

Backend > **Lección 2: Introducción al XMLHttpRequest**

## El objeto XMLHttpRequest, sus métodos y propiedades

### Métodos y propiedades del objeto XMLHttpRequest

En esta ocasión veremos cómo está compuesto este objeto y cuáles son los métodos y propiedades que se encargan de realizar la funcionalidad.

Si abrimos una consola en el navegador, escribimos XMLHttpRequest y pulsamos Enter, podremos ver que es una función que corresponde a código nativo JavaScript:

```
chrome-extension://fm...gofadop ▼ ☐ Preserve log
> XMLHttpRequest
< function XMLHttpRequest() { [native code] }
> |
```

Para poder ver su contenido, los métodos y las propiedades, es necesario instanciar un nuevo objeto a partir de esta función, el cual heredará todos su contenido. Esto lo hacemos de la siguiente manera:

```
> var miXHR = new XMLHttpRequest()
< undefined
> miXHR
< XMLHttpRequest {onreadystatechange: null, readyState: 0, timeout: 0, withCredentials: false, upload: XMLHttpRequestUpload...} ⓘ
  onabort: null
  onerror: null
  onload: null
  onloadend: null
  onloadstart: null
  onprogress: null
  onreadystatechange: null
  ontimeout: null
  readyState: 0
  response: ""
  responseText: ""
```

¡Chatea con un tutor!



U\_CODE

CSS

```

responseXML: null
status: 0
statusText: ""
timeout: 0
▶ upload: XMLHttpRequestUpload
withCredentials: false
▶ __proto__: XMLHttpRequest

```

## Resumen de los métodos y propiedades

### Propiedades

Propiedad	Descripción
readyState	Valor numérico (entero) que almacena el estado de la petición
responseText	El contenido de la respuesta del servidor en forma de cadena de texto
responseXML	El contenido de la respuesta del servidor en formato XML. El objeto devuelto se puede procesar como un objeto DOM
status	El código de estado HTTP devuelto por el servidor (200 para una respuesta correcta, 404 para "No encontrado", 500 para un error de servidor, etc.)
statusText	El código de estado HTTP devuelto por el servidor en forma de cadena de texto: "OK", "Not Found", "Internal Server Error", etc.

### Valores de la propiedad readyState

Valor	Descripción
0	No inicializado (objeto creado, pero no se ha invocado el método open)
1	Cargando (objeto creado, pero no se ha invocado el método send)
2	Cargado (se ha invocado el método send, pero el servidor aún no ha respondido)
3	Interactivo (se han recibido algunos datos, aunque no se puede emplear la propiedad responseText)
4	Completo (se han recibido todos los datos de la respuesta del servidor)

### Métodos

Método	Descripción
abort()	Detiene la petición actual
getAllResponseHeaders()	Devuelve una cadena de texto con to respuesta del servidor

¡Chatea con un tutor!



U\_CODE

CSS ▼	header("cabecera")	Devuelve una cadena de texto con el solicitada
	onreadystatechange	Responsable de manejar los eventos cada vez que se produce un cambio en el estado de la petición HTTP. Normalmente es una referencia a un objeto JavaScript.
	open("metodo", "url")	Establece los parámetros de la petición al servidor. Los parámetros necesarios son el método HTTP empleado y la URL destino (puede indicarse de forma absoluta o relativa)
	send(contenido)	Realiza la petición HTTP al servidor
	setRequestHeader("cabecera", "valor")	Permite establecer cabeceras personalizadas en la petición HTTP. Se debe invocar el método open() antes que setRequestHeader()



U\_CODE

CSS ▼

Copyright © 2016 Next U LLC.

[¿Quiénes somos?](#) [Contáctenos](#) [Condiciones de servicio](#) [Política de privacidad](#)