, prácticamente al instante se nos muestra la siguiente alerta. Ésto es así, por que el navegador en una petición asíncrona, no espera a que termine esa solicitud, y continúa ejecutando las siguientes instrucciones. Por eso, cuando se hace la llamada con el método `miXHR.send()`, dentro de la función `cargarAsync()`, el navegador sigue ejecutando las dos instrucciones siguientes, sin esperar a que termine la solicitud al servidor. Y es por eso que no se muestra nada, ya que la propiedad `responseText`, no contiene ningún resultado todavía, puesto que la petición al servidor está todavía en ejecución. **¿Cuál será entonces la solución?**

Una primera solución que se nos podría ocurrir, sería la de poner un tiempo de espera o retardo, antes de ejecutar la instrucción `textoDIV`. Ésto, lo único que va a hacer, es bloquear todavía más nuestro navegador, y además, tampoco sabemos exactamente lo que va a tardar el servidor web en procesar nuestra solicitud.

La segunda solución, consistiría en detectar cuándo se ha terminado la petición AJAX, y es entonces en ese momento, cuando accederemos a la propiedad `responseText` para coger los resultados. Ésta será la solución, que adoptaremos y que veremos en el apartado 2.1 de esta unidad.

Créditos de la imagen

Autoría:

Licencia: CC BY 2.0.

◀ DWEC07 Solucion tarea

Ir a...

7.B. Introducción a AJAX en W3Schools ▶