

Das World Wide Web (WWW)

Seit seiner Entwicklung im Jahr 1989 durch die Forschergruppe um Sir Tim Berners-Lee (Cailliau, 1995), wurde aus dem World Wide Web die wohl bekannteste und meist benutzte Anwendung des Internets. In vielen Fällen wird der Begriff Internet, obwohl er eigentlich nur die darunterliegende Netzwerkinfrastruktur bezeichnet, als Synonym für das World Wide Web verwendet. Beim WWW handelt es sich in seiner Grundstruktur um eine verteilte Sammlung von Dokumenten, welche unter Verwendung von Internet-Protokollen über eine Anwendung abrufbar sind.

Diese Dokumente sind untereinander mittels #Hyperlinks verknüpft und bilden dadurch ein weltweites Netz an Informationen. Die Anwendung, mit der auf diese Dokumente, man spricht häufig von Webseiten, zugegriffen werden kann, wird als Browser bezeichnet. Bekannte Browser sind der Internet Explorer von Microsoft, Chrome von Google, Firefox von Mozilla, Safari von Apple oder Opera von Opera Software.

“

!

Weitere Browser können Sie beispielsweise bei Wikipedia finden (http://de.wikipedia.org/wiki/Liste_von_Webbrowsern) oder in der L3T-Gruppe bei Diigo https://groups.diigo.com/group/l3t_20 unter #l3t_webtech.

Jedes Dokument im Web wird durch eine URL (Uniform Resource Locator) identifiziert. Diese URL besteht im Web zumeist aus drei Komponenten: Protokoll, Host und Pfad.

“

!

Die Komponenten der URL sind folgendermaßen angeordnet: `protokoll://host/pfad`. Der Host-Teil gibt die Adresse des Webserver, auf dem die Dokumente gespeichert sind, an. Er kann darüber hinaus einen Port, das heißt einen ‚Anschluss‘ am Server (zum Beispiel <http://www.example.org:80>), beinhalten. Der Pfad-Teil kann um einen query string, das heißt zusätzliche Informationen wie die Inhalte einer Suchanfrage (z. B.

<http://www.example.org:80/demo/example.cgi?land=de&stadt=aa>),
erweitert sein.

“

?

Identifizieren Sie die drei Teile der URL

<http://de.wikipedia.org/wiki/Wikipedia:Hauptseite>

Hypertext Transfer Protocol (HTTP)

Der Browser verständigt sich mit dem Webserver, auf dem die Dokumente gespeichert sind, über das **Hypertext Transfer Protocol**. Wie jedes Protokoll beschreibt HTTP den Aufbau der Nachrichten vom Client an den Server. Die Kommunikation erfolgt immer über Anfragen des Clients an den Server und zugehörige Antworten. Alle Nachrichten werden als Klartext, also unverschlüsselt, übermittelt.

Erstes, nachstehendes, Element wird ans Ende der vorherigen Seite gestellt

Durch diesen Aufbau der Kommunikation ist eine zentrale Problematik der Entwicklung von Webanwendungen bedingt: Der Server kann nicht von sich aus mit dem Client kommunizieren, da er immer auf eine Anfrage angewiesen ist. Wartet eine Anwendung nun auf ein bestimmtes Ereignis (zum Beispiel das Ergebnis einer aufwendigen Berechnung), kann der Server den Client nicht über dessen Eintritt verständigen, sondern der Client muss in regelmäßigen Abständen Anfragen über den Status an den Server schicken. Dies bringt natürlich einerseits zusätzlichen Netzwerkverkehr und andererseits zusätzliche Arbeitszeit des Servers mit sich. Bei HTTP handelt es sich um ein zustandsloses Protokoll. Alle Anfragen sind somit voneinander unabhängig zu betrachten. Für die meisten Anwendungen im Web ist es aber notwendig, mehrere Anfragen in Zusammenhang zueinander zu sehen. So besteht zum Beispiel der Bestellvorgang in einem Webshop durchaus aus mehreren sequentiellen Anfragen (Auswahl der Waren, Eingabe der Adresse, Bestätigung des Kaufes). Solche Zusammenhänge können nicht durch das Protokoll selbst verwaltet werden, sondern müssen durch die Anwendungen gehandhabt werden, welche ihre Daten über HTTP übertragen. Hierzu können Sitzungen (Sessions) verwendet werden, in denen eine eindeutige ID mit jeder Anfrage versendet wird. Eine andere Möglichkeit ist die Verwendung von Cookies, mit denen der Browser persistente Informationen (zum Beispiel Kundinnen- und Kundendaten) zu einem Webserver lokal speichern kann. Üblicherweise beginnen solche Sessions mit der Anmeldung mittels Login und Passwort und enden durch Abmeldung oder wenn der Browser geschlossen wird.

!

HTTP ist ein zustandsloses Anfrage-/Antwort-Protokoll und dient zur Übermittlung von Daten im WWW.

HTTPS

Die Übertragung der Daten in Klartextform bei HTTP ist nicht in jedem Fall wünschenswert. So könnten sensible Daten wie zum Beispiel Kennwörter durch beliebige ‚Zuhörer /innen‘ abgefangen werden. Aus diesem Grund wurde das **Hypertext Transfer Protocol Secure (HTTPS)** als Verfahren entwickelt, um Daten im Web abhörsicher zu übermitteln.

Das Protokoll ist an sich identisch zu HTTP, allerdings werden die Daten mittels des Protokoll Secure Socket Layer bzw. Transport Layer Security (SSL/TLS) verschlüsselt. Zu Beginn der verschlüsselten Verbindung muss sich bei HTTPS der Server identifizieren. Dies geschieht über ein Zertifikat, welches die Identität bestätigen soll. Bei der ersten Anfrage an einen Webserver kann es notwendig sein, dass die Authentizität dieses Zertifikates bestätigt werden muss.

HTML

Die eigentlichen Dokumente, die über HTTP vom Server an den Client übermittelt werden, sind meist HTML-Dokumente. Die **Hypertext Markup Language (HTML)** ist eine Auszeichnungssprache, welche Dokumentinhalte beschreibt. Der Browser zeigt anhand von HTML und der mitgelieferten Formatierungsinformation diese Dokumente an. Üblicherweise wird also das Aussehen der HTML-Dokumente in einer separaten Datei beschrieben. Diese **Cascading Style Sheets (CSS)** werden ebenfalls vom Browser über HTTP angefordert. Mit HTML5 lassen sich neben Texten, Tabellen und Bildern auch multimediale Inhalte (Videos, Animationen) beschreiben, die dann unmittelbar vom Browser wiedergegeben werden können. Für die Darstellung solcher Inhalte waren bisher zusätzliche Softwareprodukte wie etwa Adobe Flash notwendig.

Webanwendungen

Im Laufe der Zeit haben sich die Anforderungen an Informationssysteme im Web von der bloßen Zurverfügungstellung von Dokumenten in Richtung ausgefeilter Programmlogik weiterentwickelt. Wie im Kapitel #infosysteme aufgezeigt, erfordern natürlich auch Informationssysteme für das Lernen und Lehren solch eine Programmlogik. Diese Programme, die auf einem Webserver ausgeführt werden, werden **Webanwendungen** genannt.

Wie bei statischen Webseiten, das heißt reinen HTML-Dokumenten, wird ein Browser zur Interaktion mit dem Webserver verwendet und die Daten werden mittels HTTP(S) übermittelt. Im Unterschied zu einer einfachen Webseite übermittelt der Webserver beim Aufruf der URL allerdings nicht ein bereits vorliegendes Dokument, sondern ruft ein Programm auf, welches aus einer Vorlage für Inhalt und Formatierung sowie aus variablen Daten dynamisch ein Dokument erstellt und an den Client übermittelt.

Der Browser übernimmt in diesem Fall also die Rolle der Benutzer/innen-Schnittstelle für ein auf dem Server ausgeführtes Programm. Mit Webanwendungen kann komplexe Software realisiert werden, welche clientseitig nur einen Webbrowser bzw. entsprechende Plug-Ins benötigt. Für die Entwicklerin oder den Entwickler der Software wird die Softwarewartung erheblich vereinfacht, da Anpassungen nur am Server erfolgen müssen. Für die Benutzerinnen und Benutzer ergibt sich der Vorteil, dass die zusätzliche Installation einer Software weitestgehend entfällt.

“

!

Webanwendungen sind Computerprogramme, die auf einem Webserver laufen und den Browser als Benutzerschnittstelle verwenden.

Andererseits hat dieser Ansatz auch einige Nachteile. So ist eine ständige Internetverbindung mit ausreichender Bandbreite nötig. Zudem kann der Server, bedingt durch die Tatsache, dass HTTP auf dem Anfrage-/Antwort-Prinzip basiert, nicht selbstständig Informationen an die Clients senden, sondern ist auf periodische Anfragen angewiesen.

Zu guter Letzt bedeutet unter anderem das zustandslose Design des Protokolls, dass die Sicherheit der Webanwendung ein nicht zu vernachlässigender zentraler Bestandteil der Entwicklung der Software sein muss, um vor einer Vielzahl von möglichen Angriffsszenarien, wie das Ausspähen von Passwörtern, geschützt zu sein.

Webservices

Webservices sind eine spezielle Art von Webanwendungen, die der Bereitstellung von Daten für andere Applikationen dienen (World Wide Web Consortium, 2004). Sie sind üblicherweise Application Programming Interfaces (API) und stellen als solche eine einheitlich definierte Schnittstelle für fremde Anwendungen zur Verfügung, um auf die Funktionalität des Service zuzugreifen, indem der Service Daten bereitstellt.

In Zusammenhang mit Informationssystemen für das Lernen und Lehren bietet die Integration solcher Webservices innovative Ansätze für die Einbindung externer Ressourcen und Funktionalitäten in ein derartiges Informationssystem (Vossen & Westerkamp, 2003). Anders als bei Webanwendungen werden bei Webservices üblicherweise keine HTMLDokumente vom Server

geladen, da für die aufrufenden Applikationen Formatierungen sowie die Lesbarkeit für menschliche Benutzer/innen irrelevant sind und der Fokus rein auf dem Inhalt liegt.

Der Grundgedanke von Webservices ist die Möglichkeit der automatischen Verarbeitung von Daten im Web durch Softwareagenten, welche lose gekoppelt Aufgaben für Benutzer/innen ausführen. Webservices stellen ihre detailliert beschriebene Funktionalität hierbei anderen Anwendungen zur Verfügung, seien dies autonome Agenten, Webanwendungen oder andere Webservices. Dadurch können Entwickler/innen bereits bestehende Funktionalitäten in ihren eigenen Anwendungen verwenden (Mashup, siehe Abschnitt 6).

Die technologische Umsetzung von Webservices erfolgt meist über eine der folgenden Möglichkeiten:

- **SOAP/WSDL:** Ein flexibles System, in dem Nachrichten über das **Simple Object Access Protocol (SOAP)** ausgetauscht werden. Wie diese Nachrichten für die einzelnen Webservices aussehen, wird über die **Web Services Description Language (WSDL)** beschrieben. Anfragen und Antworten sind in XML, einer allgemeinen Auszeichnungssprache für hierarchische Daten, geschrieben. In den Nachrichten werden die gewünschten Funktionen des Webservices aufgerufen, die Antworten werden vom Server retourniert.
- **Representational State Transfer (REST)** bezeichnet eine Technologie, in der jede einzelne Funktion des Webservices über eine individuelle URL aufgerufen wird. Die Kommunikation erfolgt zustandslos, ist also nicht von Sitzungen, Benutzer/innen-Daten oder ähnlichem abhängig

“
?”

Beschreiben Sie die Unterschiede zwischen Webanwendungen und Webservices
Vergleichen Sie Ihre Antwort mit Tabelle 2.

Webanwendung	Webservice
Zielgruppe: (menschliche) Benutzer/innen	Zielgruppe: andere Applikationen
Ausgabe als HTML	Ausgabe als XML o.ä. für Maschinen optimierte Formate
Clientseitiges Programm: Browser	Verarbeitung in anderen Applikationen

Tab. 2: Unterschiede von Webanwendung und Webservice

Webservices verwenden als Ausgabeformat häufig XML-Dokumente. **Die EXtensible Markup Language (XML)** beschreibt eine sehr allgemeine, flexible und für individuelle Bedürfnisse

erweiterbare Auszeichnungssprache. Wie HTML dient sie zur Darstellung strukturierter Inhalte, ist aber in erster Linie nicht für die menschenlesbare Darstellung, wie beispielsweise in Webservices, gedacht. Sie ist von konkreten Plattformen und Implementierungen unabhängig und durch ihre Charakteristik als Metasprache vielseitig verwendbar.

Letzteres bedeutet, dass sich auf Basis von XML durch die anwendungsspezifische Definition und Verwendung eines Schemas Datenaustauschformate, man spricht von Dialekten, für spezialisierte Anwendungen definieren lassen.

Revision #2

Created 28 February 2025 21:10:48 by Bernd Grabner

Updated 13 February 2026 14:18:14 by Github Admin