**Matt Eastman 5.5.17**

Amp = ACCELERATED Mobile Pages

# Amp in a nut shell.

Amp is an open source framework by google, that aims to speed up load speed by offering a CDN service and implementing a lightweight (and strict) library of html.

Essentially a framework for creating mobile web pages, AMP consists of three basic parts:

1. **AMP HTML:** A subset of HTML, this markup language has some custom tags and properties and many restrictions. But if you are familiar with regular HTML, you should not have difficulty adapting existing pages to AMP HTML.
2. **AMP JS:** A JavaScript framework for mobile pages. For the most part, it manages resource handling and asynchronous loading. It should be noted that third-party JavaScript is not permitted with AMP.
3. **AMP CDN:** An optional Content Delivery Network, it will take your AMP-enabled pages, cache them and automatically make some performance optimizations.

First business to adopt amp (but now all the cool kids use it)

Twitter, Pinterest, WordPress.com, Chartbeat, Parse.ly, Adobe Analytics LinkedIn

# Amp Requirements / Rules

1. Amp pages are cached and served from google servers
   * Side bar – Historical change for google because historically google has just indexed sites / info. With its AMP search results, Google is amassing content on its own servers and keeping readers on Google.
2. Most JavaScript is forbidden, third party JavaScript is not allowed.
   * There is a hack using I frames. See JavaScript Iframe loophole section below.
3. Must contain Amp Components in right order
   * Start with the doctype <!doctype html>.
   * Contain a top-level <html ⚡> tag (<html amp> is accepted as well).
   * Contain <head> and <body> tags (They are optional in HTML).
   * Contain a <meta charset="utf-8"> tag as the first child of their <head> tag.
   * Contain a <script async src="https://cdn.ampproject.org/v0.js"></script> tag as the second child of their <head> tag (this includes and loads the AMP JS library).
   * Contain a <link rel="canonical" href="$SOME\_URL" /> tag inside their <head> tag that points to the regular HTML version of the AMP HTML document or to itself if no such HTML version exists.
   * Contain a <meta name="viewport" content="width=device-width,minimum-scale=1"> tag inside their <head> tag. It's also recommended to include initial-scale=1.
   * Contain the following in their <head> tag: <style amp-boilerplate>body{-webkit-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-moz-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start 8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s steps(1,end) 0s 1 normal both}@-webkit-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-moz-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-ms-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-o-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}</style><noscript><style amp-boilerplate>body{-webkit-animation:none;-moz-animation:none;-ms-animation:none;animation:none}</style></noscript>
   * Component notes
     + Amp in <html>
     + Async in script tags to load asyncrasisususuly
     + Amp Analytics tag - is not Not required, but if used must be in Head must be before JavaScript library
     + schema – highly recommended. You will need schema to be perfectly valid AMP
4. CSS is inline. No external CSS files. Also, Must be less than 50KB.
5. html comments not allow <!—“Sorry dude” –ed’s voice -->
6. <form> was previously not allowed but now is.
   * Forms are [now supported](https://amphtml.wordpress.com/2016/10/31/forms-now-supported-in-amp/amp/) using the amp-form extension. Like other AMP extensions, you will need to include another JavaScript file in the head of your AMP page: <script async custom-element="amp-form" src="https://cdn.ampproject.org/v0/amp-form-0.1.js"></script>.
7. Open Graph protocol tools work – Facebook like buttons
8. The use of script tag is not allowed on the page unless for application/ID+json

# JavaScript Loop hole (for third party and non-amp friendly)-

This loophole allows publishers to include ads, analytics, and other pieces of JavaScript on a page.

To do this, you need to

1. add the amp-iframe extended component to your <head>.
   1. <script async custom-element="amp-iframe" src="https://cdn.ampproject.org/v0/amp-iframe-0.1.js"></script>
2. add an amp-iframe tag in your body.
   1. It must be at least 600px or 75% of the first viewport away from the top. The URL in the iframe has to be an https url and must not be on the same domain (or at least subdomain).

Amp documentation - <https://www.ampproject.org/docs/reference/components/amp-form>

**AMP Playground -** [**https://ampbyexample.com/components/amp-form/**](https://ampbyexample.com/components/amp-form/)

Google iframe video - <https://www.youtube.com/watch?v=1z7qjukh_3I>

Forms now supported - <https://amphtml.wordpress.com/2016/10/31/forms-now-supported-in-amp/amp/>

Old Documentation - <https://searchwilderness.com/amp-forms-hack/>

# Amp analytics, Pixel, and tech integration

* AMP analytics allows you to send events back to your server based on triggers configured by you. Use  [analytics integration guide](https://github.com/ampproject/amphtml/blob/master/extensions/amp-analytics/integrating-analytics.md) to get you started. If you simply need to add a tracking pixel with dynamic parameters to your tracking URL, check out [amp-pixel](https://www.ampproject.org/docs/reference/components/amp-pixel). There are 20+ analytics providers who have added support to amp-analytics. Here is a [sample pull request](https://github.com/ampproject/amphtml/pull/1595) from the analytics provider [Parse.ly](https://www.parsely.com/help/integration/google-amp/).

**there are 5 ways to intergrate technology with AMP**

1. Add your support to the amp-analytics extension
2. Using an amp-ad extension
3. Use the amp-call-tracking extension
4. Adding a new extension/embed
5. Using amp-iframe

<https://www.ampproject.org/docs/contribute/integrate-your-tech/>

API analytics- <https://github.com/ampproject/amphtml/blob/master/extensions/amp-analytics/integrating-analytics.md>

API GIT - <https://github.com/ampproject/amphtml/tree/master/ads#available-apis>

# Catch22’s – things to look at

1. Amp pages point back to google.com instead of Teleflora.com
   1. Ex. google.com/amp/teleflora.com/amp
2. Virtually no Javascript is allowed on AMP, including google analytics tracking code and Google Tag Manager code.
   1. Amp pages can tolerate GTM tracking codes, it won’t crash. However, it will not be valid AMP and not show as AMP to mobile uses.
   2. Amp has pre-screened analytics providers such as Chartbeat, Adobe, and Parse.ly.. and maybe more that we can look into.
3. Payment page – could pose issues. We could use an amp feature amp-access-laterpay or we could just link to payment page.
   1. Later pay --The amp-access-laterpay component does not require an authorization or pingback configuration, because it is pre-configured to work with the LaterPay service. It also does not require manual setup of login links.The different purchase options can be configured on the publisher's LaterPay account, and the component will retrieve the configuration and create a list of available purchase options.You can refer to the documentation on configuring the [LaterPay Connector](http://docs.laterpay.net/connector/configuring/), LaterPay's existing front-end integration, to learn how to configure the purchase options.
      1. <https://www.ampproject.org/docs/reference/components/amp-access-laterpay>
4. Amp and Schema combo is recommended
5. With limited css styles and JavaScript we will likely need to completely rebuild all amp pages

# Amp Example: How images work

AMP requires all elements to have an explicit size set from the get-go.

<**amp-img**

src="/img/narrow.jpg"

srcset="/img/wide.jpg 640w,

/img/narrow.jpg 320w"

width="1698"

height="2911"

layout="responsive"

alt="an image">

</**amp-img**>

This amp-img element automatically fits the width of its container element, and its height is automatically set to the aspect ratio determined by the given width and height. Try it out by resizing this browser window

The layout attribute gives you easy, per-element control over how your element should render on screen. Many of these things are possible with pure CSS – but they're much harder, and require a myriad of hacks. Use the layout attribute instead.

<https://www.ampproject.org/docs/guides/responsive/control_layout>

### **\*\*\*Can use Media queries**

<**amp-img**

media="(min-width: 650px)"

src="wide.jpg"

width=466

height=355

layout="responsive">

</**amp-img**>

# Amp next steps

1. ~~Research Amp language / limitations / issues~~
2. ~~Scrape current teleflora website~~
3. ~~Look through current website and list possible issues~~
   1. Consult with web producers
4. Look through “AMP by example” and pick out features to implement
   1. Consult with web producers
5. Go to Google headquarters and get help.
6. Create UI PSD Designs of Amp’d mobile version of site
   1. Get designs approved
7. Create amp web page versions
   1. Amp on a few our hour pages
   2. Home Page
   3. Category page
   4. PDP Page
   5. Landing page - Gift guide
8. Analyze technology pros and cons verse current mobile site
   1. Conduct user testing
9. Create implementation system with OKC
10. Do the damn thing
    1. Validate amp using - <https://validator.ampproject.org/>

# Sources

Basic overview

<https://moz.com/blog/accelerated-mobile-pages-whiteboard-friday> (Video)

<https://www.wired.com/2016/02/googles-amp-speeding-web-changing-works/>

<https://googleblog.blogspot.com/2015/10/introducing-accelerated-mobile-pages.html> (google)

<http://searchengineland.com/get-started-accelerated-mobile-pages-amp-240688>

AMP PAGE – doc. Examples.

<https://www.ampproject.org/> (main)

<https://ampbyexample.com/>

<https://github.com/ampproject>

Iframe hack

Amp documentation - <https://www.ampproject.org/docs/reference/components/amp-form>

Playground - <https://ampbyexample.com/components/amp-form/>

Google iframe video - <https://www.youtube.com/watch?v=1z7qjukh_3I>

Forms now supported - <https://amphtml.wordpress.com/2016/10/31/forms-now-supported-in-amp/amp/>

Old Documentation - <https://searchwilderness.com/amp-forms-hack/>

<https://developers.google.com/analytics/devguides/collection/amp-analytics/> (analytics)

<https://www.optimizesmart.com/setting-tracking-amp-pages-google-analytics/> (Components, schema, ad tracking, Rules)

# Notes

**Google AMP Call Notes 4.26**

Emily and James Gtech solutions engineer - point of contact. Andrew and Brandon

Mobile performance specialist -- fire based

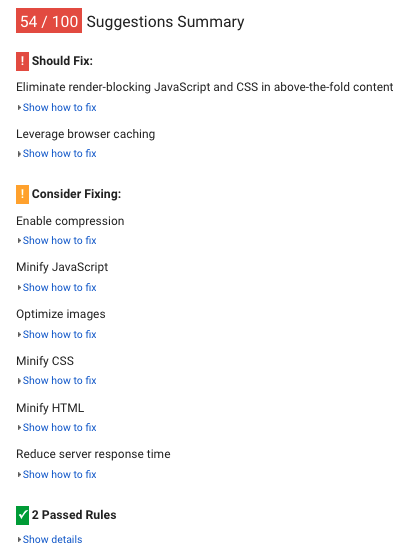
Checkout

- Chrome 59 - Chrome canneri - profiling tools - code coverage tool - % of coverage javacript file that is covered

- Json LD specific shemea markup to help with search optimization

# **Question for team and google–**

* In regards to being “AMP verified” Will we be able to iframe hack all, some, or none of our third party JavaScript?
  + List all we should get rid of, maybe keep and keep.
* Pros and cons of using amp framework but not being amp verified?
  + Is this something we see valuable?
* Further clarification on why we need to specific all images/elements and how it works responsive.
* What are the pre-screened analytic tools AMP supports? Are any of our anaylitic tools prescreened?

****