



AJAX Asynchronous Javascript and XML



—
\$("#que es?")

Es una técnica de
desarrollo web para
crear aplicaciones
interactivas



\$(".como?")

Estas aplicaciones se **ejecutan** en el **cliente**, es decir, en el navegador de los usuarios 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.



\$(".como?")

JavaScript es un lenguaje de programación (scripting language) en el que normalmente se efectúan las funciones de llamada de **Ajax** mientras que el acceso a los datos se realiza mediante **XMLHttpRequest**, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.



Como esta compuesto?

- HTML y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.



HTTP REQUEST COMUNES

GET

Método HTTP para obtener información de un servicio, no debería modificar información puede cachear llamada.

POST

Método HTTP para intercambio de información con el servidor, nunca debería *cachear* la llamada



Funcionalidades



```
$.ajax({  
  url: "htt..",  
  type: "POST",  
})
```

url: Define a que dirección apuntar el ajax, puede ser una web externa o interno de nuestro proyecto

type: Define el verbo http (GET, POST, PUT, etc) de la llamada por ajax.



```
$.ajax({  
  async: true,  
})
```

async: Define si la comunicación será asíncrona, por **default** es true



```
$.ajax({  
    success: function () {  
        // doSomething  
    }  
})
```

success: esa función se ejecuta a la vuelta de la llamada, en caso de ser exitosa la llamada, ejecuta lo que se encuentra dentro de la función



```
$.ajax({  
  error: function () {  
    // doSomething  
  }  
})
```

error: esa función se ejecuta a la vuelta de la llamada, en caso de haber error en la llamada, ejecuta lo que se encuentra dentro de la función



```
$.ajax({  
  timeout: XX  
})
```

timeout: Setea el tiempo, en milésimas de segundos, en la cual la conexión se corta.



```
$.ajax({  
  header: {...}  
})
```

header: se define el
conjunto
"clave" : "valor"
De la llamada ajax



Algunos problemas

- Al cargar páginas mediante ajax, no queda registro de navegación en el navegador, por lo que no podríamos "volver atrás"
- Las llamadas por ajax son más difíciles de ser indexadas por los buscadores
- Es posible que páginas con Ajax no puedan funcionar en teléfonos móviles, PDA u otros aparatos. Ajax no es compatible con todos los software para invidentes u otras discapacidades.



Navegadores que no permiten AJAX

- Opera 7 y anteriores
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores
- Anteriores a Safari 1.2
- Dillo
- Navegadores basados en texto como Lynx y Links
- Navegadores para personas con capacidades especiales visuales (Braille)
- Algunos navegadores de teléfonos móviles
- Navegador de la PSP



___Gracias!