

TTFB

Time to First Byte

the time between clicking a link
and the first bits of content coming in.

The Push & Pull of TTFB

- Latency: The farther the user is from the end point server, the longer the round trip.
- Upfront 1-time connection delays: DNS, TLS, etc.
- Generally, static files turn around fast
- Dynamically server-generated files are slower

CDNs and Edge Providers



Is TLS fast yet?

https://istlsfastyet.com

TLS has exactly one performance problem: it is not used widely enough.

Everything else can be optimized.

Data delivered over an unencrypted channel is insecure, untrustworthy, and trivially intercepted. We owe it to our users to protect the security, privacy, and integrity of their data — all data must be encrypted while in flight and at rest. Historically, concerns over performance have been the common excuse to avoid these obligations, but today that is a false dichotomy. Let's dispel some myths.

CPU & latency costs

The process of establishing and communicating over an encrypted channel introduces additional computational costs. First, there is the asymmetric (public key) encryption used during the TLS handshake. Then, once a shared secret is established, symmetric encryption takes over.

```
# upgrade to latest
$> openssl version
OpenSSL 1.1.1a 20 Nov 2018

# run benchmarks
$> openssl speed sha
$> openssl speed ecdh
```



Good news is, modern hardware has made great improvements to help minimize these costs, and what once may have required additional hardware can now be done efficiently by the CPU.



Redirects: minimize them.

Trailing slash or no trailing slash? No prob!

```
<link rel="canonical"  
      href="https://www.scottjehl.com/lftp">
```

	 Server Rendering	"Static SSR"	SSR with (Re)hydration	CSR with Prerendering	 Full CSR
Overview:	An application where input is navigation requests and the output is HTML in response to them.	Built as a Single Page App, but all pages prerendered to static HTML as a build step, and the JS is removed .	Built as a Single Page App. The server prerenders pages, but the full app is also booted on the client.	A Single Page App, where the initial shell/skeleton is prerendered to static HTML at build time.	A Single Page App. All logic, rendering and booting is done on the client. HTML is essentially just script & style tags.
Authoring:	Entirely server-side <small>(request-response, HTML)</small>	Built as if client-side <small>(components, DOM*, fetch)</small>	Built as client-side	Client-side	Client-side
Rendering:	Dynamic HTML	Static HTML	Dynamic HTML and JS/DOM	Partial static HTML, then JS/DOM	Entirely JS/DOM
Server role:	Controls all aspects. <small>(thin client)</small>	Delivers static HTML	Renders pages <small>(navigation requests)</small>	Delivers static HTML	Delivers static HTML
Pros:	<ul style="list-style-type: none"> 👍 TTI = FCP 👍 Fully streaming 	<ul style="list-style-type: none"> 👍 Fast TTFB 👍 TTI = FCP 👍 Fully streaming 	<ul style="list-style-type: none"> 👍 Flexible 	<ul style="list-style-type: none"> 👍 Flexible 👍 Fast TTFB 	<ul style="list-style-type: none"> 👍 Flexible 👍 Fast TTFB
Cons:	<ul style="list-style-type: none"> 👎 Slow TTFB 👎 Inflexible 	<ul style="list-style-type: none"> 👎 Inflexible 👎 Leads to hydration 	<ul style="list-style-type: none"> 👎 Slow TTFB 👎 TTI >>> FCP 👎 Usually buffered 	<ul style="list-style-type: none"> 👎 TTI > FCP 👎 Limited streaming 	<ul style="list-style-type: none"> 👎 TTI >>> FCP 👎 No streaming



Server Rendering Tools





WP Super Cache

By [Automattic](#)

Download

- Details
- Reviews
- Installation
- Support
- Development

Description

This plugin generates static html files from your dynamic WordPress blog. After a html file is generated your webserver will serve that file instead of processing the comparatively heavier and more expensive WordPress PHP scripts.

The static html files will be served to the vast majority of your users:

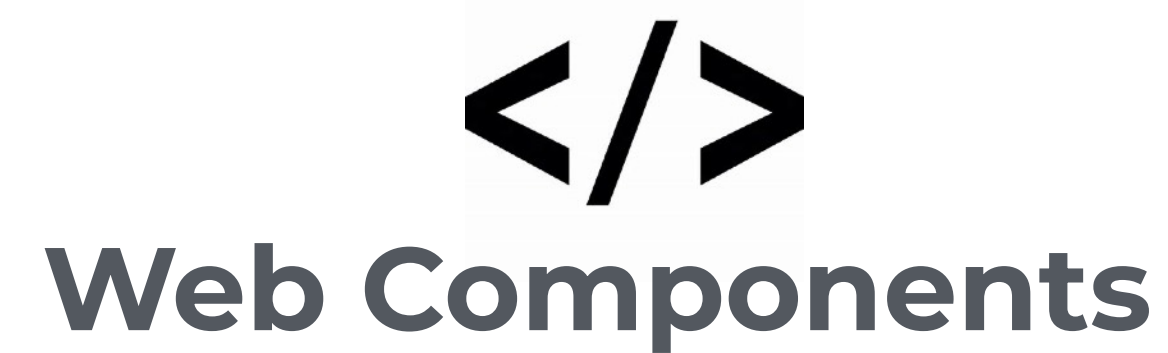
Version:	1.6.7
Last updated:	3 weeks ago
Active installations:	2+ million
WordPress Version:	3.1 or higher
Tested up to:	5.2.1



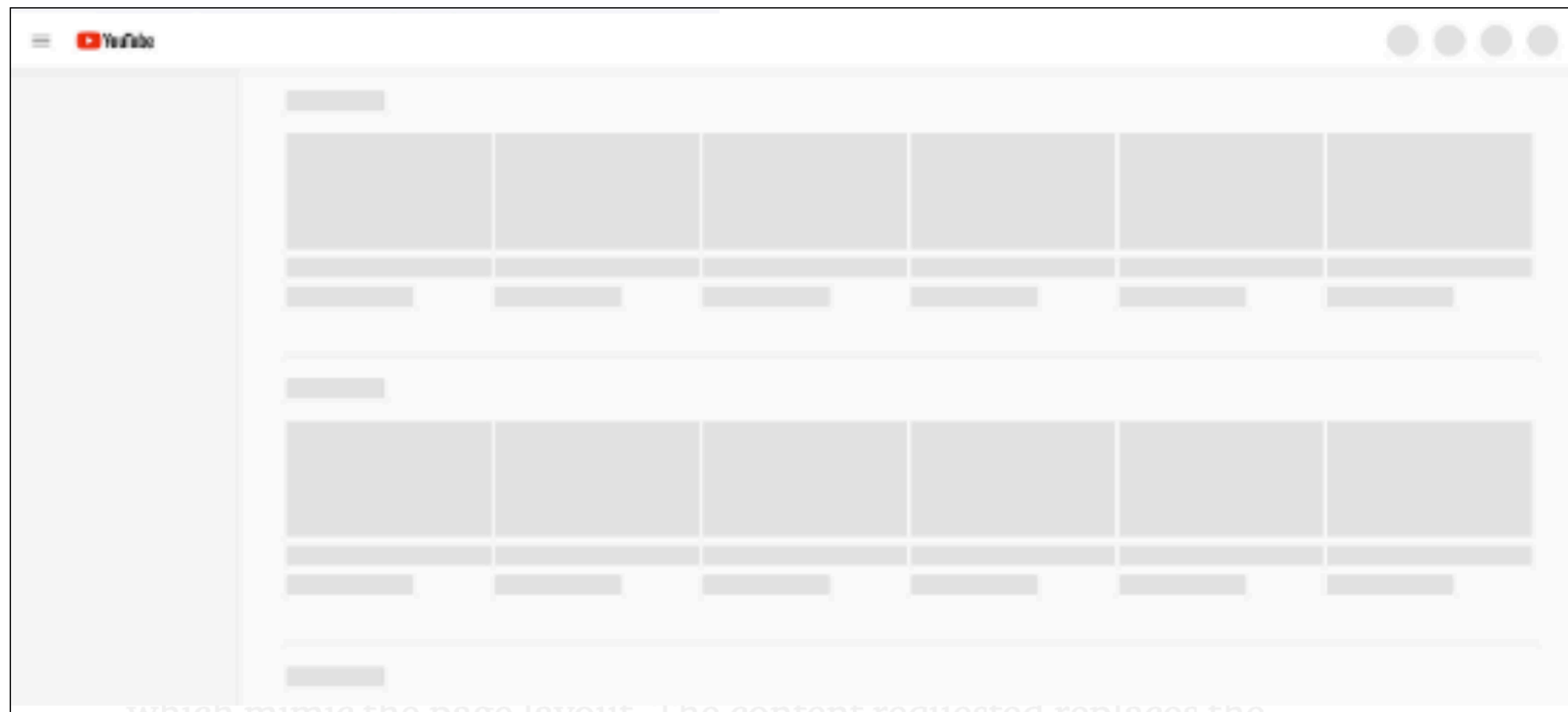
Static SSR Tools



SSR with Rehydration



Client-side with pre-rendering



Client-side Rendering



TTFB tips recapped

- Distribute files around the world on a CDN
- Optimize TLS / SSL software
- Reduce meaningless redirects
- Reduce dynamic server work time. Keep static or cached output to deliver static populated files.