Programación AJAX en JS

Módulo: **Desarrollo Web en entorno cliente**

CFGS Desarrollo de Aplicaciones Web

¿Qué es AJAX?

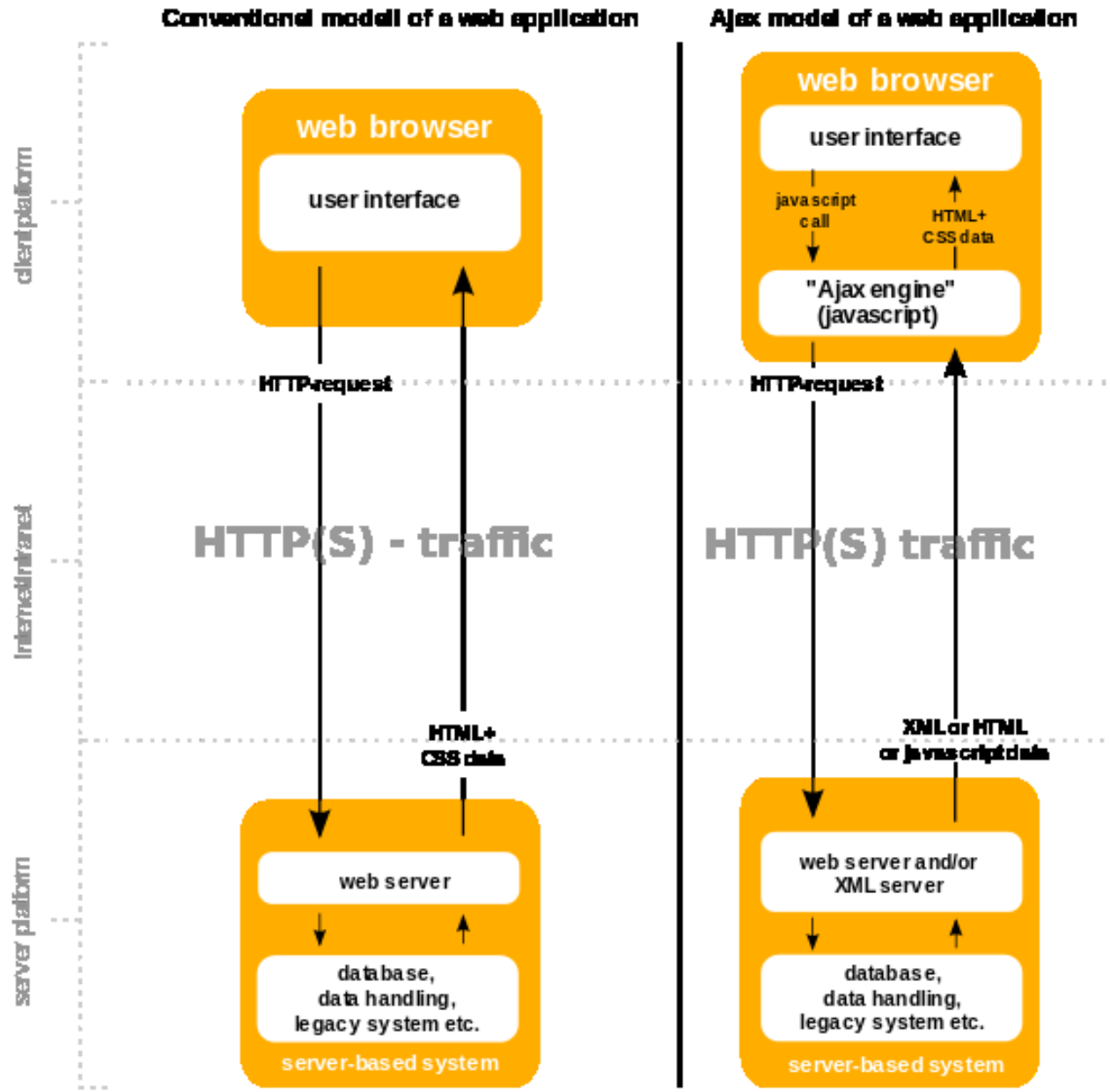
- Acrónimo de Asynchronous JavaScript And XML.
- RIA - Rich Internet Application Technoglogies – como son Applet, Adobe Flash, Java WebStart, DHTML, AJAX.
- Los datos se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.
- Se pueden realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad.
- No es un lenguaje de programación sino un conjunto de tecnologías:

DHTML = HTML + JS + DOM + CSS

AJAX = DHTML – XMLHttpRequest –

PHP/ASP.NET/JSP/NodeJS/Python/… – XML/JSON/texto plano/…

¿Qué es AJAX?


Conventional model of a web application
Ajax model of a web application
client platform
Internet/intranet
server platform
web browser
user interface
HTTP request
HTML+ CSS data
web server
database, data handling, legacy system etc.
server-based system
HTTP(S) - traffic
web browser
user interface
javascript call
HTML+ CSS data
"Ajax engine" (javascript)
HTTP request
XML or HTML or javascript data
web server and/or XML server
database, data handling, legacy system etc.
server-based system
HTTP(S) traffic

¿Qué es AJAX?

Modelo Tradicional

1. El cliente le hace una petición (HTTP request) al servidor Web
2. El servidor WEB, procesa la petición dinámicamente (PHP, ASP.NET, Python,…)
3. Una vez procesada la petición HTTP por el servidor WEB, este le devuelve una respuesta en HTML+CSS al cliente.
4. Por medio de esa página generalmente el ciclo vuelve a empezar, ya que el cliente puede hacer otras peticiones HTTP al servidor, que a su vez van a ser procesadas y el servidor WEB va a devolver otros resultados HTML, etc.

Modelo AJAX

Con AJAX, cuando el cliente hace una petición HTTP al servidor, la hace por medio de JS. El servidor procesa la petición y en vez de devolverle al cliente una página HTML, le devuelve un resultado en XML/texto plano/JSON, que es procesado por JS, y este actualiza solo las secciones de la página necesarias (sin tener que cargar una nueva página).

Ventajas

- Usa tecnologías existentes.
- Soportado por la mayoría de los navegadores actuales.
- Mejora la interactividad ya que no es necesario que los datos viajen al servidor.
- Más rápida porque cada vez que se necesita actualizar un dato en una página, no es necesario recargar toda la página, solo la sección necesaria.
- Más amigable para el usuario ya que se asemeja a las aplicaciones de escritorio.

Desventajas

- "Back" de los navegadores. Al estar siempre en la misma página podemos llegar a confundir a los usuarios ya que el botón volver de los navegadores no nos llevaría al paso anterior que ellos han visto, sino a la página anterior.
- El navegador tiene que soportar JS y tenerlo habilitado.
- Como se ejecuta más código en el lado del cliente puede interferir en el rendimiento de este.
- Política del mismo origen (problemas entre nombres de dominios)
- No es compatible con todos los lectores de pantalla y puede que no funcione correctamente en dispositivos móviles, etc.

XMLHttpRequest - Métodos

Métodos	Descripción
open(método, URL, banderaAsync, nombreuser, password)	Según el método (GET o POST) y la URL se prepara la petición. Si la banderaAsync=true Petición asíncrona, si es fase la petición es síncrona. nombreuser y password solo se usan para acceder a recursos protegidos.
send(contenido)	Ejecuta la petición, donde la variable contenido son datos que se envían al servidor.
abort()	Para la petición que está procesando.
getAllResponseHeaders()	Devuelve todas las cabeceras de la llamada HTTP como un string.
getResponseHeader(cabecera)	Devuelve la cabecera identificada por la etiqueta.
setRequestHeader(etiqueta, valor)	Establece el valor de una etiqueta de las cabeceras de petición antes de que se haga la petición.

XMLHttpRequest – Propiedades

Propiedades	Descripción
status	Código devuelto por el servidor
statusText	Texto que acompaña al código
responseText	Datos devueltos formato string
responseXML	Datos devueltos formato Objeto XML
readyState	Estado actual de la petición. 0: Sin iniciar 1: Cargando
	2: Cargado 3: Interactivo (algunos datos devueltos) 4: Completado
onreadystatechange	Puntero a la función del manejador que se llama cuando cambia readyState.

Ver ejemplos (1, 2, 3)

Método POST

- Se usa cuando queremos enviar mucha información al servidor.
- Al abrir la conexión se establece POST como primer parámetro del XMLHttpRequest.
- A continuación, llamamos al método setRequestHeader indicando qué tipo de datos enviamos y al método send pasándole los datos.

```
conexion.open('POST','pagina.php', true);
conexion.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
conexion.send("nombre=juan&apellido=acosta")
```

responseXML

- Recupera los datos como XML y se deben recorrer mediante DOM
- Es necesario que desde el servidor se remitan los datos como XML

Ver ejemplo 4

status y statusText

- status: almacena el código del estado de la petición HTTP. Algunos de los valores más usados son 200 (conexión exitosa) y 404 (página inexistente).
- statusText: almacena el texto de la petición HTTP enviado por el servidor

Lo más correcto cuando readyState contiene el valor 4, se debe verificar si la propiedad status almacena el valor 200.

```
function procesarEventos() {
  var resultados = document.getElementById("resultados");
  if(conexion1.readyState == 4) {
    if (conexion1.status==200) {
      resultados.innerHTML = conexion1.responseText;
    }
    else {
      resultados.innerHTML='';
      alert(conexion1.statusText);
    }
  }
  else
   if (conexion1.readyState==1||conexion1.readyState==2||conexion1.readyState==3){
      resultados.innerHTML = 'Procesando...';
    }
}
```

Paginación

Ver ejemplo 5

Cargar un control de tipo select

Ver ejemplo 6

Saber más …

- **Wikipedia**: https://en.wikipedia.org/wiki/Ajax_(programming)
- **w3schools.com:** http://www.w3schools.com/ajax/default.asp
- **Mozilla Developer Network:** https://developer.mozilla.org/es/docs/AJAX