



Web- og serverprogrammering



dag 9

AHAH - Asynkron HTML og HTTP

AJAX - Asynkron Javascript og XML

JSON - Javascript Object Notation



AJAX - Asynkron Javascript og XML



- En gruppe beslægtede webudviklingsteknikker til interaktive webapplikationer (GUI applikationer) på nettet
- gmail.com er en meget kendt AJAX-applikation
- AJAX-principper
 - Ingen eller meget få sideskift
 - Javascript på siden sender data til/fra serveren.
 - Til (a)synkron datatransport bruges objektet XMLHttpRequest
- Browserafhængighed
 - Microsoft IE har sin egen måde at danne XMLHttpRequest-objektet
- Masser af biblioteker der hjælper med AJAX
 - <http://www.prototypejs.org/>
- Masser af biblioteker der udnytter AJAX
 - Til mange opgaver er det ikke nødvendigt at skrive AJAX-kode selv



AJAX - Asynkron Javascript og XML



```
var ajax = danXMLHttpRequest();

//ajax.open(metode, url (samme server), asynkron);
ajax.open("GET", 'henttidXML.jsp?x=y', true);

ajax.onreadystatechange = callbackFunktionDerSkalKaldes;

//ajax.send(formulardata hvis metode er "POST");
ajax.send(null);

...

function callbackFunktionDerSkalKaldes() {

    if (ajax.readyState == 4 && ajax.status == 200) {
        alert(ajax.statusText);    // giver 'OK'
        alert(ajax.responseXML); // data fra server som XML
        alert(ajax.responseText); // data fra server som tekst
    }
}
```



```

<script language="JavaScript" type="text/javascript">
var http = danXMLHttpRequest();

function hentServerTid() {
    var minurl = 'henttidXML.jsp';
    minRand = parseInt(Math.random()*9999999999999999);
    var modurl = minurl+"?rand="+minRand;
    http.open("GET", modurl, true);
    http.onreadystatechange = brugHttpResponse;
    http.send(null);
}

function brugHttpResponse() {
    if (http.readyState == 4) {
        if (http.status == 200) {
            var tidValue = http.responseXML.getElementsByTagName("tidnu")[0];
            document.getElementById('vistid').innerHTML = tidValue.childNodes[0].nodeValue;
        }
        } else {
        document.getElementById('vistid').innerHTML = '';
        }
    }
}
</script>
</head>
<body style="background-color:#cccccc" onLoad="hentServerTid()">
<center>
<h1>Ajax-demonstration</h1>
<h2>Hent servertiden uden at opdatere siden</h2>
<form>
<input type="button" value="Hent servertid" onClick="hentServerTid()">

</form>
<div id="vistid" class="displaybox"></div>
</center>
</body>
</html>
<?xml version="1.0" ?>
<% Thread.sleep(1000); %>
<tid1><tidnu><%= new java.util.Date() %></tidnu></tid1>

```





AHAH - Asynkron HTML og HTTP



- XML er ikke altid lykken, den skal sættes sammen på serveren og den skal parses af Javascript
 - `ajax.responseXml`
- Simplere at returnere HTML-fragmenter eller tekst
 - `ajax.responseText`
- Denne tekst kan sættes direkte ind i HTML-siden



AHAH - Asynkron HTML og HTTP



```
<html><head>
<title>Keywords-metatag-opsnapper</title>
</head><body>
<script type="text/javascript" src="mitAHAHlib.js"></script>
<form>
<table>
<tr><td>URL: http://</td>
<td><input type="text" id="minurl" name="minurl" size=30>
  <input type="button" onclick="kaldAHAH('keywords.jsp?
url='+document.getElementById('minurl').value,'visdiv', 'Vent venligst; indlæser indhold ...')"
value="Hent" >
</td></tr>
<tr><td colspan=2 height=50 id="visdiv"></td></tr>
</table>
</form></body></html>
```

```
<%
java.net.URL url = new java.net.URL("http://" + request.getParameter("url"));
java.io.BufferedReader ind = new java.io.BufferedReader(new java.io.InputStreamReader(url.openStream()));
String s = ind.readLine();
while (s != null)
{
  if (s.indexOf("keywords") != -1) {
    for (String e : s.split("[,\\\"\\'<=>]")) out.println(e + "<br/>");
  }
  s = ind.readLine();
}
%>
```



AHAH - Asynkron HTML og HTTP



```
function kaldAHAH(url, sideElement, kaldMessage) {
    document.getElementById(sideElement).innerHTML = kaldMessage;
    try {
        req = new XMLHttpRequest(); /* f.eks. Firefox */
    } catch(e) {
        try {
            req = new ActiveXObject("Msxml2.XMLHTTP"); /* IE-versioner */
        } catch (e) {
            try {
                req = new ActiveXObject("Microsoft.XMLHTTP"); /* IE-versioner */
            } catch (e) {
                req = false;
            }
        }
    }
    req.onreadystatechange = function() {svarAHAH(sideElement);};
    req.open("GET",url,true);
    req.send(null);
}

function svarAHAH(sideElement) {
    var output = '';
    if(req.readyState == 4) {
        if(req.status == 200) {
            output = req.responseText;
            document.getElementById(sideElement).innerHTML = output;
        }
    }
}
```



JSON - Javascript Object Notation



- XML

ajax.responseXML

.getElementByTagName("firstname");

- JSON

var p = eval("(" + ajax.responseXML + ")");

p.firstName

p.address.city

p.phoneNumbers[0]

```
<Person firstName="John" lastName="Smith">
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumber type="home">
    212 555-1234
  </phoneNumber>
  <phoneNumber type="fax">
    646 555-4567
  </phoneNumber>
</Person>
```

```
{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021"
  },
  "phoneNumbers": [
    { "type": "home", "number": "212 555-1234" },
    { "type": "fax", "number": "646 555-4567" }
  ]
}
```

Kilde: <http://en.wikipedia.org/wiki/JSON>



JSON i en AJAX/AHAAH-applikation



```
var the_object;
var http_request = new XMLHttpRequest();
http_request.open( "GET", url, true );
http_request.onreadystatechange = function () {
    if ( http_request.readyState == 4 ) {
        if ( http_request.status == 200 ) {
            the_object = eval( "(" + http_request.responseText + ")" );
        } else {
            alert( "There was a problem with the URL." );
        }
        http_request = null;
    }
};
http_request.send(null);
```



Åben Dokumentlicens



- Dette foredragsmateriale er under Åben Dokumentlicens (ÅDL)
 - Du har derfor lov til frit at kopiere dette værk
 - Bruger du dele af værket i et nyt værk, skal de dele, der stammer fra dette værk, igen frigives under ÅDL
- Den fulde licens kan ses på <http://www.sslug.dk/linuxbog/licens.html>

