Class: WebRequest

> *Intercept and modify the contents of a request at various stages of its lifetime.*

Process: Main
This class is not exported from the `'electron'` module. It is only available as a return value of other methods in the Electron API.

Instances of the `WebRequest` class are accessed by using the `webRequest` property of a `Session`.

The methods of `WebRequest` accept an optional `filter` and a `listener`. The `listener` will be called with `listener(details)` when the API's event has happened. The `details` object describes the request.

⚠ Only the last attached `listener` will be used. Passing `null` as `listener` will unsubscribe from the event.

The `filter` object has a `urls` property which is an Array of URL patterns that will be used to filter out the requests that do not match the URL patterns. If the `filter` is omitted then all requests will be matched.

For certain events the `listener` is passed with a `callback`, which should be called with a `response` object when `listener` has done its work.

An example of adding `User-Agent` header for requests:

```
const { session } = require('electron')

// Modify the user agent for all requests to the following urls.
const filter = {
  urls: ['https://*.github.com/*', '*://electron.github.io']
}

session.defaultSession.webRequest.onBeforeSendHeaders(filter, (details, callback) =>
{
  details.requestHeaders['User-Agent'] = 'MyAgent'
  callback({ requestHeaders: details.requestHeaders })
})
```

Instance Methods

The following methods are available on instances of `WebRequest`:

`webRequest.onBeforeRequest([filter, ]listener)`

- `filter` WebRequestFilter (optional)
- `listener` Function | null
    - `details` Object
        - `id` Integer
        - `url` string
        - `method` string
        - `webContentsId` Integer (optional)
        - `webContents` WebContents (optional)
        - `frame` WebFrameMain (optional)

        - `resourceType` string - Can be `mainFrame` , `subFrame` , `stylesheet` , `script` , `image` , `font` , `object` , `xhr` , `ping` , `cspReport` , `media` , `webSocket` or `other` .
        - `referrer` string
        - `timestamp` Double
        - `uploadData` [UploadData[]]()
    - `callback` Function
        - `response` Object
            - `cancel` boolean (optional)
            - `redirectURL` string (optional) - The original request is prevented from being sent or completed and is instead redirected to the given URL.

The `listener` will be called with `listener(details, callback)` when a request is about to occur.

The `uploadData` is an array of `UploadData` objects.

The `callback` has to be called with an `response` object.

Some examples of valid `urls` :

```
'http://foo:1234/'
'http://foo.com/'
'http://foo:1234/bar'
'*://*/*'
'*://example.com/*'
'*://example.com/foo/*'
'http://*.foo:1234/'
'file://foo:1234/bar'
'http://foo:*/'
'*://www.foo.com/'
```

`webRequest.onBeforeSendHeaders([filter, ]listener)`

- `filter` [WebRequestFilter]() (optional)
- `listener` Function | null
    - `details` Object
        - `id` Integer
        - `url` string
        - `method` string
        - `webContentsId` Integer (optional)
        - `webContents` WebContents (optional)
        - `frame` WebFrameMain (optional)
        - `resourceType` string - Can be `mainFrame` , `subFrame` , `stylesheet` , `script` , `image` , `font` , `object` , `xhr` , `ping` , `cspReport` , `media` , `webSocket` or `other` .
        - `referrer` string
        - `timestamp` Double
        - `uploadData` [UploadData[]]() (optional)

    - `requestHeaders` Record<string, string>
  - `callback` Function
    - `beforeSendResponse` Object
      - `cancel` boolean (optional)
      - `requestHeaders` Record<string, string | string[]> (optional) - When provided, request will be made with these headers.

The `listener` will be called with `listener(details, callback)` before sending an HTTP request, once the request headers are available. This may occur after a TCP connection is made to the server, but before any http data is sent.

The `callback` has to be called with a `response` object.

`webRequest.onSendHeaders([filter, ]listener)`

- `filter` WebRequestFilter (optional)
- `listener` Function | null
  - `details` Object
    - `id` Integer
    - `url` string
    - `method` string
    - `webContentsId` Integer (optional)
    - `webContents` WebContents (optional)
    - `frame` WebFrameMain (optional)
    - `resourceType` string - Can be `mainFrame`, `subFrame`, `stylesheet`, `script`, `image`, `font`, `object`, `xhr`, `ping`, `cspReport`, `media`, `webSocket` or `other`.
    - `referrer` string
    - `timestamp` Double
    - `requestHeaders` Record<string, string>

The `listener` will be called with `listener(details)` just before a request is going to be sent to the server, modifications of previous `onBeforeSendHeaders` response are visible by the time this listener is fired.

`webRequest.onHeadersReceived([filter, ]listener)`

- `filter` WebRequestFilter (optional)
- `listener` Function | null
  - `details` Object
    - `id` Integer
    - `url` string
    - `method` string
    - `webContentsId` Integer (optional)
    - `webContents` WebContents (optional)
    - `frame` WebFrameMain (optional)
    - `resourceType` string - Can be `mainFrame`, `subFrame`, `stylesheet`, `script`, `image`, `font`, `object`, `xhr`, `ping`, `cspReport`, `media`, `webSocket` or `other`.

    - `referrer` string
    - `timestamp` Double
    - `statusLine` string
    - `statusCode` Integer
    - `responseHeaders` Record<string, string[]> (optional)
  - `callback` Function
    - `headersReceivedResponse` Object
      - `cancel` boolean (optional)
      - `responseHeaders` Record<string, string | string[]> (optional) - When provided, the server is assumed to have responded with these headers.
      - `statusLine` string (optional) - Should be provided when overriding `responseHeaders` to change header status otherwise original response header's status will be used.

The `listener` will be called with `listener(details, callback)` when HTTP response headers of a request have been received.

The `callback` has to be called with a `response` object.

`webRequest.onResponseStarted([filter, ]listener)`

- `filter` [WebRequestFilter](#) (optional)
- `listener` Function | null
  - `details` Object
    - `id` Integer
    - `url` string
    - `method` string
    - `webContentsId` Integer (optional)
    - `webContents` WebContents (optional)
    - `frame` WebFrameMain (optional)
    - `resourceType` string - Can be `mainFrame`, `subFrame`, `stylesheet`, `script`, `image`, `font`, `object`, `xhr`, `ping`, `cspReport`, `media`, `webSocket` or `other`.
    - `referrer` string
    - `timestamp` Double
    - `responseHeaders` Record<string, string[]> (optional)
    - `fromCache` boolean - Indicates whether the response was fetched from disk cache.
    - `statusCode` Integer
    - `statusLine` string

The `listener` will be called with `listener(details)` when first byte of the response body is received. For HTTP requests, this means that the status line and response headers are available.

`webRequest.onBeforeRedirect([filter, ]listener)`

- `filter` [WebRequestFilter](#) (optional)
- `listener` Function | null
  - `details` Object

        - `id` Integer
        - `url` string
        - `method` string
        - `webContentsId` Integer (optional)
        - `webContents` WebContents (optional)
        - `frame` WebFrameMain (optional)
        - `resourceType` string - Can be `mainFrame` , `subFrame` , `stylesheet` , `script` , `image` , `font` , `object` , `xhr` , `ping` , `cspReport` , `media` , `webSocket` or `other` .
        - `referrer` string
        - `timestamp` Double
        - `redirectURL` string
        - `statusCode` Integer
        - `statusLine` string
        - `ip` string (optional) - The server IP address that the request was actually sent to.
        - `fromCache` boolean
        - `responseHeaders` Record<string, string[]> (optional)

The `listener` will be called with `listener(details)` when a server initiated redirect is about to occur.

`webRequest.onCompleted([filter, ]listener)`

- `filter` WebRequestFilter (optional)
- `listener` Function | null
    - `details` Object
        - `id` Integer
        - `url` string
        - `method` string
        - `webContentsId` Integer (optional)
        - `webContents` WebContents (optional)
        - `frame` WebFrameMain (optional)
        - `resourceType` string - Can be `mainFrame` , `subFrame` , `stylesheet` , `script` , `image` , `font` , `object` , `xhr` , `ping` , `cspReport` , `media` , `webSocket` or `other` .
        - `referrer` string
        - `timestamp` Double
        - `responseHeaders` Record<string, string[]> (optional)
        - `fromCache` boolean
        - `statusCode` Integer
        - `statusLine` string
        - `error` string

The `listener` will be called with `listener(details)` when a request is completed.

`webRequest.onErrorOccurred([filter, ]listener)`

- `filter` WebRequestFilter (optional)
- `listener` Function | null

- `details` Object
  - `id` Integer
  - `url` string
  - `method` string
  - `webContentsId` Integer (optional)
  - `webContents` WebContents (optional)
  - `frame` WebFrameMain (optional)
  - `resourceType` string - Can be `mainFrame`, `subFrame`, `stylesheet`, `script`, `image`, `font`, `object`, `xhr`, `ping`, `cspReport`, `media`, `webSocket` or `other`.
  - `referrer` string
  - `timestamp` Double
  - `fromCache` boolean
  - `error` string - The error description.

The `listener` will be called with `listener(details)` when an error occurs.