

Improving Web Performance

Robert Boedigheimer
@boedie

About Me

- Microsoft MVP – Developer Technologies
- Progress Developer Expert – Fiddler
- ASPInsiders
- Pluralsight Author
- 3rd Degree Black Belt, Tae Kwon Do

- @boedie
- robert@boedie.dev
- www.boedie.dev

Performance Rules

- For most web sites, **80-90% of time is spent waiting for page resources**
 - Not optimizing (server-side) code!
- *Make fewer HTTP requests*
- Send as little as possible
- Send it as infrequently as possible

Fiddler

- Tracing tool built specifically for HTTP(S)
- Eric Lawrence (@ericlaw)
- Acquired by Telerik in 2012

- <https://www.telerik.com/fiddler> (free)

- Fiddler Everywhere (cross platform, built on .NET Core)

Metrics

- *First Byte (IT)*
- **SpeedIndex**
- Time to Interactive
- Onload
- Fully Loaded

Measure

- WebPageTest.org
- Fiddler

- Synthetic Monitoring
 - “Lab” data
- Real User Monitoring (RUM)
 - “Field” data
 - W3C Navigation Timing (DNS, connection, TTFB)
 - W3C Resource Timing

Is it Good Enough?

- Does web site performance impact **your** goals?
 - Amazon and Walmart.com
 - 100ms delay lost 1% of sales!
 - *IT and customer goals*
- How does it compare with competitors?
 - Speedcurve Benchmarks (tinyurl.com/wgvajvj)
 - Google CrUX (tinyurl.com/y9yrzeb3)
- Avoid unnecessary optimizations

HTTP Compression

- Server evaluates the “Accept-Encoding” header for request, compresses resulting response
- Gzip, Deflate, *Brotli*
- [Used 53% less bandwidth, ~25% faster synthetic measurements]
- IIS Compression
 - Enable dynamic compression

Content Expirations

- Leverage browser's cache
- My goal is returning visitor makes 1 request!
- Set expiration times for content folders
- Avoids “If-Modified-Since” and 304 response
- Change URL if need to override browser caching

Content Delivery Network (CDN)

- “Bandwidth Doesn’t Matter Much” - tinyurl.com/btqpclr
 - Latency has much larger impact for most web sites
- Global network of servers
 - Geographically closer to users
 - Offloads work from your servers
- Typically used for static files (images, CSS, JavaScript, *fonts*, etc.)

Bundling and **Minification**

- Minimize CSS, JavaScript, *possibly HTML*
 - Remove whitespace, comments, excessive semicolons, etc.
- *Bundle/combine to reduce number of HTTP requests (HTTP 1.1)*
- [Home page was **46% faster**]

Images

- Avoid if possible! (CSS 3)
- Use proper type
 - SVG
 - JPEG (photos)
- Responsive Images (srcset, <picture>)
- *Client Hints*

JPEG Images

- Compression
 - Tradeoff size vs. quality, often cut size by 50+% without any impact
- Progressive
- Image optimizers
 - Jpegtran – removes unnecessary metadata, lossless

Image Lazy Loading

- JavaScript plugin and modified HTML to request images as scroll down
 - data-src – set to image URL (or data-srcset)
 - src – set to 1x1 clear pixel image
- Pages with lots of images only load those “above the fold”
- [Github.com/aFarkas/lazysizes](https://github.com/aFarkas/lazysizes)
- *Native support*

Deferred Loading

- Use **data-deferred-src** instead of **src**
 - When onload is complete a custom script will copy URL to src
- ``
 - Uses bandwidth but immediately available when scroll
- `<iframe>` fine, some `<script>` won't work
- Great for many 3rd party beacons and resources

Loading Resources

- JavaScript (avoid blocking parsing)
 - `async` – execute when downloaded
 - `defer` – execute in source order
- Resource Hints
 - *dns-prefetch*
 - `preconnect`
 - `preload`

 - `prefetch`
 - `prerender`

Miscellaneous

- **Critical Rendering Path!**
- URLs are case sensitive (impacts caching)!
- Strict Transport Security (HSTS)
- Font subsetting

HTTP/2

- Binary framing layer
- Streams
 - Prioritization and dependencies
- **Fully multiplexed on single TCP connection**
- **Header Compression (HPACK)**
- *Server Push*

Techniques to *(probably)* Avoid (HTTP/2)

- **Bundling JavaScript and CSS files**
- CSS Sprites
- Domain Sharding
 - Using multiple host names so browser uses more connections
- Inlining
 - Data URIs, CSS (critical ok), JavaScript

Tools

- Review a web page for how well it implements various performance techniques like compression, expirations, etc
 - **Google PageSpeed** (“Lab” and “Field”)
 - tinyurl.com/m65jex6
- Chrome DevTools
 - Network throttling
 - Audits (**Google Lighthouse** – “Lab”)
- Fiddler (AutoResponder)

Resources

- Pluralsight Courses
 - Web Performance
 - A Web Developer's Guide to Images
 - **Debugging Your Website with Fiddler and Chrome DevTools**
 - Fiddler

Questions

- @boedie
- robert@boedie.dev
- www.boedie.dev

- Slides and code - <https://tinyurl.com/y9y2e6zb>