

HTTP 2.0

SPDY vs. Speed+Mobility

Web Performance Optimizations • Ruben Deyhle • 23.06.12

Warum HTTP 2.0?

HTTP ist veraltet?

- HTTP 0.9: 1991
- HTTP 1.0: 1996
- HTTP 1.1: 1999

-  My Computer
-  Network Neighborhood
-  Inbox
-  Recycle Bin
-  Internet Explorer
-  The Microsoft Network
-  Outlook Express

Yahoo! - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print

Address  http://www.yahoo.com Go Links >>



YAHOO!


CLICK For Details
You could win \$10,000!
Weekly Picks

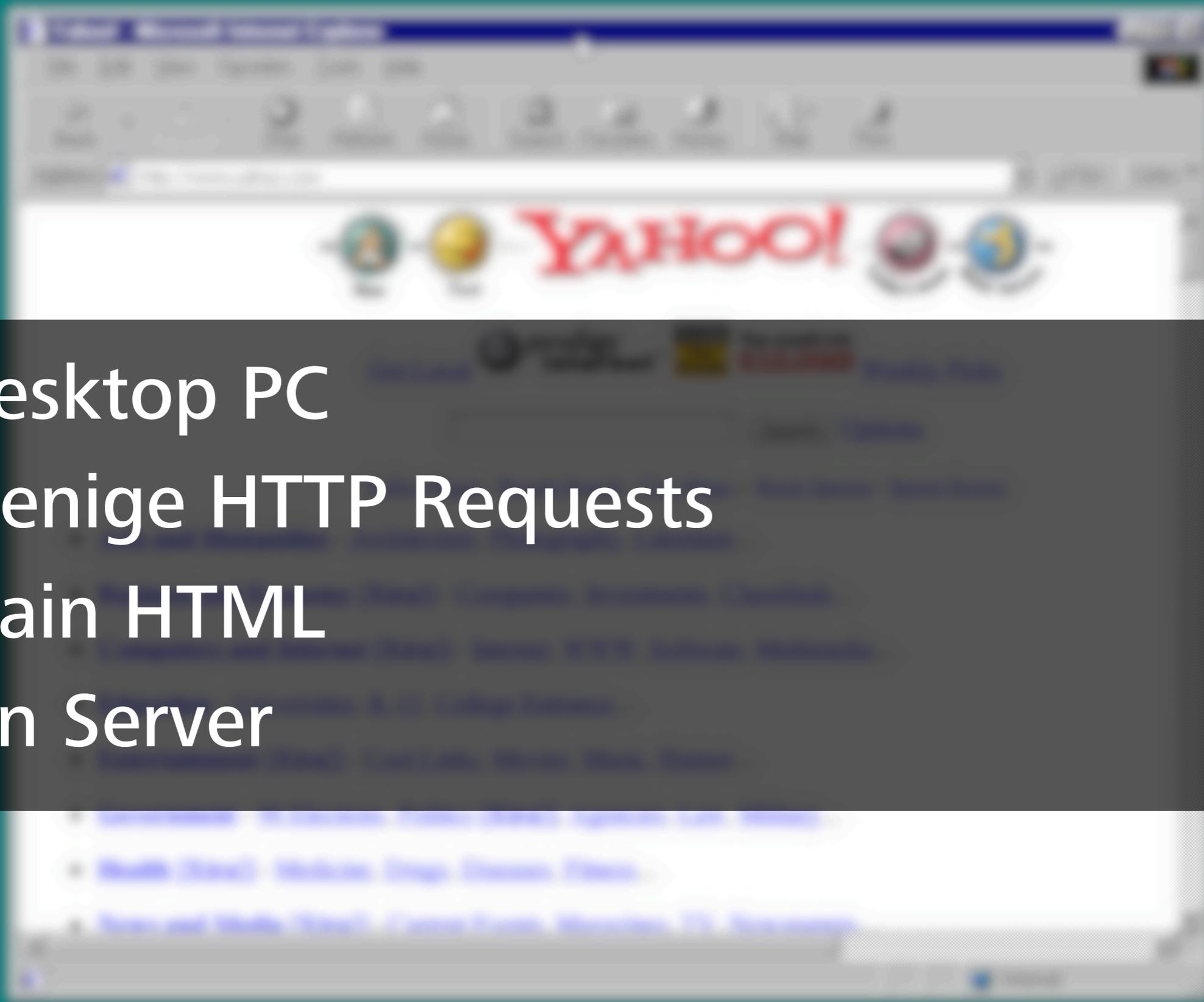
Search
Options

Yellow Pages -
 People Search -
 City Maps --
 Stock Quotes -
 Sports Scores

- [Arts and Humanities](#) - [Architecture](#), [Photography](#), [Literature](#)...
- [Business and Economy \[Xtra!\]](#) - [Companies](#), [Investments](#), [Classifieds](#)...
- [Computers and Internet \[Xtra!\]](#) - [Internet](#), [WWW](#), [Software](#), [Multimedia](#)...
- [Education](#) - [Universities](#), [K-12](#), [College Entrance](#)...
- [Entertainment \[Xtra!\]](#) - [Cool Links](#), [Movies](#), [Music](#), [Humor](#)...
- [Government](#) - [96 Elections](#), [Politics \[Xtra!\]](#), [Agencies](#), [Law](#), [Military](#)...
- [Health \[Xtra!\]](#) - [Medicine](#), [Drugs](#), [Diseases](#), [Fitness](#)...
- [News and Media \[Xtra!\]](#) - [Current Events](#), [Magazines](#), [TV](#), [Newspapers](#)...

Internet

- Desktop PC
- wenige HTTP Requests
- Plain HTML
- Ein Server



Und heute?

Telekom.de 12:41 48%

www.facebook.com/ Google

Willkommen bei Facebook - anmelden, registrieren oder mehr erfahren

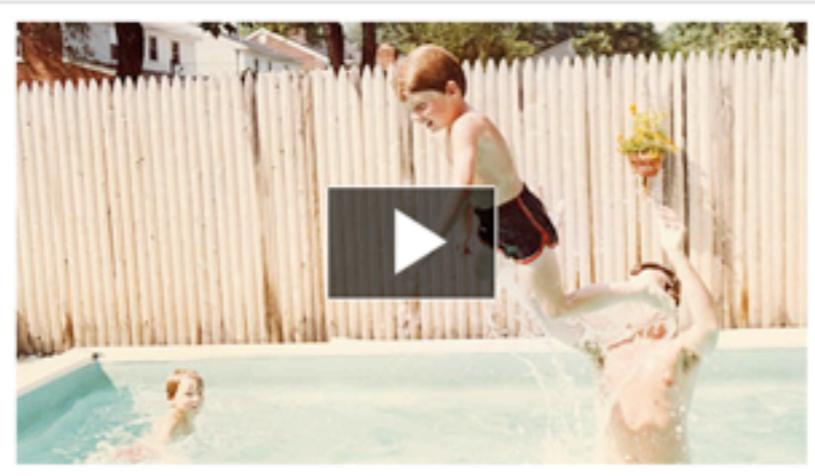
facebook

E-Mail oder Telefon Passwort **Anmelden**

Angemeldet bleiben Passwort vergessen?

Deine Facebook-Chronik

Erzähle deine Lebensgeschichte mit einem neuartigen Profil [Mehr dazu.](#)



Registrieren

Facebook ist und bleibt kostenlos.

Vorname:

Nachname:

Deine E-Mail-Adresse:

E-Mail nochmals eingeben:

Neues Passwort:

Ich bin: **Geschlecht auswählen:**

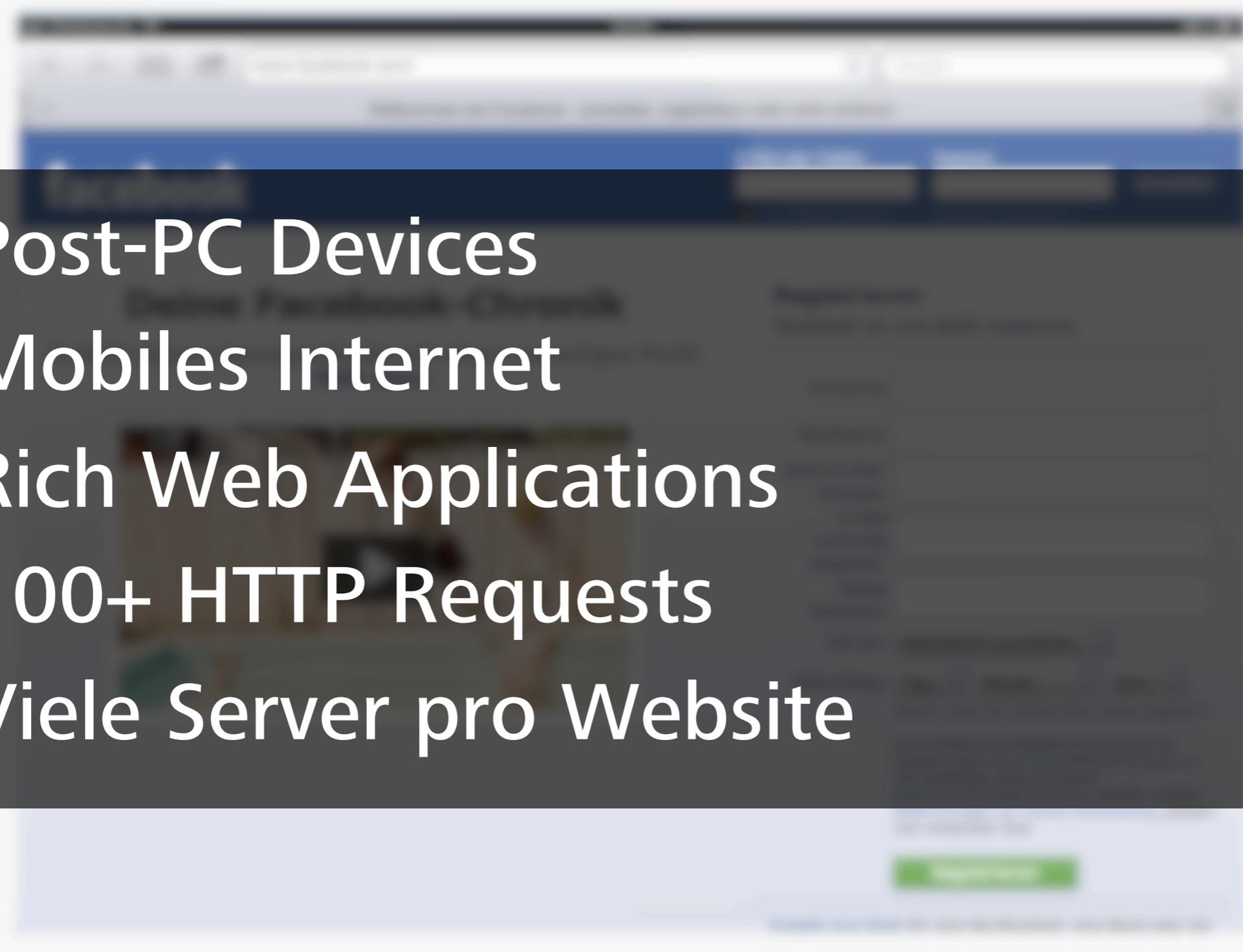
Geburtsdag: **Tag:** **Monat:** **Jahr:**

Warum muss ich meinen Geburtstag angeben?

Durch Klicken auf Registrieren stimmst du unseren [Allgemeine Geschäftsbedingungen](#) zu und bestätigst, dass du unsere [Datenverwendungsrichtlinien](#), einschl. unserer [Bestimmungen zur Cookie-Verwendung](#), gelesen und verstanden hast.

Registrieren

[Erstelle eine Seite für eine Berühmtheit, eine Band oder ein](#)

- 
- Post-PC Devices
 - Mobiles Internet
 - Rich Web Applications
 - 100+ HTTP Requests
 - Viele Server pro Website

Browser

Server

GET index.html

200 OK

GET style.css

200 OK

Alles synchron!

- Optimierungen:
 - alles was wir hier gelernt haben...
 - weniger Requests
 - Verteilen auf mehrere Server
 - Mehrere TCP-Verbindungen parallel

Problem dabei:

- 3-Wege Handshake für jede Verbindung
- TCP Slow Start

- langsam!

HTTP ist veraltet

- für 90er-Websites gemacht
- langsam
- SSL nur optional
 - noch immer viel unverschlüsselter Verkehr

Wir brauchen HTTP 2.0!

Designziele:

- effizienter und schneller als HTTP 1.1
- idealerweise nur 1 TCP-Verbindung
- abwärtskompatibel

Bislang zwei nennenswerte Vorschläge:

- Google: SPDY
- Microsoft: HTTP Speed+Mobility

SPDY

Multiplexing

- nur 1 TCP-Verbindung
- aber: direkt mehrere Requests absenden
- beliebig viele Daten parallel übertragen
- Priorisierung

Browser

Server

GET

GET

GET

200 OK

200 OK

200 OK

Server Push & Hint

- Server Push:
 - Auslieferung ohne Anfrage
 - Kein Caching
- Server Hint:
 - ähnlich, aber Server informiert nur
 - normale Anfrage wenn nicht im Cache

Außerdem:

- Kompression (gzip) für alles
- SSL standardmäßig aktiv

SPDY

- bereits einsatzfähig
- ursprünglich nur HTTP-Aufsatz
- entspricht den HTTP 2.0-Anforderungen
 - SPDY 3.0 bei IETF eingereicht

HTTP Speed+Mobility

HTTP Speed+Mobility

- Microsofts Vorschlag für HTTP 2.0
 - März 2012, noch kaum Details
- Kernziele:
 - nicht nur Browser, auch Apps
 - mobil: weniger Energieverbrauch
- nutzt SPDY und WebSockets

Exkurs: WebSockets

WebSocket

- Real-Time Webanwendungen
- Eigenständiges Protokoll (`ws://`)
- Bidirektionale Verbindung
- Server Push
- kein HTTP-Overhead

Ziele von S+M

- Abwärtskompatibilität
- Strenge Layer-Architektur
- Features optional

Abwärtskompatibilität

- HTTP-Semantik wird beibehalten
- Verwendung von existierenden Standards
- Dadurch keine Probleme mit Proxies etc.
- HTTP 1.1 upgrade-Header (wie WebSockets)
- kein Server-Push, da nicht mit HTTP-Semantik vereinbar

Layer-Architektur

- keine Aufgaben der Transportschicht übernehmen
- Congestion Control bleibt bei TCP

Optionale Features

- Push, Verschlüsselung und Kompression nur als optionale Komponenten
- Client weiß am besten, welche Daten er benötigt
- dadurch flexibler als SPDY
- weniger Last für mobile Clients
 - Akku und Bandbreite sparen

Funktionsweise

1. Verbindungslayer: WebSocket
2. Session Layer: WebSocket-Extension
3. Multiplexing Layer: Multiplexing wie SPDY, aber auch als WebSocket-Extension realisiert
4. HTTP Layer: wie SPDY

Stand

- März 2012: Ankündigung, Draft 1
- 11.05.12: Erster Prototyp, unvollständig
- 15.06.12: Draft 2

- Einige Details noch zu klären
- Derzeit NICHT einsatzfähig

Vergleich

Gemeinsamkeiten

- Multiplexing
 - nur 1 TCP-Verbindung
- Abwärtskompatibilität
- erfüllt HTTP 2.0 Anforderungen

Unterschiede

SPDY

- Speed als Ziel
 - Server-Push
 - Kompression
- für alles
- einsatzbereit

HTTP S+M

- Speed und Mobilität
 - Komponenten optional
 - WebSockets als Basis
- frühes Entwicklungs-
stadium

Insgesamt: hauptsächlich Semantik

Ausblick

HTTP 2.0

- Vermutlich SPDY als Basis
 - Vorschläge von Microsoft mit einbeziehen
 - Juli 2013
-
- Bis dahin: SPDY einsetzen

