

Desenvolupament d'aplicacions web

Desenvolupament web a l'entorn client

UT 7.1 : AJAX

Índex de continguts

XMLHTTP.....	3
XMLHttpRequest.....	3
Seqüència d'una petició.....	4
Mètodes de l'objecte XMLHttpRequest.....	5
Propietats de l'objecte XMLHttpRequest.....	6
Esdeveniments de l'objecte XMLHttpRequest.....	7
Exemple.....	8

XMLHTTP

L'any 1998 Microsoft estava desenvolupant el seu servidor Exchange 2000, que incloïa un client via web. L'equip de desenvolupament tenia dues alternatives:

- Pàgines completament estàtiques.
- Pàgines web dinàmiques.

Els desenvolupadors es varen decantar per aquesta darrera opció, utilitzant *DHTML*, bàsicament HTML i JavaScript. De totes maneres, al cap de l'equip, Alex Hopmann, aquella solució no l'acabava de convèncer. Cada acció de l'usuari significava enviar un formulari al servidor i tornar a carregar la pàgina completa.

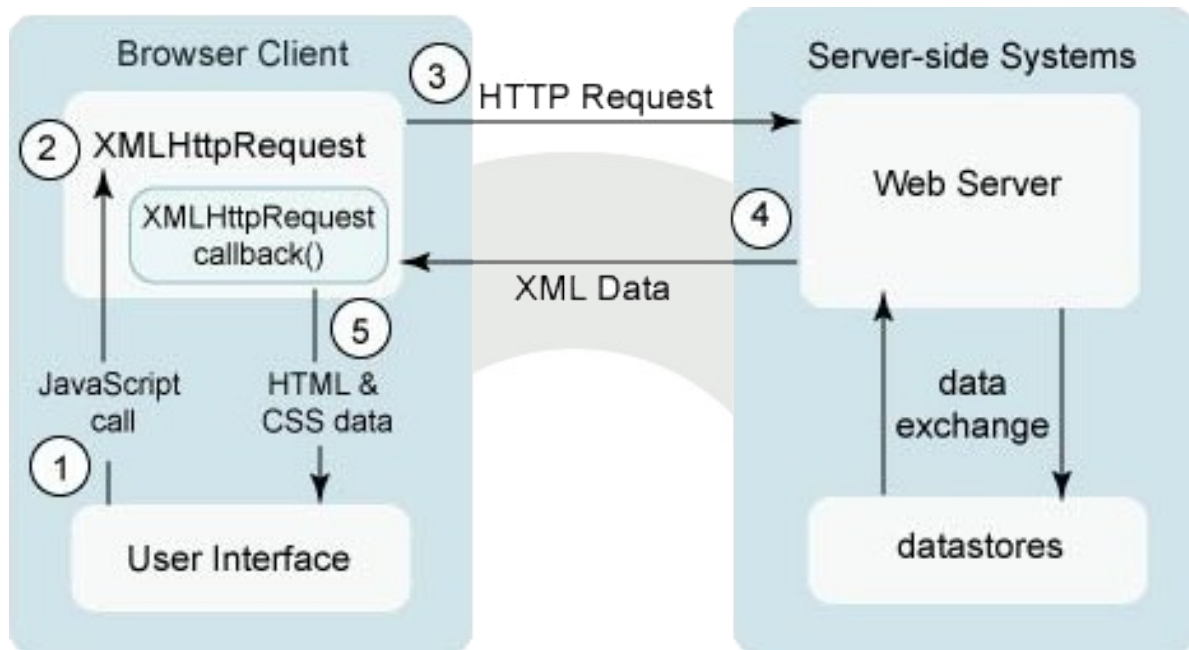
L'equip va desenvolupar una llibreria, *XMLHTTP*, que inclou l'objecte *XMLHttpRequest*. Aquest objecte permet fer peticions al servidor i rebre la resposta de manera que pugui ser processada al costat del client. A més pot fer-ho de forma asíncrona, és a dir, s'envia la petició, l'usuari pot seguir treballant en la pàgina, i en arribar la resposta es processa, modificant per exemple algun element de la pàgina.

XMLHttpRequest

Com hem dit l'objecte *XMLHttpRequest* és la peça central de la llibreria *XMLHTTP*. Permet intercanviar informació entre el client i el servidor de forma asíncrona i sense que l'usuari ho noti. Ens permetrà:

- Actualitzar una pàgina sense haver de carregar-la tota de bell nou,
- demanar dades al servidor una vegada la pàgina s'ha carregat,
- rebre dades del servidor una vegada la pàgina s'ha carregat,
- enviar dades al servidor en segon pla...

Seqüència d'una petició



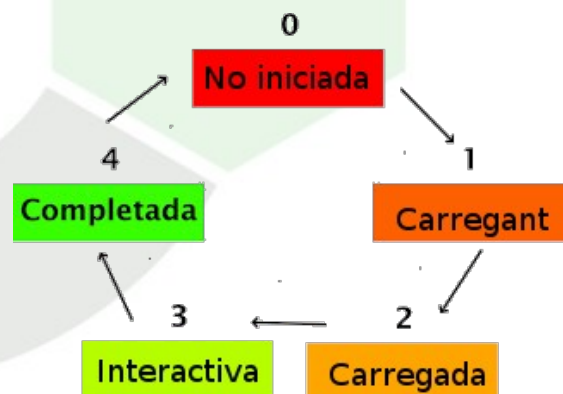
1. L'usuari provoca una crida a l'objecte.
2. Es crea l'objecte.
3. Es realitza la petició.
4. El servidor la processa i torna la resposta.
5. En rebre la resposta, el client executa la funció assignada a l'objecte. La funció interpretarà les dades i normalment provocarà la modificació de l'HTML i el CSS que veu l'usuari.

Mètodes de l'objecte XMLHttpRequest

- **.open(mètode, url [, async, usuari, password]):** Els tres darrers paràmetres són opcionals. Prepara la petició. S'ha de dir obligatòriament el mètode (GET, POST, ...) i la url. `async=true`, per defecte, obliga a la petició a realitzar-se de forma asíncrona, és a dir, l'execució de l'aplicació segueix i en arribar la resposta es tractarà. Amb `false` s'executa síncronament, el client queda "congelat" fins que arribi la resposta.
- **.send(dades):** Envia la petició al servidor, afegint-hi el paràmetre. El paràmetre pot ser, per exemple, una cadena de text amb els paràmetres d'un POST, un array de bytes, un objecte XML DOM, ...
- **.abort():** cancel·la la petició actual(en esgotar-se un timeout, per exemple).
- **.setRequestHeader("capçalera", "valor"):** afegeix un parell etiqueta/valor a la capçalera de la petició.
- **.getResponseHeader(string):** torna un valor de la capçalera de la resposta.
- **.getAllResponseHeaders():** torna una cadena amb tots els valors de la capçalera de la resposta.

Propietats de l'objecte XMLHttpRequest

- **.responseText**: Emmagatzema el contingut de la resposta del servidor en forma de cadena de text (en JSON per exemple).
- **.responseXML**: Emmagatzema el contingut de la resposta del servidor en format XML que es pot tractar com un objecte DOM
- **.status**: El codi d'estat HTTP tornat per el servidor(200 OK, 404 Not found, ...)
- **.statusText**: El text corresponent a l'estat.
- **.readyState**: Valor sencer que indica l'estat de la petició:
 - **0** No iniciada: L'objecte s'ha creat però no s'ha invocat el mètode open.
 - **1** Carregant: L'objecte s'ha creat, s'ha executat open, però no send
 - **2** Carregada: S'ha invocat send, però el servidor no ha respost
 - **3** Interactiva: S'està rebent la resposta del servidor.
 - **4** **Completada**: S'ha rebut la resposta del servidor i es pot tractar a partir de *responseText* i *responseXML*.



Esdeveniments de l'objecte XMLHttpRequest

- **.onreadystatechange**: Es dispara quan es produeix un canvi en l'estat de la petició HTTP. Normalment l'associarem a una funció JavaScript que serà l'encarregada d'actuar.

Aquesta funció hauria de comprovar en quin estat es troba la petició, quin codi de sortida ha tornat, ...

Per exemple, el més normal és que l'únic estat que ens interressi sigui el 4, complet. Llavors si és aquest estat hem de comprovar quin codi ens ha enviat el servidor:

200 - 299 Tot ha anat bé, podem processar la resposta

Qualsevol altre codi: hi ha hagut problemes. El desenvolupador ha de decidir com actuar. Segons el codi d'error i el possible contingut de la resposta podrem saber que ha passat i reaccionar en conseqüència.

Exemple

Aquí teniu un exemple complet d'una funció que demana dades a una api REST, els llibres d'un autor determinat, i en rebre aquestes dades les incorpora a la pàgina.

```
// Mostra a la pàgina els llibres de l'autor seleccionat.
function carregaLlibres(idAutor) {
    const llibres = document.getElementById("llibres");
    // Elimina el contingut del div que mostra els llibres
    esborra(llibres);
    //Crea l'objecte
    const req = new XMLHttpRequest();
    // Assigna la funció a l'esdeveniment per reaccionar als canvis.
    req.addEventListener("readystatechange", function () {
        // Si ha finalitzat la petició
        if (req.readyState == 4) {
            //Si l'estat és OK
            if (req.status >= 200 && req.status<=299) {
                // Recuperam les dades tornades per el servidor
                const resultat = JSON.parse(req.responseText);
                // Inclou els llibres a la pàgina
                mostraLlibres(resultat);
            }
            // Si no ha anat bé, el codi és distint de 200
        } else {
            // En aquest cas concret el servidor ens torna un JSON amb un
            // missatge indicant l'error i el mostrem a la pàgina.
            // El tractament de l'error podria ser qualsevol altre.
            const missatge = JSON.parse(req.responseText);
            llibres.append(document.createTextNode(missatge.error));
        }
    });
}
```



```
    }  
    });  
  
    //Prepara la connexió a u ip/autors/32/llobres  
    req.open("GET", url + "autors/" + idAutor + "/llibres");  
    //Envia la petició  
    req.send();  
}
```