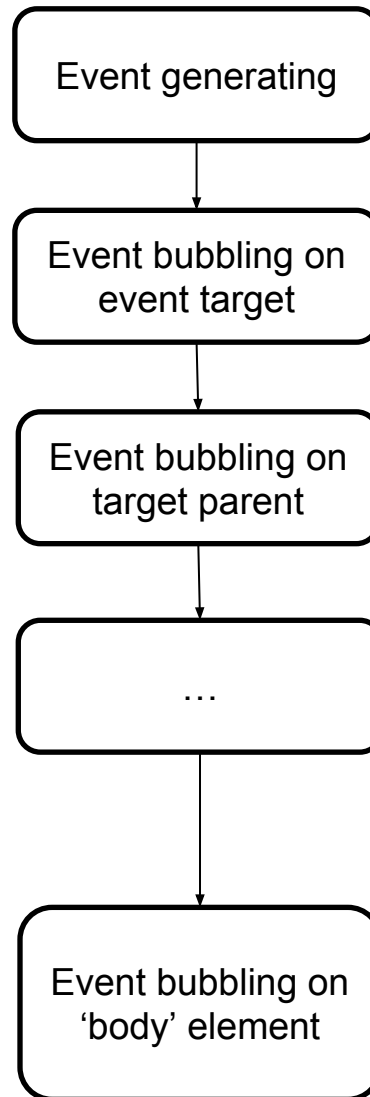


- ◆ UI events
- ◆ Focus events
- ◆ Mouse events
- ◆ Wheel events
- ◆ Input events
- ◆ Keyboard events
- ◆ Composition events



Properties

bubbles	Returns whether or not a specific event is a bubbling event
cancelBubble	Sets or returns whether the event should propagate up the hierarchy or not
cancelable	Returns whether or not an event can have its default action prevented
composed	Returns whether the event is composed or not
currentTarget	Returns the element whose event listeners triggered the event
defaultPrevented	Returns whether or not the preventDefault() method was called for the event
eventPhase	Returns which phase of the event flow is currently being evaluated
isTrusted	Returns whether or not an event is trusted
target	Returns the element that triggered the event
timeStamp	Returns the time (in milliseconds relative to the epoch) at which the event was created
type	Returns the name of the event
...	



Stop handling methods

- ◆ `preventDefault` – prevent default event
- ◆ `stopPropagation` – prevent bubbling
- ◆ `stopImmediatePropagation` – prevent bubbling and other handler

Event creation

1. `var event = new Event('build');`
2. `elem.addEventListener('build', function (e) { ... }, false);`
3. `elem.dispatchEvent(event);`

Custom event creation

```
1.  var event = new CustomEvent('build', { 'detail':  
    elem.dataset.time });  
2.  function eventHandler(e) {  
3.    log('The time is: ' + e.detail);  
4.  }
```

Call standart events



```
1. function simulateClick() {  
2.     var event = new MouseEvent('click', {  
3.         'view': window,  
4.         'bubbles': true,  
5.         'cancelable': true  
6.     });  
7.     var cb = document.getElementById('checkbox');  
8.     var canceled = !cb.dispatchEvent(event);  
9.     if (canceled) {  
10.        // A handler called preventDefault.  
11.        alert("canceled");  
12.    } else {  
13.        // None of the handlers called preventDefault.  
14.        alert("not canceled");  
15.    }  
16. }
```


- ◆ HTTP defines a set of request methods that indicate what the desired action will be performed for a given resource.
- ◆ Each implements its own semantics, but each group of commands shares common properties: methods can be safe, idempotent, or cached

1. GET

- The GET method requests a view of the resource. Queries using this method can only retrieve data.

2. POST

- POST is used to send entities to a specific resource. Often causes a state change or some side effects on the server.

3. PUT

- PUT replace all current representations of the resource request data.

4. DELETE

- DELETE deletes the specified resource.

1. HEAD

- HEAD requests the resource in the same way as the GET method, but without the response body

2. CONNECT

- CONNECT establishes a "tunnel" to the server defined by the resource

3. OPTIONS

- OPTIONS is used to describe resource connection parameters.

4. TRACE

- TRACE makes a call, return a text message with a resource.

5. PATCH

- PATCH is used to partially modify a resource

GET



Request has body	No
Successful response has body	Yes
Safe	Yes
Idempotent	Yes
Cacheable	Yes
Allowed in HTML forms	Yes

Request has body	No
Successful response has body	No
Safe	Yes
Idempotent	Yes
Cacheable	Yes
Allowed in HTML forms	No

POST



Request has body	Yes
Successful response has body	Yes
Safe	No
Idempotent	No
Cacheable	Only if freshness information is included
Allowed in HTML forms	Yes

PUT



Request has body	Yes
Successful response has body	No
Safe	No
Idempotent	Yes
Cacheable	No
Allowed in HTML forms	No

DELETE



Request has body	May
Successful response has body	May
Safe	No
Idempotent	Yes
Cacheable	No
Allowed in HTML forms	No

Request has body	No
Successful response has body	Yes
Safe	No
Idempotent	No
Cacheable	No
Allowed in HTML forms	No

OPTIONS



Request has body	No
Successful response has body	Yes
Safe	Yes
Idempotent	Yes
Cacheable	No
Allowed in HTML forms	No

Request has body	No
Successful response has body	No
Safe	No
Idempotent	Yes
Cacheable	No
Allowed in HTML forms	No

PATCH



Request has body	Yes
Successful response has body	Yes
Safe	No
Idempotent	No
Cacheable	No
Allowed in HTML forms	No

- Asynchronous JavaScript And XML is a programming practice of building more complex, dynamic webpages
- Using XHR

- 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**) - 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.

XHR object properties



- 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).
- 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.

XMLHttpRequest.readyState

Value	State	Description
0	UNSENT	Client has been created. <code>open()</code> not called yet.
1	OPENED	<code>open()</code> has been called.
2	HEADERS_RECEIVED	<code>send()</code> has been called, and headers and status are available.
3	LOADING	Downloading; <code>responseText</code> holds partial data.
4	DONE	The operation is complete.

[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()`.

- XMLHttpRequest.open(method, url)
- XMLHttpRequest.open(method, url, async)
- XMLHttpRequest.open(method, url, async, user)
- XMLHttpRequest.open(method, url, async, user, password)

MIME-TYPE

Resolution	Type	MIME
.aac	AAC audio file	audio/aac
.abw	AbiWord document	application/x-abiword
.arc	Archive document (multiple files embedded)	application/octet-stream
.avi	AVI: Audio Video Interleave	video/x-msvideo
.azw	Amazon Kindle eBook format	application/vnd.amazon.ebook
.bin	Any kind of binary data	application/octet-stream
.bz	BZip archive	application/x-bzip
.bz2	BZip2 archive	application/x-bzip2
.csh	C-Shell script	application/x-csh
.css	Cascading Style Sheets (CSS)	text/css
.csv	Comma-separated values (CSV)	text/csv
.doc	Microsoft Word	application/msword
.eot	MS Embedded OpenType fonts	application/vnd.ms-fontobject
.epub	Electronic publication (EPUB)	application/epub+zip
.gif	Graphics Interchange Format (GIF)	image/gif
.htm .html	HyperText Markup Language (HTML)	text/html
.ico	Icon format	image/x-icon
.ics	iCalendar format	text/calendar
.jar	Java Archive (JAR)	application/java-archive
.jpeg .jpg	JPEG images	image/jpeg
.js	JavaScript (ECMAScript)	application/javascript
.json	JSON format	application/json
.mid .midi	Musical Instrument Digital Interface (MIDI)	audio/midi
.mpeg	MPEG Video	video/mpeg
.mpkg	Apple Installer Package	application/vnd.apple.installer+xml
.odp	OpenDocument presentation document	application/vnd.oasis.opendocument.presentation

MIME-TYPE



Resolution	Type	MIME
.ods	OpenDocument spreadsheet document	application/vnd.oasis.opendocument.spreadsheet
.odt	OpenDocument text document	application/vnd.oasis.opendocument.text
.oga	OGG audio	audio/ogg
.ogv	OGG video	video/ogg
.ogx	OGG	application/ogg
.otf	OpenType font	font/otf
.png	Portable Network Graphics	image/png
.pdf	Adobe Portable Document Format (PDF)	application/pdf
.ppt	Microsoft PowerPoint	application/vnd.ms-powerpoint
.rar	RAR archive	application/x-rar-compressed
.rtf	Rich Text Format (RTF)	application/rtf
.sh	Bourne shell script	application/x-sh
.svg	Scalable Vector Graphics (SVG)	image/svg+xml
.swf	Small web format (SWF) or Adobe Flash document	application/x-shockwave-flash
.tar	Tape Archive (TAR)	application/x-tar
.tif	Tagged Image File Format (TIFF)	image/tiff
.ts	Typescript file	video/vnd.dlna.mpeg-tts
.ttf	TrueType Font	font/ttf
.vsd	Microsoft Visio	application/vnd.visio
.wav	Waveform Audio Format	audio/x-wav
.weba	WEBM audio	audio/webm
.webm	WEBM video	video/webm
.webp	WEBP image	image/webp
.woff	Web Open Font Format (WOFF)	font/woff
.woff2	Web Open Font Format (WOFF)	font/woff2
.xhtml	XHTML	application/xhtml+xml
.xls	Microsoft Excel	application/vnd.ms-excel
.xml	XML	application/xml
.xul	XUL	application/vnd.mozilla.xul+xml
.zip	ZIP archive	application/zip
.3gp	3GPP audio/video container	video/3gpp
		audio/3gpp if it doesn't contain video
.3g2	3GPP2 audio/video container	video/3gpp2
		audio/3gpp2 if it doesn't contain video
.7z	7-zip archive	application/x-7z-compressed