

Praktikum DBAE / WI

Asynchronous JavaScript & XML

Pascal Reuss, M.Sc.

Raum A08b Spl.
Email: reusspa@uni-hildesheim.de

AJAX

- Vorteile
 - Reduzierung des Traffics
 - Schnellere Reaktion auf Benutzereingaben
 - Teilweise Verbesserung der Usability
 - Ressourcenschonender (Zeit und Bandbreite)
 - Kein Browser-Plugin notwendig
- Nachteile
 - Javascript muss aktiviert sein
 - Oft nicht barrierefrei
 - Suchmaschinenoptimierung nicht unterstützt
 - Teilweise Verschlechterung der Usability (Browsernavigation, Lesezeichen)

AJAX

Klassische Anwendungen laden bei jeder Änderung die komplette Seite neu

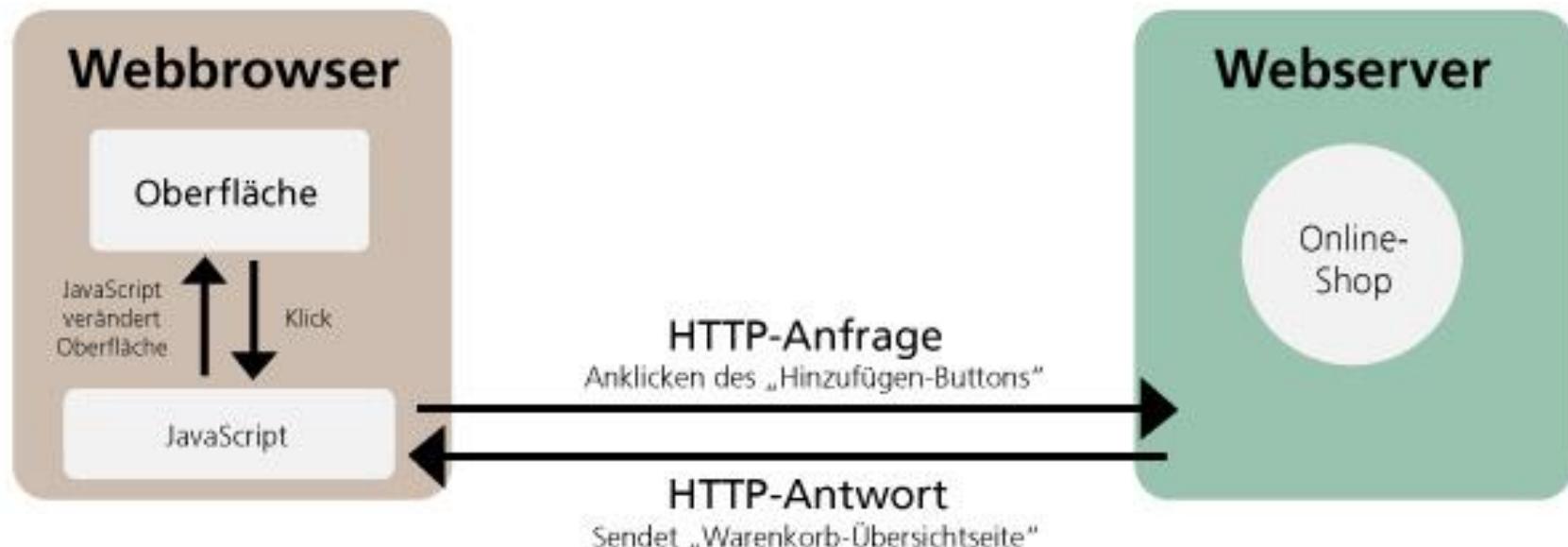
Die Daten werden über die HTTP-Verbindung an den Server geschickt und dieser sendet die aktualisierten Daten zurück (siehe Grafik)



AJAX

Mit AJAX wird eine Verbindung unter Verwendung von Javascript aufgebaut.

Die Antwort des Servers wird ebenfalls mit Hilfe von Javascript ausgewertet. Dadurch kann nur ein bestimmter Teil aktualisiert werden.



AJAX

Verwendung von AJAX macht die Handhabung und Gestaltung von Webseiten dynamischer.

Verschiedene Technologien finden dabei Verwendung:

- (X)HTML
- CSS
- Javascript
- DOM (Document Object Model)
- XMLHttpRequest
- XML

Außerdem kann AJAX mit anderen Technologien kombiniert werden:

- PHP, Python, Perl, Java, JSON, etc

AJAX

Das Versenden der Daten geschieht bei AJAX über das

XMLHttpRequest-Objekt

- Grundlage der AJAX-Technik
- Wird von Javascript verwendet
- Schnittstelle zur Kommunikation mit dem Server
- Probleme mit älteren Browsern

AJAX

Methoden des XMLHttpRequest-Objekts:

- `abort()`
bricht den laufenden Request ab
- `getResponseHeader(header)`
gibt einen bestimmten Header der Antwort zurück
- `getAllResponseHeaders()`
gibt alle Header der Antwort zurück
- `Open(method, url)`
*bereit eine http-Anfrage vor
method kann POST, GET oder PUT sein
url ist die Adresse, an welche die Anfrage geht*
- `Send(body)`
*sendet die http-Anfrage
body kann POST-Daten enthalten*
- `setRequestHeader(header, value)`
setzt einen Header für die Http-Anfrage auf einen Wert

AJAX

Attribute des XMLHttpRequest-Objekts:

- **readyState**
aktueller Status
 - 0 *Nicht initialisiert*
 - 1 *Objekt ist bereit, keine Daten gesendet*
 - 2 *Anfrage wurde gesendet*
 - 3 *Daten werden empfangen*
 - 4 *Alle Daten wurden empfangen*
- **onreadystatechange**
Referenz auf die Funktion, die aufgerufen wird, wenn sich die Anfrage ändert
- **responseText**
Http-Antwort als Text
- **responseXML**
Http-Antwort als XML, Zugriff per DOM möglich
- **status**
Http-Statuscode der Antwort (200, 404, etc)
- **statusText**
Http-Statustext der Antwort

AJAX

```
<html>
<head>
    <title>Nur ein Test</title>
    <script type="text/javascript">
        // Objekt erzeugen
        // hier ohne Fallunterscheidung, da für das Beispiel einfacher
        var XMLHttpRequest = new XMLHttpRequest();

        // Funktion, die bei Statusänderungen reagiert
        function handleStateChange()
        {
            // Derzeitigen Status zurückgeben
            alert(`ReadyState= ${XMLHttpRequest.readyState} , HTTP-Status = ${XMLHttpRequest.status}`);
        }

        // Wenn Dokument geladen ausführen
        window.onload = function() {
            // Anfrage vorbereiten
            XMLHttpRequest.open('GET', '/');
            // Handler hinterlegen
            XMLHttpRequest.onreadystatechange = handleStateChange;
            // Anfrage abschicken
            XMLHttpRequest.send(null);
        }
    </script>
</head>
<body>
    <p>Es sollte sich etwas tun</p>
</body>
</html>
```

AJAX Formular

```
<html>
  <head>
    <script src=„ajax.js“></script>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Formular mit Ajax</title>
  </head>
  <body>
    <div id="resultdiv"></div>
    <form action='.' method='post'>
      Name:<input type='text' id='user' onChange=sendForm();><br>
      Adresse:<input type='text' id='user'><br>
      Email:<input type='text' id='user'><br>
      <input type='button' value='Senden'>
    </form>
  </body>
</html>
```

Name:

Adresse:

Email:

Senden

AJAX Servlet

```
public void doPost(HttpServletRequest request, HttpServletResponse response) {  
  
    String user = request.getParameter("user");  
  
    try {  
        response.addHeader("Access-Control-Allow-Origin", "");  
        response.setCharacterEncoding("UTF-8");  
  
        OutputStream stream = response.getOutputStream();  
        OutputStreamWriter writer = new OutputStreamWriter(stream, "UTF-8");  
  
        writer.write(user);  
        writer.flush();  
  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

AJAX Javascript 1

```
function sendForm() {  
  
    httprequest = getRequestObject();  
  
    if (httprequest == null) {  
        alert("Kein Ajax unterstützt");  
        return;  
    }  
  
    var url = "process";  
  
    var user = document.getElementById("user").value;  
  
    httprequest.open("POST", url, true);  
    httprequest.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
    httprequest.setRequestHeader("Content-length", user.length);  
  
    httprequest.onreadystatechange=getOutput;  
  
    httprequest.send("user="+user);  
}
```

AJAX Javascript 2

```
function getRequestObject() {
    //neue Browser
    if (window.XMLHttpRequest) {
        return new XMLHttpRequest();
    }

    //alte Browser (IE5, etc.)
} else {
    return new ActiveXObject("Microsoft.XMLHTTP");
}

}

function getOutput() {
    if (httprequest.readyState == 4 && httprequest.status == 200) {
        var result = httprequest.responseText;

        document.getElementById("resultdiv").innerHTML = "Hello " + result;
    }
}
```

AJAX

Sicherheit

- AJAX ist Clientseitig ausgeführter Code
- Anfällig für Manipulation
- Aber die Sicherheit hängt immer von der Implementierung ab!
- Sicherheit auf der Serverseite gewährleisten
- Mögliche Sicherheitsprobleme
 - Autocomplete mittels AJAX
 - Logging von Usereingaben
 - Javascript Injection

AJAX

Frameworks

- Dojo dojotoolkit.org
- jQuery jquery.com
- MooTools mootools.net
- Prototype prototypejs.org
- Apache Wicket wicket.apache.org
- Sajax www.modernmethod.com/sajax/index.phtml
- Rich Ajax Plattform Eclipse Plugin