



Seminar Webtechnologie

AJAX

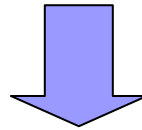
Michael Jantz

Übersicht

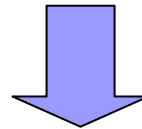
- Wozu dient AJAX?
- Was ist AJAX?
- Beispielanwendung
- Frameworks
- Diskussion
- Ausblick
- (zusätzliche Beispiele)

Wozu dient AJAX?

Man will: Desktopähnliche, besser benutzbare
Webanwendungen



Wartezeiten verringern oder verstecken



Asynchrone, dynamische Datenübertragung

Was ist das Web 2.0?

- Oberbegriff für den Einsatz neuester Technologien (AJAX, Atom, Trackback, RSS...)
- Name eines wenig beachteten Buches von Tim O'Reilly (erschienen 2002)
- Name einer jährlichen Konferenz
- Ziel ist ein vernetztes, einfaches, kollaboratives und offenes Internet

Was ist das Web 2.0?

Vergleich Web 1.0

- Britannica online vs Wikipedia
- personal websites → blogging
- „Akamai → BitTorrent
- Screen scraping vs Webservices
- Publishing vs Participation
- screen scraping → web services
- Directories („taxonomy“) vs Tagging („folksonomy“)

Was ist das Web 2.0? Anforderungen

- Netzbasiert im Browser
- Dynamische Seiten, personalisierte Oberfläche
- Rollen- und Rechtesystem
- Benutzer besitzen ihre "eigenen" Daten, kollaborative Bearbeitung möglich
- Benutzer bilden eine Community, sie haben ein Zusammengehörigkeitsgefühl

Was ist das Web 2.0?

„Synonyme“

- Oftmals synonym gebrauchte Begriffe:
 - Social Web (Social Bookmarking, Social Software)
 - Tagging zur Vernetzung von Inhalt und Mensch
 - Semantic Web
 - Erweiterung des WWW um maschinenlesbare Daten um die Semantik der Inhalte formal festzulegen (Vorschlag von Tim Berners Lee)

Was ist das Web 2.0?

Gefahren

- Datenschutz?
 - AGB von Google, Flickr, Youtube je gelesen?
- Social Phishing
 - Immer mehr Daten zu jedem Einzelnen verfügbar
- Bsp. Gmail

Was ist AJAX?

- AJAX ist Apronym für:

A synchronous

J avaScript

A nd

X ML

Was ist AJAX?

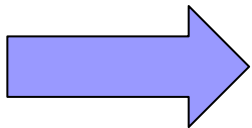
Begriff

- Begriffsherkunft unbekannt.
- **Jesse James Garrett** (adaptive path) in seinem Artikel: „Ajax: A New Approach to Web Applications“ (18.02.05)
(<http://www.adaptivepath.com/publications/essays/archives/000385.php> Stand 07.2006)
- Vorher schon verwendet z.B. GoogleMaps (08.02.05)

Was ist AJAX?

Asynchron?

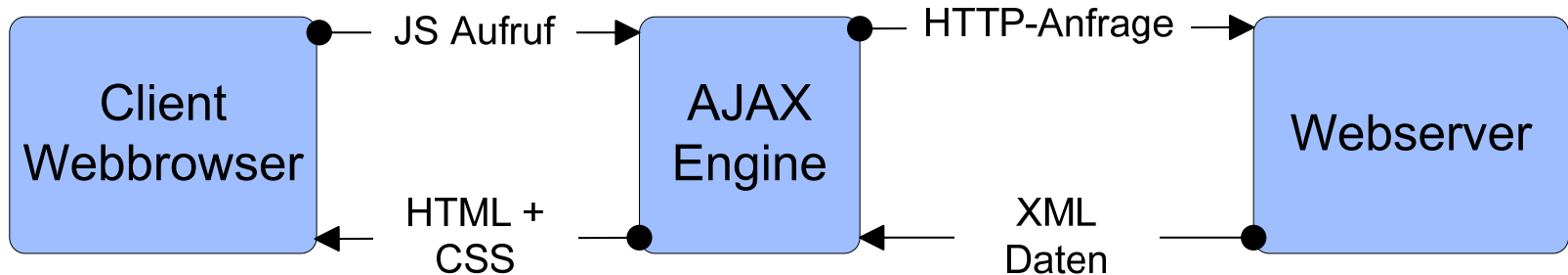
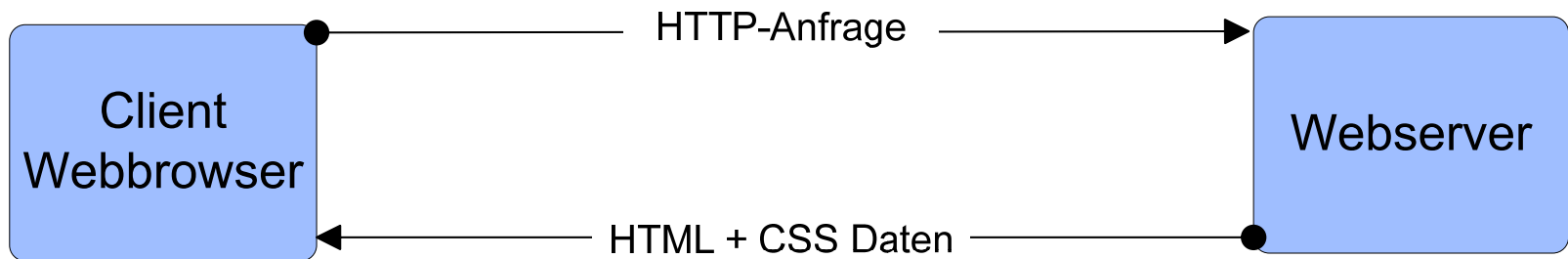
- Haupteigenschaft einer AJAX-Anwendung
- Bereitgestellt durch XMLHttpRequest
- Ununterbrochene Benutzerinteraktion möglich

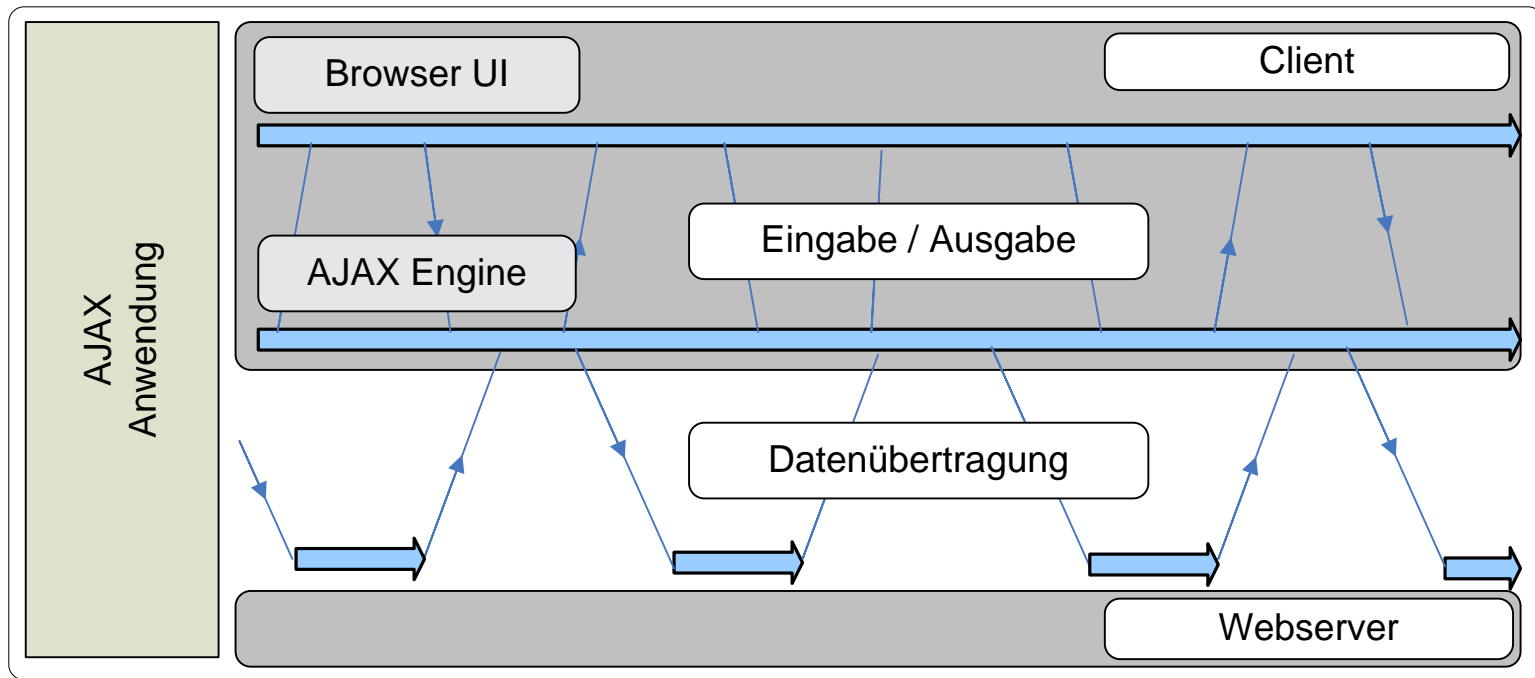
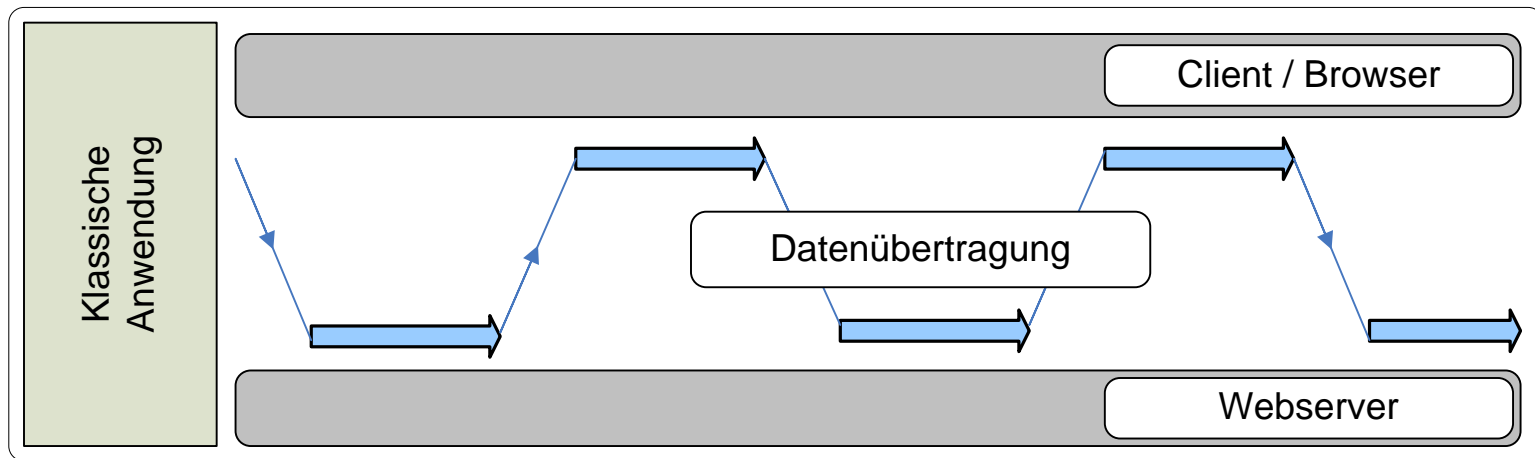


„nicht mehr zustandslos“

Was ist AJAX?

Asynchron





Was ist AJAX?

JavaScript

- clientseitigen Implementierung
- Kein fester Standard
- Bemühungen des W3C zur Standardisierung
- Alternativen: VBScript, ASP, JScript.NET, ...

Was ist AJAX?

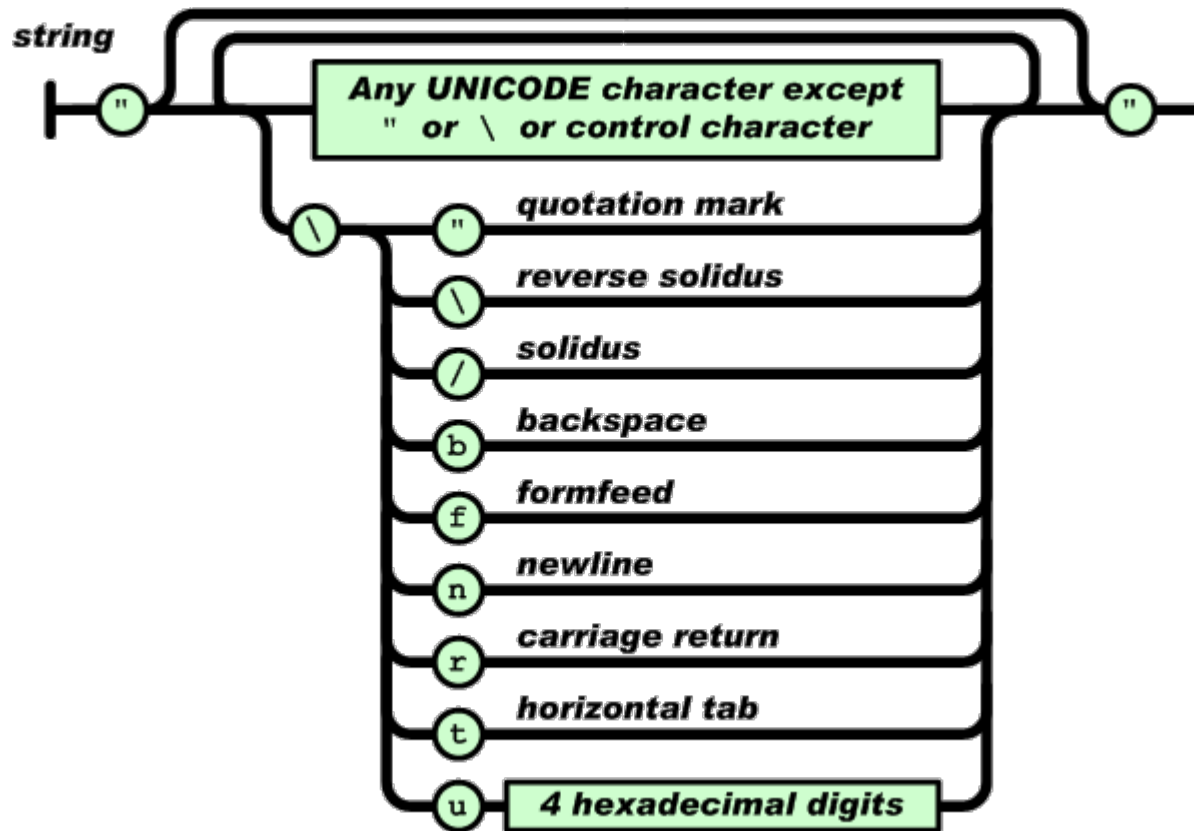
XML

- Kapselung der Daten zwischen Client und Server
- Alternativen: einfacher Text, JSON, ...

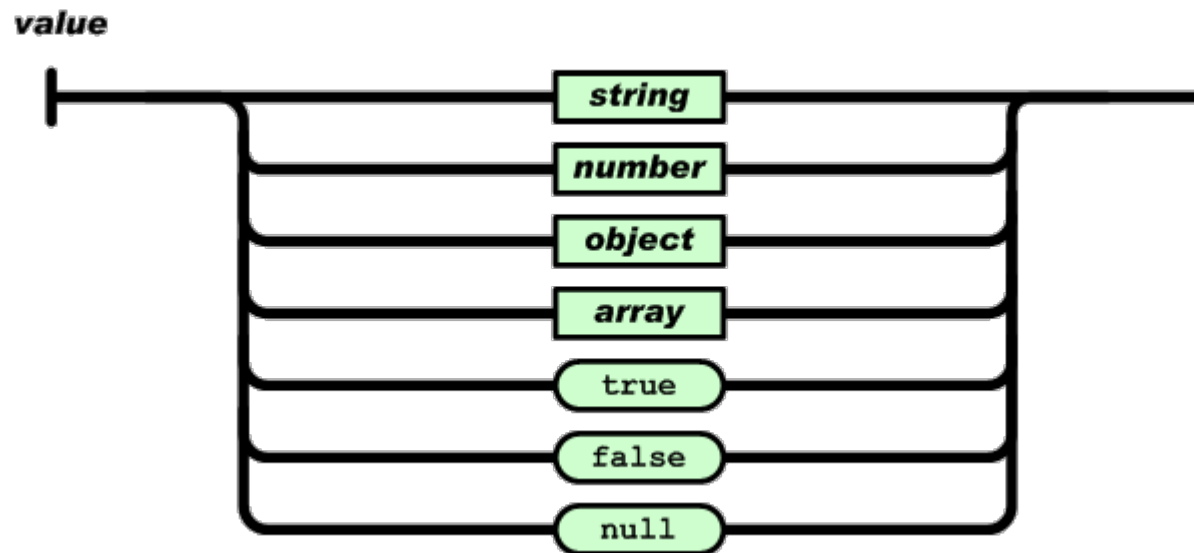
JSON

- Textformat für Daten
- Programmiersprachenunabhängig
- Konventionen verwandt zur C-Familie
- Basiert auf 2 Konstrukten
 - Ansammlung von Name-Wert-Paaren
 - Geordnete Liste von Werten

JSON – Datenstrukturen (2)



JSON – Datenstrukturen (3)



Was ist AJAX?

Bestandteile

- Weitere Techniken:
 - Präsentation durch XHTML und CSS
 - DOM-Manipulationen
 - XSLT
 - ...

Was ist AJAX?

Neues?

- Beschreibt keine neue Technik, sondern Zusammenspiel von Techniken
- Zahllose Alternativen erlaubt, deshalb mehr Beschreibung der Anwendungsart
- Es funktioniert

XMLHttpRequest-Objekt

- Bietet Kommunikationsmöglichkeit mit Server
- Sowohl synchrone als auch asynchrone Kommunikation
- Kernelement der „klassischen“ AJAX-Anwendung

XMLHttpRequest-Objekt

- 1998 Remote-Skripting-Komponente
- 1999 Einführung von XMLHttpRequest als ActiveX-Objekt im IE 5
- Native Unterstützung durch Mozilla 1.0 (2002), Safari 1.2 (2004), Opera 8 (2005), IE 7 (2006?)
- 02.2006 Gründung der „Open Ajax Initiative“
- 04.2006 Erster Standardisierungsvorschlag des W3C

XMLHttpRequest-Objekt

- Genaue API bei einigen Browserherstellern einsehbar
 - IE
 - Mozilla
 - Safari
- Ausserdem Standardvorschlag bei W3C
 - <http://www.w3.org/TR/XMLHttpRequest/>

Was bietet XMLHttpRequest?

■ Prozeduren

- abort()
- open(Methode, URL, asynchron)
 - Methode = GET, POST, PUT, HEAD, ...
 - URL = exakter Domainname, nur innerhalb der eigenen Domain
 - Asynchron = boolean, ob asynchrone Anfrage oder nicht
- overrideMimeType(„text/xml“)
- setRequestHeader("label", "value")
- setRequestHeader('Content-Type', 'application/x-www-form-urlencoded')
- send(content)
 - content = zu sendende Daten urlencoded oder aber „null“
- getAllResponseHeaders ()
- getResponseHeader(„headername“)

Was bietet XMLHttpRequest?(2)

■ Felder

- onreadystatechange
- readyState
- responseText
- responseXML
- status
- statusText

Beispielanwendung

- GoogleSuggest ähnliche Umsetzung eines Wörterbuchs

```
<body>
  <div align="center">
    <h1 align="center">Dictionary</h1>
    <div align="center">Type in part of a word to have it defined.</div>
    <form action="find.php" method="GET" target="searchWindow">
      <input autocomplete="off" type="text" name="q" id="q" size="20" >
      <div align="left" class="ausgabe" id="searchResult" name="searchResult"></div>
    </form>
  </div>
</body>
```

Beispielanwendung (2)

- JS Events

```
<input autocomplete="off" type="text" name="q" id="q" size="20"  
onkeyup="loadXMLDoc('find.php?q='+this.value)">
```

Beispielanwendung (3)

```
var req = null;
function loadXMLDoc(url) {
  // Internet Explorer
  try { req = new ActiveXObject("Msxml2.XMLHTTP"); }
  catch(e) {
    try { req = new ActiveXObject("Microsoft.XMLHTTP"); }
    catch(oc) { req = null; }
  }
  // neue Browser
  if (req == null && typeof XMLHttpRequest != "undefined") {
    req = new XMLHttpRequest();
  }
  // Call the processChange() function when the page has loaded
  if (req != null) {
    req.onreadystatechange = processChange;
    req.open("GET", url, true);
    req.send(null);
  }
}
```

Beispielanwendung (4)

- Antwort auswerten und anzeigen

```
function processChange(evt) {  
    // The page has loaded and the HTTP status code is 200 OK  
    if (req.readyState == 4) {  
        if (req.status == 200) {  
  
            // Write the contents of this URL to the searchResult layer  
            document.getElementById ("searchResult").innerHTML = req.responseText;  
        }  
    }  
}
```

Beispielanwendung (5)

■ Serverseite

```
<?php
mysql_connect("localhost", "root", "");
mysql_select_db("test");
$q = addslashes($_GET["q"]);
$limit = 10;
if ($q) {
    $query = mysql_query("SELECT * FROM dict_list WHERE word LIKE '$q%' LIMIT $limit");
    $results = array();
    while ($row = mysql_fetch_assoc($query)) {
        $word = $row["word"];
        $definition = $row["definition"];
        $results[] = "<b>$word</b>: $definition";
    }
    echo implode("<br>\n", $results);
}
?>
```

Beispielanwendung 2

- Formularüberprüfung
 - Schnelleres Arbeiten
 - Visuelles Feedback
 - Rücksicht nehmen auf Usereingaben

Frameworks

- Arbeitserleichterung für komplexe Projekte
- Mittlerweile unüberschaubare Zahl
- Viele „AJAX-Frameworks“ ergänzen „nur“ visuelle Effekte
- Für fast alle Sprachen verfügbar
- Meist leider schlecht oder gar nicht dokumentiert

Frameworks

einige Beispiele

- Google Web Toolkit
 - Verbindung von JAVA und AJAX Anwendungen
- Yahoo UI Library
- Adobe Spry Framework
- Mochikit

Frameworks

weitere Beispiele

- Prototype
 - Bildet Kommunikationsgrundlage für viele andere
- Rico
 - Basiert auf Prototype, ergänzt Wrapper und Effekte
- SAJAX
 - Für viele Sprachen verfügbar (PHP, Perl, Python, ...)



Vorteile von AJAX

- Kein Plug-In nötig
- Bandbreitenverringering
- Gesteigerte Usability
- Minimierung von Wartezeit
- Desktopähnliche Programme
- Rechenintensive Aufgaben verteilbar

Nachteile von AJAX

- Noch kein fester Standard
- Testintensiv
- JS oder sogar ActiveX nötig
- Fallbacklösung nötig
- Teilweise keine Trennung von Struktur und Layout (semantisches Web)
- Browserfunktionen gestört (Ladeanzeige, Drucken, zurück)
- Nicht barrierefrei
- Vielleicht höhere Serverlast durch vermehrte Anfragen

Ausblick

- Art der Anwendungen werden an Bedeutung gewinnen
- Zukunft des Begriffs AJAX ungewiss

Größere AJAX-Projekte

- [GoogleMaps](#)
- [Inquisitor](#)
- [ajaxlaunch](#)
- [NexImage](#)
- [iRows](#)