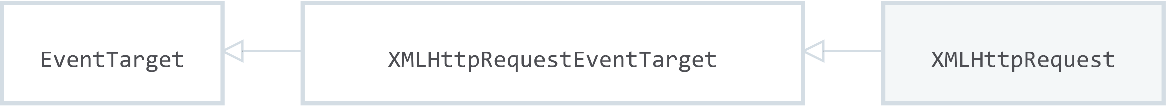
XMLHttpRequest

Use `XMLHttpRequest` (XHR) objects to interact with servers. You can retrieve data from a URL without having to do a full page refresh. This enables a Web page to update just part of a page without disrupting what the user is doing. `XMLHttpRequest` is used heavily in Ajax programming.



Despite its name, `XMLHttpRequest` can be used to retrieve any type of data, not just XML, and it supports protocols other than HTTP (including `file` and `ftp`).

If your communication needs involve receiving event or message data from the server, consider using server-sent events through the `EventSource` interface. For full-duplex communication, WebSockets may be a better choice.

Constructor

`XMLHttpRequest()`

The constructor initializes an XMLHttpRequest. It must be called before any other method calls.

Properties

This interface also inherits properties of `XMLHttpRequestEventTarget` and of `EventTarget`.

`XMLHttpRequest.onreadystatechange`

An `EventHandler` that is called whenever the `readyState` attribute changes.

`XMLHttpRequest.readyState` Read only

Returns an `unsigned short`, the state of the request.

`XMLHttpRequest.response` Read only

Returns an `ArrayBuffer`, `Blob`, `Document`, JavaScript object, or a `DOMString`, depending on the value of `XMLHttpRequest.responseType`. that contains the response entity body.

`XMLHttpRequest.responseText` Read only

Returns a `DOMString` that contains the response to the request as text, or `null` if the request was unsuccessful or has not yet been sent.

`XMLHttpRequest.responseType`

Is an enumerated value that defines the response type.

`XMLHttpRequest.responseURL` Read only

Returns the serialized URL of the response or the empty string if the URL is null.

`XMLHttpRequest.responseXML` Read only Not available to workers

Returns a `Document` containing the response to the request, or `null` if the request was unsuccessful, has not yet been sent, or cannot be parsed as XML or HTML.

`XMLHttpRequest.status` Read only

Returns an `unsigned short` with the status of the response of the request.

`XMLHttpRequest.statusText` Read only

Returns a `DOMString` containing the response string returned by the HTTP server. Unlike `XMLHTTPRequest.status`, this includes the entire text of the response message ("`200 OK`", for example).

> **Note:** According to the HTTP/2 specification (8.1.2.4 Response Pseudo-Header Fields), HTTP/2 does not define a way to carry the version or reason phrase that is included in an HTTP/1.1 status line.

`XMLHttpRequest.timeout`

Is an `unsigned long` representing the number of milliseconds a request can take before automatically being terminated.

`XMLHttpRequestEventTarget.ontimeout`

Is an `EventHandler` that is called whenever the request times out.

`XMLHttpRequest.upload` Read only

Is an `XMLHttpRequestUpload`, representing the upload process.

`XMLHttpRequest.withCredentials`

Is a `Boolean` that indicates whether or not cross-site `Access-Control` requests should be made using credentials such as cookies or authorization headers.

Non-standard properties

`XMLHttpRequest.channel` Read only

Is a `nsIChannel`. The channel used by the object when performing the request.

`XMLHttpRequest.mozAnon` Read only

Is a boolean. If true, the request will be sent without cookie and authentication headers.

`XMLHttpRequest.mozSystem` Read only

Is a boolean. If true, the same origin policy will not be enforced on the request.

`XMLHttpRequest.mozBackgroundRequest`

Is a boolean. It indicates whether or not the object represents a background service request.

`XMLHttpRequest.mozResponseArrayBuffer` 🗑 Obsolete since Gecko 6 Read only

Is an `ArrayBuffer`. The response to the request, as a JavaScript typed array.

`XMLHttpRequest.multipart` 🗑 Obsolete since Gecko 22

This Gecko-only feature, a boolean, was removed in Firefox/Gecko 22. Please use Server-Sent Events, Web Sockets, or `responseText` from progress events instead.

Event handlers

`onreadystatechange` as a property of the `XMLHttpRequest` instance is supported in all browsers.

Since then, a number of additional event handlers have been implemented in various browsers (`onload`, `onerror`, `onprogress`, etc.). See Using XMLHttpRequest.

More recent browsers, including Firefox, also support listening to the `XMLHttpRequest` events via standard `addEventListener()` APIs in addition to setting `on*` properties to a handler function.

Methods

`XMLHttpRequest.abort()`

Aborts the request if it has already been sent.

`XMLHttpRequest.getAllResponseHeaders()`

Returns all the response headers, separated by CRLF, as a string, or `null` if no response has been received.

`XMLHttpRequest.getResponseHeader()`

Returns the string containing the text of the specified header, or `null` if either the response has not yet been received or the header doesn't exist in the response.

`XMLHttpRequest.open()`

Initializes a request. This method is to be used from JavaScript code; to initialize a request from native code, use `openRequest()` instead.

`XMLHttpRequest.overrideMimeType()`

Overrides the MIME type returned by the server.

`XMLHttpRequest.send()`

Sends the request. If the request is asynchronous (which is the default), this method returns as soon as the request is sent.

`XMLHttpRequest.setRequestHeader()`

Sets the value of an HTTP request header. You must call `setRequestHeader()` after `open()`, but before `send()`.

Non-standard methods

`XMLHttpRequest.init()`

Initializes the object for use from C++ code.

> ❗ **Warning:** This method must *not* be called from JavaScript.

`XMLHttpRequest.openRequest()`

Initializes a request. This method is to be used from native code; to initialize a request from JavaScript code, use `open()` instead. See the documentation for `open()`.

`XMLHttpRequest.sendAsBinary()` 👎

A variant of the `send()` method that sends binary data.

Specifications

Specification	Status	Comment
XMLHttpRequest	LS Living Standard	Live standard, latest version