



# HTTP2

#OMTalk2015Uckermark

09. November 2015 | Christopher Wagner

## Es geht nicht um ... diese wichtigen Ressourcen

- PageSpeed
  - Rendering-Trees
  - Datenbankoptimierung
  - (Caching)
  - (Serverauslastung)
  - HTTPS everywhere
  - SSL Migration
  - Let's encrypt (lesen!)
- <https://goo.gl/OwMyIU>
  - <https://goo.gl/pWGDmt>
  - <https://goo.gl/jFYynm>  
(T. Schwarz, Freigabe bitten)
  - <https://goo.gl/4m8vUz>
  - <https://goo.gl/ymBTa1>
  - <https://letsencrypt.org/>





## Evolution

...

- 2009
  - 2011 – G-TechTalks - Präse
  - 2011-2014
  - 2014-2015
- Entwicklungsstart
  - <http://y2u.be/TNBkxA313kk>
  - Entwicklung (SPDY 1 – 3.1)
  - HTTP2 (Drafts + Abschluss)  
<https://goo.gl/2riqRQ>



## Was bringt mir das – ein Vergleich (http1 / http2)?

<https://http2.golang.org/gophertiles>

A grid of 180 tiled images is below. Compare:

[\[HTTP/2, 0 latency\]](#) [\[HTTP/1, 0 latency\]](#)  
[\[HTTP/2, 30ms latency\]](#) [\[HTTP/1, 30ms latency\]](#)  
[\[HTTP/2, 200ms latency\]](#) [\[HTTP/1, 200ms latency\]](#)  
[\[HTTP/2, 1s latency\]](#) [\[HTTP/1, 1s latency\]](#)



A grid of 180 tiled images is below. Compare:

[\[HTTP/2, 0 latency\]](#) [\[HTTP/1, 0 latency\]](#)  
[\[HTTP/2, 30ms latency\]](#) [\[HTTP/1, 30ms latency\]](#)  
[\[HTTP/2, 200ms latency\]](#) [\[HTTP/1, 200ms latency\]](#)  
[\[HTTP/2, 1s latency\]](#) [\[HTTP/1, 1s latency\]](#)



## HTTP1.1

... womit Du in Zukunft keine Zeit mehr verschwendest

- Domain-Sharding (sub1., sub2.domain.com)
- CSS Sprites
- Datei-Zusammenfassung
- Datenkompression



## HTTP2 Vorteile ggü. SPDY

### ... im Vergleich

#### HTTP2

- SSL nicht zwingend (aber:)
- Schnellere Verschlüsselung
- Muti Host Multiplexing
- Sichere Kompression (HPACK)
- Verbesserte Priorisierung
- Server push

#### SPDY

- SSL benötigt
- Schnelle Verschlüsselung
- Single Host Multiplexing
- Kompression (inkl. Header)
- Priorisierung
- Server push



## HTTP2 - Vorteile

### In a nutshell ...

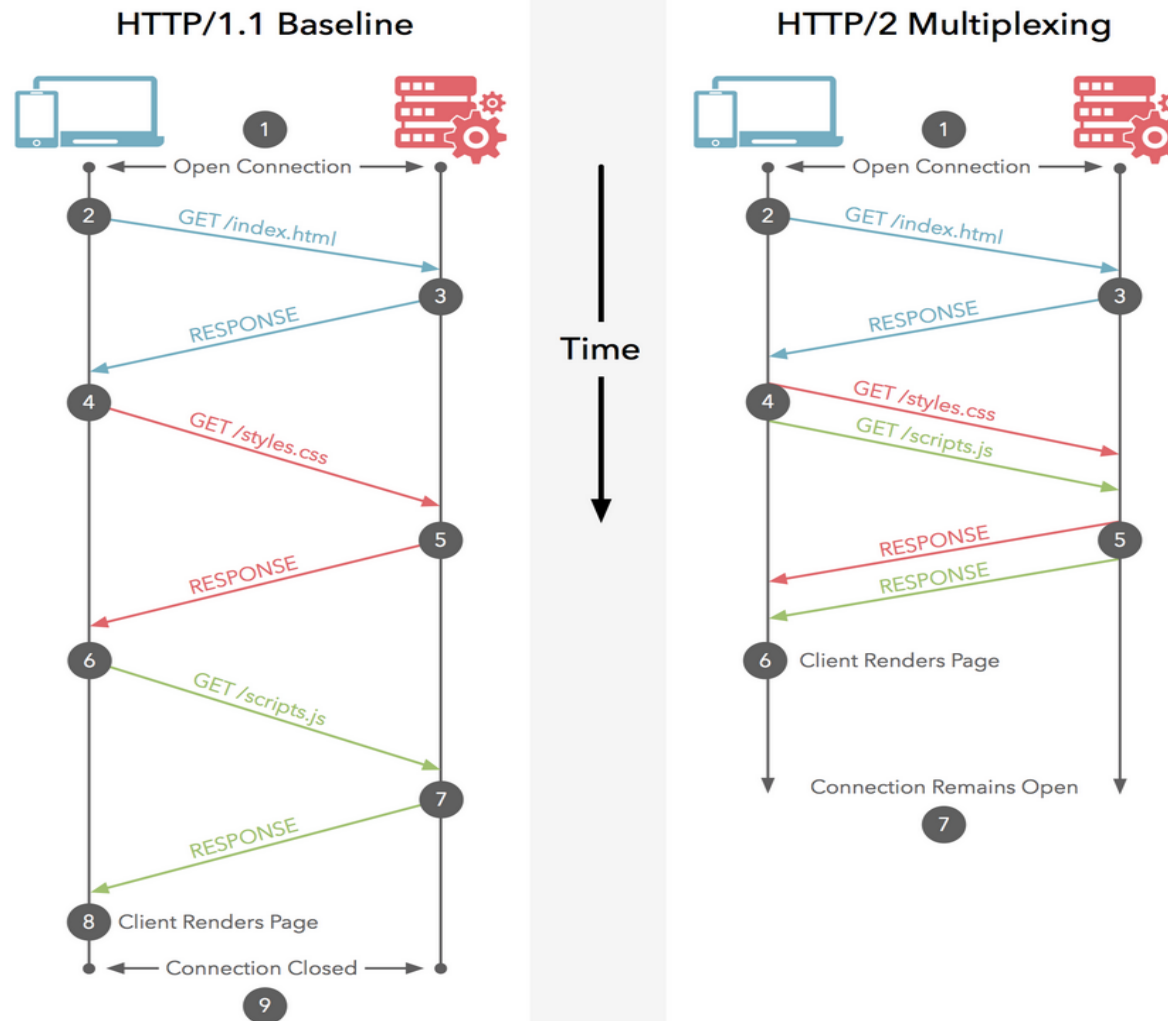
- vollkommen binär
- vollkommen multiplexfähig statt geordnet und blockierend (Wasserfall)
- benutzt EINE Verbindung für Parallelisierung, nicht mehrfache TCP-Connections (Streams)
- nutzt Header – Kompression um den Overhead zu reduzieren
- kann Daten priorisiert übertragen
- erlaubt Server-Push





# HTTP2 - Vorteile

## Warum Multiplexing?



## HTTP2 - Vorteile

### Warum Multiplexing?

- EINE TCP Connection für alle Files vs. MEHRERE TCP Connections für mehrere Files (Reduktion Round Trip Time = schnellere Renderings in mobilen Netzwerken)
- Ressourcen am Server sparen (Bandbreite, Memory, aktive Connections)



## HTTP2 - Vorteile

### Warum Multiplexing? Kein Wasserfall

URL	Status	Größe	Lokale IP	Remote-IP	Zeitlinie
+ GET www.1aseo.de	200 OK	6,1 KB	192.168.2.116:57065	84.200.18.110:443	996ms
+ GET bootstrap.min.css	200 OK	16,7 KB	192.168.2.116:57065	84.200.18.110:443	58ms
+ GET bootstrap-responsive.min.css	200 OK	3,9 KB	192.168.2.116:57065	84.200.18.110:443	57ms
+ GET bootstrap-extended.css	200 OK	2,1 KB	192.168.2.116:57065	84.200.18.110:443	56ms
+ GET bootstrap-frontend.min.css	200 OK	2,0 KB	192.168.2.116:57065	84.200.18.110:443	68ms
+ GET bootstrap-responsive-frontend.i	200 OK	2,2 KB	192.168.2.116:57065	84.200.18.110:443	54ms
+ GET system.css	200 OK	422 B	192.168.2.116:57065	84.200.18.110:443	54ms
+ GET general.css	200 OK	881 B	192.168.2.116:57065	84.200.18.110:443	53ms
+ GET template.css	200 OK	9,5 KB	192.168.2.116:57065	84.200.18.110:443	80ms
+ GET template_pro.css	200 OK	7,7 KB	192.168.2.116:57065	84.200.18.110:443	115ms
+ GET blue.css	200 OK	2,0 KB	192.168.2.116:57065	84.200.18.110:443	77ms
+ GET business.css	200 OK	469 B	192.168.2.116:57065	84.200.18.110:443	76ms
+ GET jsn_mobile.css	200 OK	4,5 KB	192.168.2.116:57065	84.200.18.110:443	120ms
+ GET custom2.css	200 OK	3,4 KB	192.168.2.116:57065	84.200.18.110:443	112ms
+ GET modal.css	200 OK	997 B	192.168.2.116:57065	84.200.18.110:443	110ms
+ GET style-modules.min.css	200 OK	10,3 KB	192.168.2.116:57065	84.200.18.110:443	133ms



## HTTP2 - Vorteile

### Warum Header-Kompression?

- Weniger Bytes, die übertragen werden müssen
- Resultat: schnellere Seiten-Rendingzeiten



# HTTP2 - Vorteile

## Server Push

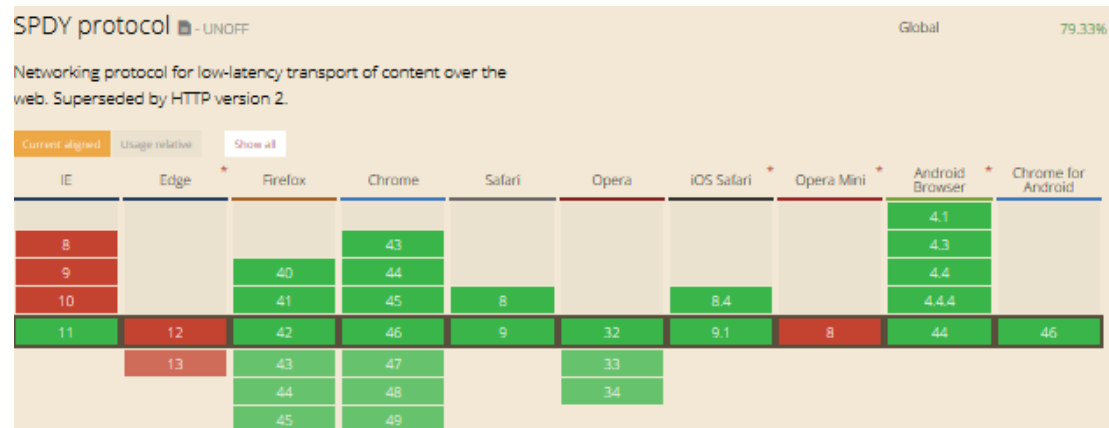
Name	Size	Time	Timeline – Start Time	400.00 ms	600.00 ms
nghttp2.org	2.6 KB	331 ms			
screen.css	8.6 KB	1 ms			
modernizr-2.0.js	3.3 KB	317 ms			
jquery.min.js	0 B	95 ms			
octopress.js	3.8 KB	315 ms			
css?family=PT+Serif:regular,italic,bold,bolditalic	0 B	87 ms			
css?family=PT+Sans:regular,italic,bold,bolditalic	0 B	87 ms			



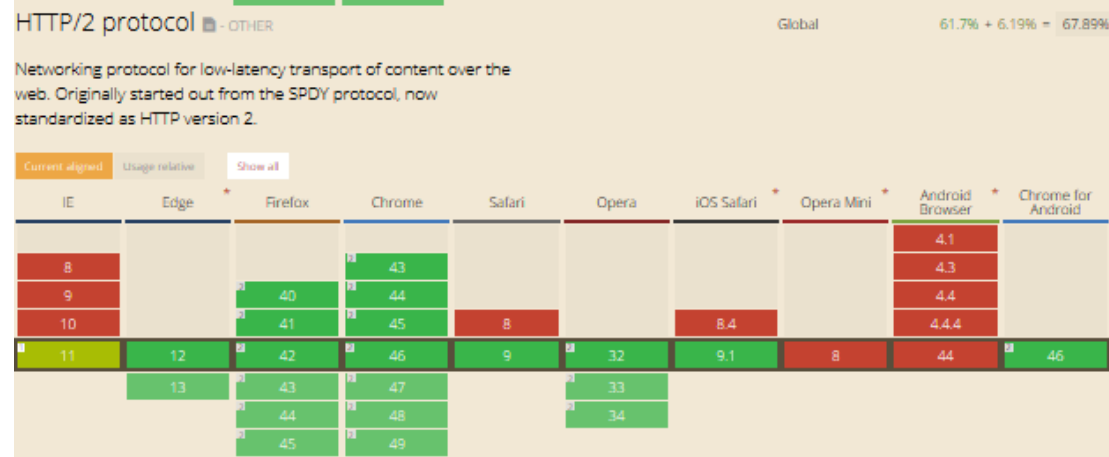
# HTTP2 - Vorteile

## Caniuse.io?

### SPDY3.1



### HTTP2



## HTTP2

### Wer nutzt es?

- Google (HTTP2, QUICK)
- Facebook (SPDY 3.1)
- Twitter (HTTP2)
- Viele der TOP-Webseiten



## HTTP2

### Wie teste ich?

<https://tools.keycdn.com/http2-test>

FF oder Chrome Plugins („SPDY Indicator“)





## HTTP2

### Weitere Ressourcen

- <https://www.chromium.org/spdy/spdy-whitepaper>
- <https://www.youtube.com/watch?v=TNBkxA313kk> (Vorstellung SPDY seitens Programmierer in 12-2011)
- <https://nghttp2.org/> (C Dokumentation für H2, weitere Infos)
- <https://ma.ttias.be/service-side-push-http2-nghttp2/> (Serverside push Test/Vorführung)
- <https://www.nginx.com/blog/http2-r7/> (Infos zur Adaptation von H2 seitens nginx)
- <https://http2.github.io/faq/> (FAQ zu H2)
- <https://http2.golang.org> (viele Beispiele zu den Funktionen)
- <http://chimera.labs.oreilly.com/books/1230000000545/index.html> („High Performance Browser Networking“)





## Kontakt

1aSEO

Christopher Wagner

Bahnhofstraße 58

51143 Köln

Telefon +49 (0)2203 8093047

Mobil +49 (0)176 63390462

chris@1aseo.de

<https://www.1aseo.de>