C H E C K L I S T

ATABIX FRONT-END PERFORMANCE STANDARDS

*“It’s required to be a good web citizen and even if you don’t care about users with slow connections, you do care about your ranking in Google. Your ranking in Google is at risk if you don’t optimize.” –* Patrick Sexton

**GENERAL**

* Use Gzip compression on all HTML, CSS, JS & SVG files (.htaccess)
* Browser-caching headers (.htaccess)
* JS files combined and minified
* CSS files combined and minified
* Load CSS before JavaScript
* Removal of all comments in (S)CSS en HTML

**HTML5**

* Pre-fetch assets
* Specify a character set (UTF-8)
* Remove all comments from HTML

**(S)CSS**

* Remove inline style blocks and use <link> CSS files in the <head>
* Remove unused CSS
* Remove unused SASS imports from foundation
* Remove all comments from SCSS & CSS files
* Efficient usage of CSS selectors
* Avoid CSS @import
* CSS blessed file(s) between IE9 LTE conditional tags

**JAVASCRIPT**

* Load 3rd party assets async.
* Functions.js
* Minified JavaScript (functions & libraries)
* JS libraries included on designated pages
* Latest Foundation library gzipped and served from Atabix CDN
* jQuery & jQuery UI served from CDN
* Postpone loading JavaScript which isn’t executed onload
* Postpone parsing JavaScript until absolutely necessary
* Intelligent script loader for parallel async processing, like requireJS  
  (see: <http://requirejs.org/> )
* Modernizr.js minified and out of the head
* HTML5 shiv placed in the head between IE9 LTE conditional tags
* TypeKit JS, embed code in the head instead of 2 JS file imports  
  (see: http://help.typekit.com/customer/portal/articles/649336-embed-code)

**IMAGES / SVG**

* Combine images using sprites
* All images and sprites optimized and compressed (see: kraken.io)
* Serve scaled images
* Use data URL’s for small images on HTTPS connections

(see: <http://dataurl.net/#about> )

* Use vector images when possible
* Optimize SVG with an SVG cleaning tool  
  (see: <https://github.com/svg/svgo> )

**URL’s**

* Removal of all broken links, missing images and any other bad requests.
* Resources served from multiple hostnames for parallel processing
* Minimize redirects

**CACHING & COMPRESSION**

* Use a Content Delivery Network (CDN)
* Use gzip compression on all HTML, CSS, JS & SVG files (.htaccess)
* Browser-caching headers (.htaccess)
* Make redirects cacheable
* Enable public caching in the HTTP header of static assets like images en JavaScript files (also on Amazon uploads)
* Use Google Library API’s

**GOALS**

* Server response-time under 0.8sec
* Single pages smaller than 500kb
* Google Pagespeed between 80 en 100
* **Guidelines:** <https://varvy.com/>