OUTBRAIN JAVASCRIPT IMPLEMENTATION GUIDE

OUTLINE

I. **Initial Set Up:** What your account strategist needs before testing.

II. **The Outbrain JS:** What our code looks like and how to install it.

III. **Indexing URLs & Learning About Content:** How we index your site and learn about your content.

IV. **Testing and Styling:** How we validate the code and implement styling.

V. **Launch**

I. Initial Set Up

Here is a list of items your account strategist will need in order to prepare your site for launch:

- **Where do you plan to install Outbrain?**
  Outbrain supports multiple page types such as articles, blogs, slideshows, and videos.

- **What areas of your site would you like to recommend in the Outbrain widget?**
  Outbrain can set up content whitelists if you only want to recommend specific sections of your site.

- **What do you want the Outbrain widget to look like?**
  Our widget is a humble guest on your page and will inherit CSS to remain consistent with your site design. If you have something specific in mind, let us know. You name it, we can probably do it.

- **Quality image file of logo.**
  Send us a high-quality file of your site's logo and we'll attach it to any article that doesn't have a thumbnail. That way great, image-free content gets the chance to shine in image-based widgets.

II. The Outbrain JS

You can start working with the code as soon as the initial set up is complete. We take the quality of our products very seriously. Every pixel and line of code we download to your page has an essential function. Here are a few facts that illustrate our commitment:

- The Outbrain widget is less than 25KB (8 KB zipped).

- When serving thumbnails, we only serve images in the exact size of the thumbnail, which means our total impact on page size and load time is minimal.

- Our widget loads asynchronously, and therefore will never delay page loading. In the rare cases we have issues with our recommendation server, the widget will simply not load. (This rarely happens thanks to our backend redundancy.)

Implementing the code involves two simple steps:

1. **The Code:** Place the code your account strategist provides in the area of your site where you would like our widget to appear. Please note that your setup may require multiple pieces of code.

   Here is a typical example of our code:

   ```
   <div class="OUTBRAIN" data-src="DROP_PERMALINK_HERE" data-widget-
   id="WIDGET_ID" data-ob-template="TEMPLATE" ></div>
   <script type="text/javascript" async="async" src="http://widgets.outbrain.
   com/outbrain.js"></script>
   ```

   * This code is not for use on your site. Your account strategist will provide you with the proper TEMPLATE and WIDGET_ID values.

2. **The Permalink:** Pass one dynamic parameter into the data-src variable (highlighted above). This value should be the URL of the page on which the widget is loading.

   Option A
   We recommend using the canonical or og:url of the page, as long as one of these is consistently used across the site. A couple of reasons why canonical and og:urls are great:

   - Outbrain won't crawl URLs with extraneous parameters like marketing codes, sessionIDs, referrer strings, etc. Crawling URLs with any of these parameters causes duplication in our system and prevents us from being able to

serve the best recommendations to your readers.

✅ http://m.nypost.com/f/mobile/news/local/newark_tsa_bomb_boozled_eTIZBp2X7B299qO5WCWvAK

❌ http://m.nypost.com/;s=C1RQpzi9IBXYhst_quWLp36/f/mobile/news/local/newark_tsa_bomb_boozled_eTIZBp2X7B299qO5WCWvAK

- Outbrain will only crawl the first page of a multi-page story, which also prevents us from recommending any of the secondary pages in your widget. This is particularly relevant for slideshows.

  ✅ http://www.foxnews.com/slideshow/entertainment/2009/10/06/celebrities-technology-mix

  ❌ http://www.foxnews.com/slideshow/entertainment/2009/10/06/celebrities-technology-mix/#slide=2

Option B
If you don't use canonical or og:urls and cannot implement them, the href location is your next best option.

III. Indexing URLs & Learning About Content

We use the data-src value for two purposes. The first is simply to index the URLs from your site. Outbrain crawls a URL the first time our JavaScript loads on the page. The second is to learn about content on that page.

1. **Indexing URLs:** Outbrain needs to index your site before launch to ensure the best possible performance on day one. Indexing in advance also gives us recommendations to display during the testing process. Here are the ways we can crawl your site before we're live:

   ### Option A
   RSS feeds. If you have a list of RSS feeds, please send it to your account strategist.

   ### Option B
   No feeds? No problem. We can also use a sitemap or list of URLs.

   ### Option C– Advanced Dev Ninja Option
   We can make our JS results invisible. This means that you could roll the code out to production before the testing phase is complete. In this scenario, we would hide the widget on our end and just use the code for the purpose of indexing URLs. This also allows us to get a lot of the data needed to better inform our recommendations. Without this data, the recommendations may not be optimal for the first day or two in production.
   * See this in action here: http://widgets.outbrain.com/external/demos/draft-demo.html

   ### Option D
   If none of these options are feasible, we will crawl your site whenever the code launches in production. In this scenario, the widget may take a few hours to fully populate with recommendations.

2. **Learning about content:** Outbrain will also crawl information about the URLs we are indexing in order to serve more interesting and relevant recommendations to your readers.

Here's a list of what we look for:

- Title
- Description
- Publish Date
- Image
- Author
- Content

The best way for us to gather this information is via microdata, like Google's Schema and Facebook's Open-graph, or via Outbrain proprietary tags. We highly recommend deploying one of these options across your site and have provided tag examples for each of the above attributes in Appendix A.

IV. Testing and Styling

Now that we have indexed your content and know where to pick up important information, we can begin testing. This process is for validating code and styling the widget(s). Please make sure that the placements and page types in testing mirror the eventual experience in production. Also note that the recommendation results in testing do not reflect what will come through in production since we will not have all possible behavioral data yet.

If you have a test environment ...
You will need to do the following in order for the Outbrain widgets to work:

- **Whitelist our crawler IP addresses and user agent (your Account Strategist will provide both sets of IPs if needed).** If your test environment is behind a firewall and our crawlers don't have access, the widget won't work. Please contact your account strategist for the most up to date crawler IP information.

- **Grant your account strategist access to the test environment.** The fastest and most reliable way for Outbrain to validate the code and style the widgets is to do so directly in testing. If needed, your account strategist can provide you with our office IP addresses.

- **Pass production permalinks in testing.** It's ideal if you can pass production URLs into the data-src variable in testing, either dynamically or hard-coded. This prevents us from indexing your test domains. This step is not a requirement for the widgets to work, but streamlines the process and ensures a faster styling turnaround.

If not, don't worry...
Outbrain can also work without a proper dev site:

- We can hide the recommendation results so that you can launch the code directly in production. Outbrain has our own shareable draft environment where the results will be visible. We can use this environment to finalize styling and guarantee everything is functioning correctly.
    * See this in action here: http://widgets.outbrain.com/external/demos/draft-demo.html
- You can launch the code visibly on one low-traffic production page.

Once your account strategist has a working test page...
She/he can begin implementing the UI upon which your teams agreed in section I!

V. Launch

Once Outbrain has validated everything in testing, you can push our JS to production! Just let your account strategist know the expected date and time of launch. If you encounter any issues post-launch, please contact your account strategist.

APPENDIX A: SUPPORTED MICRODATA

Necessary Information

Attribute	Suggested Microdata	Example

Title	vr:title
	OR
	og:title
	OR
	itemprop=title
	OR
	itemprop=headline
	OR
	name=title
		`<meta property="vr:title" content="4 Tips To Marketing Your Legacy" />`
		
		`<meta property="og:title" content="4 Tips To Marketing Your Legacy" />`
		
		`<meta itemprop="title" content="4 Tips To Marketing Your Legacy" />`
		
		`<meta itemprop="headline" content="4 Tips To Marketing Your Legacy" />`
		
		`<meta name="title" content="4 Tips To Marketing Your Legacy" />`

Publish Date	vr:published_time
	OR
	itemprop=datePublished
	OR
	name=pubdate
		`<meta property="vr:published_time" content="2013-03-13T06:07:39Z" />`
		
		`<meta itemprop="datePublished" content="2013-03-13T06:07:39Z" />`
		
		`<meta content="2013-03-13T06:07:39Z" name="pubdate" />`

Image	vr:image
	OR
	og:image
	OR
	link rel=image_src
	OR
	itemprop=meta-thumbnailUrl
		`<meta property="vr:image" content="http://wp.outbrain.com/wp-content/themes/outbrain/images/logo.png" />`
		`<meta property="og:image" content="http://wp.outbrain.com/wp-content/themes/outbrain/images/logo.png" />`
		
		`<link rel="image_src" href="http://wp.outbrain.com/wp-content/themes/outbrain/images/logo.png" />`
		
		`<meta itemprop="meta-thumbnailUrl" content="http://wp.outbrain.com/wp-content/themes/outbrain/images/logo.png" />`

Content	itemprop='body'
	OR
	itemprop=articleBody
		`<div itemprop="body"> This is the content, contains HTML </section>`
		
		`<section itemprop="articleBody"> This is the content, contains HTML </section>`

Description	og:description **OR** itemprop=description	`<meta content="When it comes to celebrating your brand's legacy, getting consumers to join in can be a tricky proposition. Here are four tips to doing it well." itemprop="description" property="og:description"/>`
Author	name=author **OR** itemprop=author	`<meta content="Brandon Carter" itemprop="author" name="author"/>`

Optional Information (Improves Crawling Accuracy)

Attribute	Suggested Microdata	Example
Language	og:locale **OR** itemprop=inLanguage	`<meta content="en-US" property="og:locale" itemprop="inLanguage"/>`
Keywords	name=keywords **OR** itemprop=keywords	`<meta content="" itemprop="keywords" name="keywords"/>`
Content Type	og:type	`<meta content="article" property="og:type"/>`

Property	og:site_name
	OR
	itemprop=sourceOrganization
		» Ex1:
		`<meta property="og:site_name" content="E! Online" />`
		» Ex2:
		`<meta content="CNN" itemprop="sourceOrganization"/>`

Section	name=section **OR** itemprop=articleSection	`<meta content="health" itemprop="articleSection" name="section"/>`