



Grundlagen Internet-Technologien

INF3171

AJAX und Shibboleth

Version 1.0

17.07.2025



Jetzt aktualisieren! Chrome-Sicherheitslücke wird angegriffen



Google hat in der Nacht zum Mittwoch den Chrome-Webbrowser aktualisiert. Das Update schließt auch eine bereits attackierte Lücke.



(Bild: heise online / dmk)

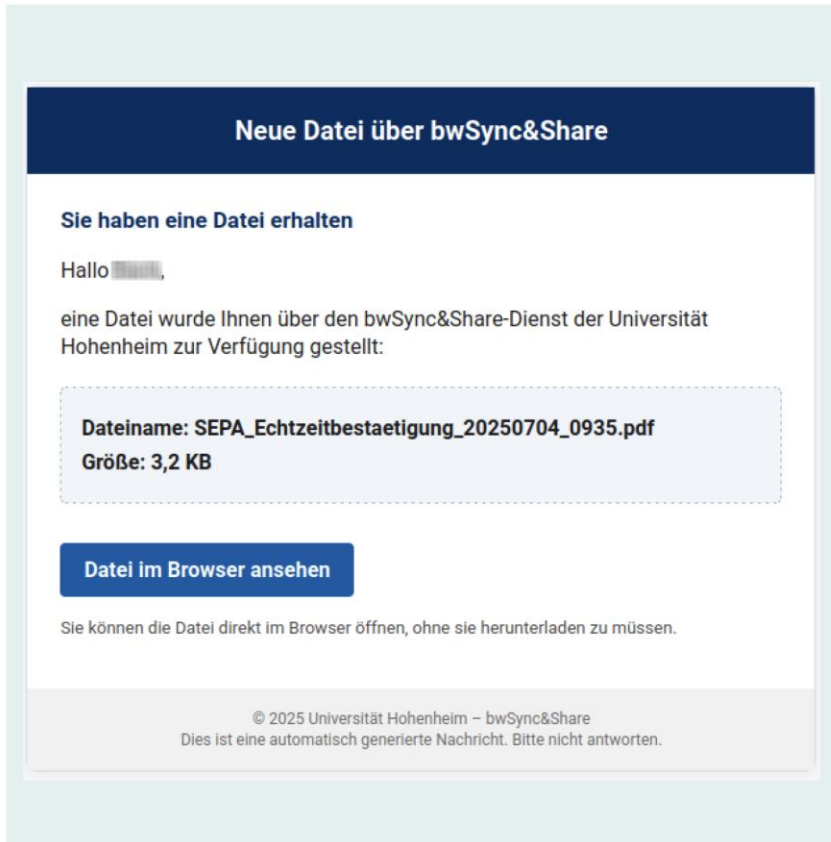
08:31 Uhr | Lesezeit: 3 Min. | Security

Von Dirk Knop

Erneut muss Google Sicherheitslücken im Webbrowser Chrome abdichten, von denen eine in freier Wildbahn bereits von Angreifern missbraucht wird. Wer Chrome nutzt, sollte sicherstellen, dass der Browser in aktueller Version läuft.



Phishing aktuell



- Mailversand (intern) am 04.07.25 ca. 13:39h an 175 Personen.
- Phishing URL:
→ <https://uni-hohenheim.bwsyncandshare.de/idp/profile/SAML2/Redirect/SSO/?execution=e1s2>
→ <https://uni-hohenheim.bwsyncandshare.de/idp/profile/SAML2/Redirect/SSO/?execution=e1s2>
- Korrekte URL:
→ https://bwsyncandshare.kit.edu/apps/user_saml/saml/selectUserBackEnd?redirectUrl=



Titel		Gesendet	Geöffnet	Angeclickt	Verletzlich	Kompromittiert	Mehrfachclick
	Juli Kampagne (Studierende) Aus: <u>SharePoint</u>	6102	1160	143	0		32
	Juli Kampagne (Beschäftigte) Aus: <u>SharePoint</u>	3932	756	155	0		14
	März Kampagne (Studierende) Aus: <u>Microsoft Services</u>	24928	8208	4164	0		725
	März Kampagne (Beschäftigte) Aus: <u>Microsoft Services</u>	7500	1740	356	0		66

Microsoft



FAHRER		ZEIT
1	  B. HEALY EF EDUCATION - EASYPOST	41:01:13
2	  T. POGACAR UAE TEAM EMIRATES - XRG	+29
3	  R. EVENEPOEL SOUDAL QUICK-STEP	+1:29
4	  J. VINGEGAARD TEAM VISMA-LEASE A BIKE	+1:46
5	  M. JORGENSON TEAM VISMA-LEASE A BIKE	+2:06
6	  K. VAUQUELIN ARKÉA - B&B HOTELS	+2:26
7	  O. ONLEY TEAM PICNIC POSTNL	+3:24
8	  F. LIPOWITZ RED BULL-BORA-HANSGRÖHE	+3:34
9	  P. ROGLIC RED BULL-BORA-HANSGRÖHE	+3:41
10	  T. JOHANNESSEN	+5:03



• Ajax:

- zunächst Abkürzung für
Asynchronous JavaScript And XML
- Jesse J. Garrett (AdaptivePath)
 - <http://www.adaptivepath.com/publications/essays/archives/000385.php>
(nicht mehr aktiv)
- nicht an sich neu („Technologiemix“),
aber *marketingmäßig bedeutend*
- heute eigenständiges Synonym, kein Akronym





A screenshot of a web browser displaying the Adaptive Path website. The browser's address bar shows the URL www.adaptivepath.com/ideas/ajax-new-approach-web-applications. The website's header includes the 'adaptive path' logo and navigation links for 'Work', 'Ideas', 'Events', 'About', and 'Contact'. A search bar is located on the right. The main content area features a large title 'Ajax: A New Approach to Web Applications' dated February 18, 2005, by Jesse James Garrett. The article text discusses the challenges of web interaction design compared to desktop software. To the right of the article is a sidebar with 'ABOUT JESSE' (President) and a list of 'MORE IDEAS FROM JESSE' including 'Ten Years Later: Way Beyond Digital', 'Congratulations About.me!', 'Weighing the Pros and Cons—Professionals and Consumers, That Is', 'iPad Frees Designers From the Tyranny of QWERTY', and 'Facebook and the User Experience of Privacy'.

Ajax: A New Approach to Web Applications

February 18, 2005

BY JESSE JAMES GARRETT 0 Comments

If anything about current interaction design can be called “glamorous,” it’s creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn’t on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.

Despite this, Web interaction designers can’t help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web’s rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.

That gap is closing. Take a look at [Google Suggest](#). Watch the way the suggested terms update as you type, almost instantly. Now look at [Google Maps](#). Zoom in. Use your cursor to grab the map and scroll around a bit. Again, everything happens almost instantly, with no waiting for pages to reload.

ABOUT JESSE
President

MORE IDEAS FROM JESSE

[Ten Years Later: Way Beyond Digital](#)
March 2, 2011

[Congratulations About.me!](#)
December 20, 2010

[Weighing the Pros and Cons—Professionals and Consumers, That Is](#)
October 27, 2010

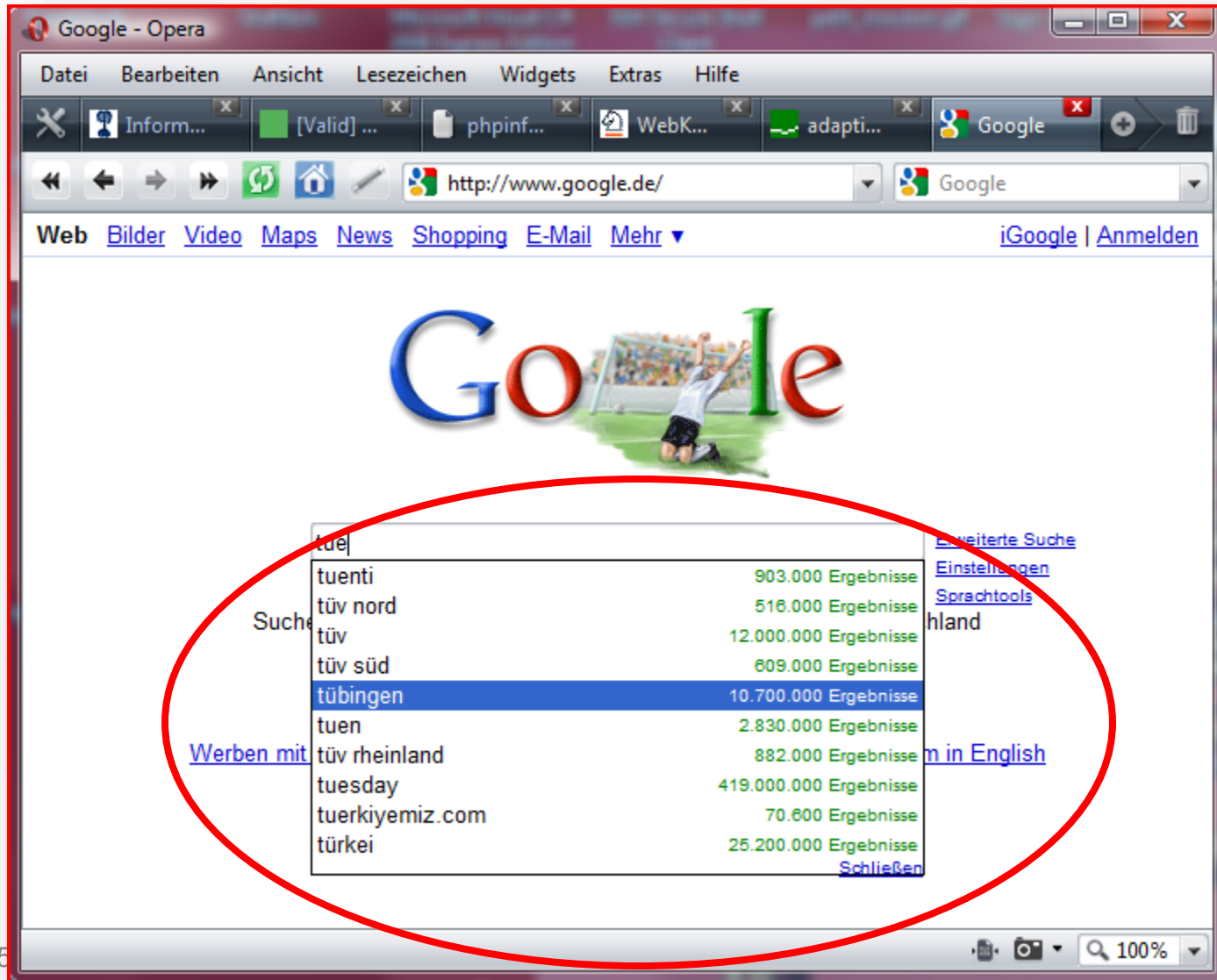
[iPad Frees Designers From the Tyranny of QWERTY](#)
June 11, 2010

[Facebook and the User Experience of Privacy](#)
June 3, 2010



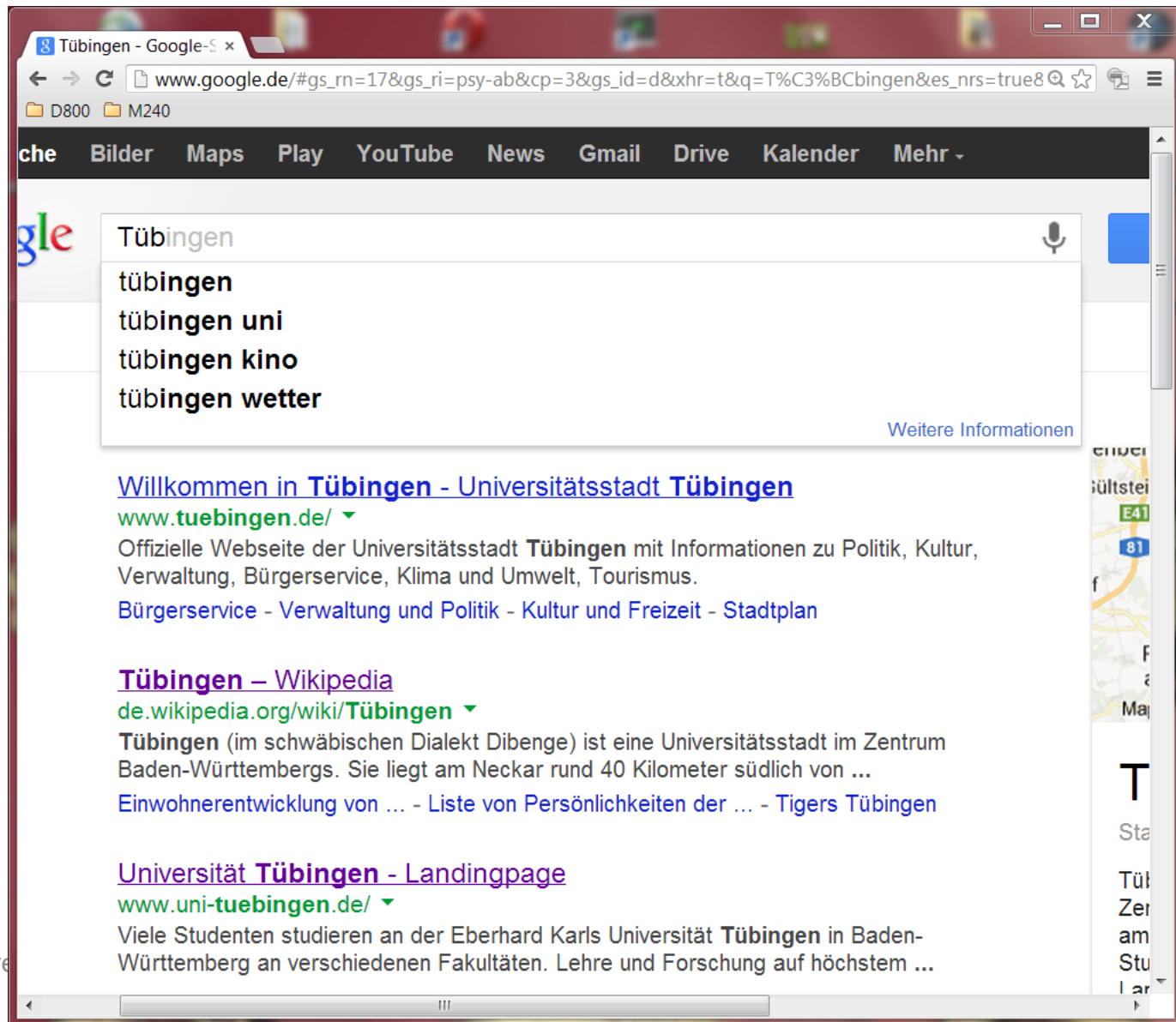


Beispiel für Ajax





Beispiel für Ajax





Automatische Vervollständigung bei der Suche nutzen

Beim Eingeben von Suchbegriffen bei Google finden Sie das Gesuchte schneller, wenn Sie auf die automatischen Vervollständigungen achten. Diese Vervollständigungen werden Ihnen als mögliche Suchbegriffe angezeigt, die Sie verwenden können. Sie hängen mit den Suchbegriffen zusammen, die Sie bereits eingegeben haben, und wurden auch schon von anderen Nutzern verwendet.

So funktioniert die automatische Vervollständigung

Vervollständigungen werden anhand von Faktoren wie der Häufigkeit und Aktualität von Suchbegriffen generiert. Wenn Sie eine Vervollständigung auswählen, wird nach diesem Begriff gesucht.

Woher stammen die Begriffe der Vervollständigungen?

- Aus Ihrer aktuellen Sucheingabe
- Aus relevanten Google-Suchen, die Sie in der Vergangenheit ausgeführt haben (sofern Sie in Ihrem Google-Konto angemeldet waren und [Web- & App-Aktivitäten](#) aktiviert hatten).
- Aus Suchanfragen anderer Nutzer, u. a. zu Topmeldungen 📈. Topmeldungen sind beliebte Themen in Ihrer Region, die sich im Laufe des Tages ändern. Sie hängen nicht mit Ihrem Suchverlauf zusammen. Topmeldungen [finden Sie bei Google Trends](#) 🔗.

Hinweis: Vervollständigungen von Suchanfragen stellen keine Antwort auf Ihre Suche dar. Außerdem sind sie keine Aussagen von anderen Nutzern oder von Google zu Ihren Suchbegriffen.





technische Grundlage

- zunächst: proprietäre Lösung von Microsoft für IE 5 mit ActiveX
- allgemein:
JavaScript-Objekt
XMLHttpRequest
 - wird von allen Browsern unterstützt
 - Firefox ab 1.0, Opera ab 8, Safari ab 1.2, ...
 - auch IE ab Version 7 😊

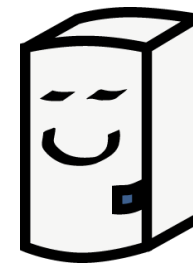


Ablauf einer Ajax-Anfrage

- Browser kann JavaScript-gesteuert kontinuierlich weitere Anfragen an Server stellen (asynchron!)
- Neuladen eines Teils des Browsers



Browser



Datenhaltung

Ajax



HTTP, XML, JSON, ...

Webserver



Detail der Abfrage

- drei Schritte
 - Erzeugung des XMLHttpRequest-Objektes:
`var http = new XMLHttpRequest();`
 - Methode `open` aus dieser Klasse
 - Parameter: HTTP-Methode (GET/POST), Server-URL, synchron (false) oder asynchron (true)
 - Anzeige des Ergebnisses über JavaScript-Event `onreadystatechange`

(Variable `readyState`: 0: nicht initialisiert, 1: Laden, 2: fertig, 3: warten, 4: beendet)



Beispiel 1

- Nachladen einer Datei mittels Ajax:
 - Website wird geladen
 - Ajax fügt asynchron eine Datei (HTML-Code) ein





Web - GIT Ajax/ajaxtest.html - Eclipse

File Edit Source Navigate Search Project Run Window Help

Quick Access

Java Perl JavaScript PHP Ruby Resource PyDev Web

helloworld.php ajaxtest.html

```
8
9      <script type="text/javascript"><!--
10      var http = null;
11      // Initialisieren http-Objekt und Abfangen alter IE
12      if (window.XMLHttpRequest)
13          http = new XMLHttpRequest();
14      else if (window.ActiveXObject)
15          http = new ActiveXObject("Microsoft.XMLHTTP");
16      else alert( "Browser unterstuetzt kein Ajax");
17      }
18
19      if (http != null) {
20          // asynchrone http-Verbindung zum Server
21          http.open("GET", "ajaxFile.html", true);
22          // bei Aenderung von readyState wird "ausgeben" aufgerufen
23          http.onreadystatechange = ausgeben;
24          // Absenden der Anfrage
25          http.send(null);
26      } // if
27
28      function ausgeben() {
29          if (http.readyState == 4) {
30              document.getElementById("ausgabe").innerHTML =
31                  http.responseText; // Rueckgabe des HTTP-Requests als Text
32          } // if
33      } // function
34      //--></script>
35  </head>
36  <body>
37      <hr><center><h2>Grundlagen Internet-Technologien</H2><h3>Ajax</h3><hr>
38      HTML-Ausgabe <u>um</u>ber Ajax (asynchron) vom Server:
39
40      <div id="ausgabe"></div>
41
42      </center><hr>
43  </body>
44 </html>
```

html/#text | Writable | Smart Insert | 2:7



```
<!DOCTYPE html>
<html>
```

```
<head>
```

```
<title>Grundlagen Internet-Technologien: Ajax</title>
```

```
<link rel="stylesheet" type="text/css"
      href="../css/webkompendium.css">
```

```
<script type="text/javascript"><!--
```

```
var http = null;
```

```
// Initialisieren http-Objekt und Abfangen alter IE
```

```
if (window.XMLHttpRequest)
```

```
    http = new XMLHttpRequest();
```

```
else {if (window.ActiveXObject)
```

```
    http = new ActiveXObject("Microsoft.XMLHTTP");
```

```
    else alert( "Browser unterstuetzt kein Ajax");
```

```
}
```

```
if (http != null) {
```

```
    // asynchrone http-Verbindung zum Server
```

```
    http.open("GET", "ajaxFile.html", true);
```

```
    // bei Aenderung von readyState wird "ausgeben" aufgerufen
```

```
    http.onreadystatechange = ausgeben;
```

```
    // Absenden der Anfrage
```

```
    http.send(null);
```

```
} // if
```

```
function ausgeben() {
```

```
    if (http.readyState == 4) {
```

```
        document.getElementById("ausgabe").innerHTML =
```

```
        http.responseText;    // Rueckgabe des HTTP-Requests als Text
```

```
    } // if
```

```
} // function
```

```
//--></script>
```

```
</head>
```

```
<body>
```

```
<hr><center><h2>Grundlagen Internet-Technologien</H2><h3>Ajax</h3><hr>
```

```
HTML-Ausgabe &uuml;ber Ajax (asynchron) vom Server:
```

```
<div id="ausgabe"></div>
```

```
</center><hr>
```



zrvwa01@infodienste =>
zrvwa01@infodienste => more ajaxFile.html
<p>Grundlagen Internet-Technologien:
Ajax</p>
Mehr Dynamik im Web mit Ajax</p>
<p>Tour de France 2025</p>





Grundlagen Internet-Technologien: / X

Nicht sicher http://134.2.6.167/~zrvwaC 170% Suchen

Grundlagen Internet-Technologien

Ajax

HTML-Ausgabe über Ajax (asynchron) vom Server:

Grundlagen Internet-Technologien:
Ajax Mehr Dynamik im Web mit Ajax

Tour de France 2025





Beispiel 2: Kommunikation mit Server

- Client: Ajax-Request, welcher Parameter enthält
- Server: PHP-Script zum Auslesen der Parameter, formatiertes Rücksenden an Client





```
<!DOCTYPE html>
<html>

  <head>

    <title>Grundlagen Internet-Technologien: Parameteruebergabe mit Ajax</title>
    <link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
    <link rel="shortcut icon" href="/css/favicon.ico">

    <script type="text/javascript"><!--
      var http = null;
      // Initialisieren http-Objekt und Abfangen alter IE
      if (window.XMLHttpRequest) {
        http = new XMLHttpRequest();
      } else if (window.ActiveXObject) {
        http = new ActiveXObject("Microsoft.XMLHTTP");
      }

      if (http != null) {
        // Parameteruebergabe mit GET
        http.open("POST", "parameter.php?n=42&text=Tuebingen", true);
        // bei Aenderung von readyState wird "ausgeben" aufgerufen
        http.onreadystatechange = ausgeben;
        // Ausgabe
        http.send(null);
      } // if

      function ausgeben() {
        if (http.readyState == 4) {
          document.getElementById("Ausgabe").innerHTML =
            http.responseText;
        } // if
      } // function
    //--></script>
  </head>

  <body>

    <hr><center><h2>Grundlagen Internet-Technologien<BR>
    Parameter&uuml;lbergabe mit Ajax an Server (PHP)</h2><hr>
    </center>

    <div id="Ausgabe"></div>
```



<!--

Grundlagen Internet-Technologien:
PHP-Skript zur Ausgabe aller mit GET oder POST
uebergegebener Parameter

-->

<h3>GET</h3><code>

<?php

foreach(\$_GET as \$key => \$value)

echo "Parameter \$key hat Wert \$value
";

?>

</code>

<h3>POST</h3><code>

<?php

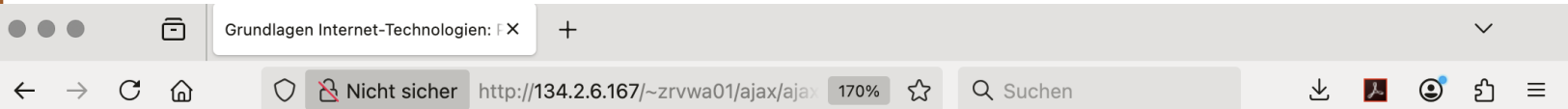
foreach(\$_POST as \$key => \$value)

echo "Parameter \$key hat Wert \$value
";

?>

</code>





Grundlagen Internet-Technologien

Parameterübergabe mit Ajax an Server (PHP)

GET

Parameter n hat Wert 42

Parameter text hat Wert Tuebingen

POST





Übersicht XMLHttpRequest-Objekt

- Konstruktor
- Methoden
 - `abort`
 - `getAllResponseHeaders`
 - `getResponseHeader`
 - `open`
 - `send`
- Eigenschaften
 - `onreadystatechange`
 - `readyState`
 - `responseText`
 - `responseXML`
 - `status`
 - `statusText`



Probleme von Ajax

- Ajax hat auch Kritikpunkte
 - Usability: beispielsweise haben die Browsertasten vor/zurück nun eine unlogische Bedeutung
 - auch Ajax-Site ist nicht mehr zustandslos!
 - Links setzen: man kann nur auf den Startzustand einer Ajax-Anwendung einen Link setzen





Websockets

- Alternative zu AJAX: Websockets
 - Bestandteil von HTML5
 - ähnlich, aber anders
 - *kontinuierliche* bidirektionale Verbindung
 - typischerweise **json** als Austauschformat







Authentifizierung

- wie authentifiziert sich ein Nutzer?
- die einfache Antwort: Kennwort
 - Regeln? (“acht Zeichen mit Sonderzeichen und Ziffern“)
 - Ablauf?
- Biometrie?



Authentifizierung im Web

- besondere Herausforderung:
Authentifizierung im Web
- „Shibboleth“ (SAML2): Kennwort verlässt die
Heimatinrichtung nie
 - funktioniert nur mit http/https



```
graph LR; A[Login] --> B[Lokalisierungsdienst  
(Discovery Service)]; B --> C{Benutzer bekannt?}; C -- nein --> A; C -- ok --> D{Benutzer berechtigt?}; D -- ok --> E[Zugriff ja/nein]; D -- Benutzerrollen --> E;
```

The diagram illustrates the process flow of the Discovery Service (Lokalisierungsdienst). It starts with a 'Login' step, which leads to the 'Lokalisierungsdienst (Discovery Service)'. From there, the process asks 'Benutzer bekannt?' (User known?). If the answer is 'nein' (no), it loops back to 'Login'. If 'ok' (yes), it asks 'Benutzer berechtigt?' (User authorized?). If the answer is 'ok', it proceeds to 'Zugriff ja/nein' (Access yes/no). If the answer is 'Benutzerrollen' (User roles), it also proceeds to 'Zugriff ja/nein'.



Benutzerrollen



EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



LOGIN @ UNI-TUEBINGEN

Shibboleth · Identity Provider

Dies ist die zentrale Anmeldeseite für das Single Sign-On der Universität Tübingen.

Ein Dienst hat angefordert, dass Sie sich über Ihr persönliches Benutzerkonto der Universität Tübingen anmelden.

Login-ID

› Passwort vergessen?

› Hilfe benötigt?

› Nutzungsbedingungen

Zentrales Passwort

☒ Anmeldung nicht speichern

☐ Löschen der früheren Einwilligung zur Weitergabe Ihrer Informationen an diesen Dienst.

Anmelden





An den Dienst zu übermittelnde Informationen

Benutzername	zrvwa01
Vorname(n)	Thomas
E-Mail-Adresse	thomas.walter@uni-tuebingen.de
Bereichsspezifische Zugehörigkeit	member@uni-tuebingen.de
Berechtigung	urn:mace:dir:entitlement:common-lib-terms http://moodle.uni-tuebingen.de/user
Nachname	Walter

Die oben aufgeführten Informationen werden an den Dienst **moodle02.zdv.uni-tuebingen.de** weitergegeben, falls Sie fortfahren. Sind Sie einverstanden, dass diese Informationen bei jedem Zugriff auf diesen Dienst an ihn weitergegeben werden?

Wählen Sie die Dauer, für die Ihre Entscheidung zur Informationsweitergabe gültig sein soll:

☒ **Bei nächster Anmeldung erneut fragen.**

› Ich bin einverstanden, meine Informationen dieses Mal zu senden.

☐ **Erneut fragen, wenn sich die Informationen ändern, welche diesem Dienst weitergegeben werden.**

› Ich bin einverstanden, dass dieselben Informationen in Zukunft automatisch an diesen Dienst weitergegeben werden.

Diese Einstellung kann jederzeit mit der Checkbox auf der Anmeldeseite widerrufen werden.

Ablehnen

Akzeptieren



...und nun...

- haben wir wesentliche Erweiterungen für die Web-Programmierung kennen gelernt, insbesondere Ajax

- als nächstes:

Zusammenfassung

