

window.XMLHttpRequest Objekt des Internet Explorer (ab IE 8)

Objekt für domainübergreifender Zugriff über HTTP per XML

Sicherer Datenservice für jede Seite bei anonymen Zugriff auf jeden Server zum Zweck des XML-Datenaustausches zwischen Domains.

Hinweis: XMLHttpRequest hat eine Sicherheitskomponente implementiert: Domainübergreifend nicht möglich.

Attribute:

contentType	Contenttyp-Header setzen bzw. liefern
responseText	Body der Antwort des Servers
timeout	Timeoutwert setzen bzw. liefern

Methoden:

abort()	beendet das aktive Senden / Empfangen
open()	erzeugt eine Verbindung zum Server
send()	Datenstring an den Server senden, der die Daten verarbeiten soll

Event:

onerror	tritt ein, wenn Zugriff auf Objekt-Daten nicht komplett werden kann
onload	tritt ein wenn Objekt-Daten vom Server komplett eingetroffen sind
onprogress	tritt ein, wenn der Browser beginnt, Objekt-Daten vom Server zu empfangen
ontimeout	tritt ein, wenn Zugriff auf Objekt-Daten wegen Überschreitung der Zeit laut Attribut timeout nicht komplett werden kann

Beispiel:

```
if (window.XMLHttpRequest)
{
    var oReq = new XMLHttpRequest();
    if (oReq) {
        oReq.open("GET", "http://example.com/test.xml");
        oReq.send();
        alert(oReq.statusText);
    }
}
```

Attribute

Attribut contentType

content-type header setzen oder liefern

Syntax:

```
[ v = ] XMLHttpRequest.contentType [ = v ]
```

v text des content-type header

lesen und schreiben

kein Standardwert

Beispiel:

```
// 1. Create XDR object
xdr = new XMLHttpRequest();
// 2. Open connection with server using POST method
xdr.open("POST", "http://www.contoso.com/xdr.txt");
// 3. Set the content-type
xdr.contentType = "text/plain";
// 4. Send string data to server
xdr.send("data to be processed");
```

Attribut responseText

Body der Antwort vom Server liefern

Zugriff nur möglich, wenn onprogress oder onload eingetreten ist (sonst erzeugt der Zugriff einen Ausnahmefehler erzeugt)

onload tritt nach onprogress ein

Body nur komplett verfügbar, wenn onload eingetreten ist (bei onprogress nicht vollständig verfügbar)

Syntax:

```
[ p = ] XMLHttpRequest.responseText
```

p String

Attribut timeout

Timeoutwert setzen oder liefern

maximale Wartezeit auf Antwort des Servers



Syntax:

[v =] XDomainRequest.timeout [= v]

v Millisekunden

lesen und schreiben
kein Standardwert

Methoden

Methode abort()

beendet ein Senden / Empfangen
nur aufrufbar nach senden und bevor onload-Event eintritt
ansonsten immer Error erzeugt

Syntax:

XDomainRequest.abort();

liefert nichts

Beispiel:

```
// 1. Create XDR object
xdr = new XDomainRequest();
// 2. Open connection with server using POST method
xdr.open("POST", "http://www.contoso.com/xdr.txt");
// 3. Send string data to server
xdr.send("data to be processed");
// 4. abort
xdr.abort();
```

Methode open()

Verbindung zum Server erzeugen

Syntax:

XDomainRequest.open(methode, url)

methode String
'GET'
'POST'

url String
Url vom Server

liefert nichts

Beispiel:

```
// 1. Create XDR object
xdr = new XDomainRequest();
// 2. Open connection with server using POST method
xdr.open("POST", "http://www.contoso.com/xdr.txt");
// 3. Send string data to server
xdr.send("data to be processed");
```

Methode send()

sendet einen Datenstring zum Server, der den String verarbeiten soll

Syntax:

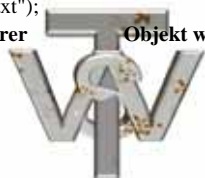
XDomainRequest.send([varBody])

varBody optional
String
Standard ist ""

liefert nichts

Beispiel:

```
// 1. Create XDR object
xdr = new XDomainRequest();
// 2. Open connection with server using POST method
xdr.open("POST", "http://www.contoso.com/xdr.txt");
```



```
// 3. Send string data to server
xhr.send("data to be processed");
```

Events

Event onerror

tritt ein, wenn ein Fehler das Zugriff verhindert
welcher Fehler es ist, ist nicht ermittelbar
onerror wird nicht ausgelöst durch ontimeout

Syntax:

```
<SCRIPT FOR = XMLHttpRequest EVENT = onerror>
XMLHttpRequest.onerror=handler;
```

Event Information:

Bubbles No
Cancels No

Beispiel:

```
<script type="text/javascript">
function err()
{
    alert("XMLHttpRequest onerror");
}
...
xhr.onerror = err;
```

Event onload

tritt ein, wenn Objektdaten komplett vom Server empfangen wurden (nach einem send)
wenn eingetreten, dann ist responseText komplett gefüllt und zugreifbar

Syntax:

```
<SCRIPT FOR = XMLHttpRequest EVENT = onload>
XMLHttpRequest.onload=handler;
```

Event Information:

Bubbles No
Cancels No

Beispiel:

```
function load()
{
    alert("XMLHttpRequest onload");
    alert("Got: " + xhr.responseText);
}
...
xhr.onload = load;
```

Event onprogress

tritt ein, solange Browser Daten vom Server empfängt (nach send und vor Eintritt vom Event onload)
wenn eingetreten, dann ist responseText nicht komplett gefüllt

Syntax:

```
<SCRIPT FOR = XMLHttpRequest EVENT = onprogress>
XMLHttpRequest.onprogress=handler;
```

Event Information:

Bubbles No
Cancels No

Beispiel:

```
function progres()
{
    alert("XMLHttpRequest onprogress");
    alert("Got: " + xhr.responseText);
}
...
xhr.onprogress = progres;
```



Event ontimeout

tritt ein, wenn Zugriff innerhalb Zeitspanne laut Attribut timeout nicht komplett erfolgt ist (nach send) responseText ist nicht verfügbar, so dass ein Zugriff auf dieses Attribut einen Fehler erzeugt

Hinweis: Zugriff ist komplett erfolgt, wenn Event onload ausgelöst wurde

Syntax:

```
<SCRIPT FOR = object EVENT = ontimeout>  
XDomainRequest.ontimeout=handler;
```

Event Information:

Bubbles No
Cancels No

Beispiel:

```
function timeo()  
{  
    alert("XDR ontimeout");  
}  
...  
xdr.ontimeout = timeo;
```

