Live Blog SEO - technical documentation

This feature requires Live Blog version 3.3 or higher and knowledge on how to inject content into your article pages (for example based on Edge Side Include technology)

In order to enable both the generation of static content and the updating of a blog on a client without requiring a refresh, the new default theme behaves like an isomorphic webapp, a term coined by AirBnB. To elaborate, isomorphism is the functional aspect of seamlessly switching between client- and server-side rendering without losing state. In our case, we're making use of a templating language that is available both on the server and the client: Jinja2, an integral part of the well-known web framework Django, already in use elsewhere in the Liveblog stack – and its client-counterpart Nunjucks, courtesy of Mozilla.

After the initial load, subsequent renders are done by requesting the API via XHR and then rendering the response to the same templates used by the server. This way, we're making a step towards separation of concerns: all static resources live in the theme. We're feeding unmodified API responses to our templates and expect them to contain everything we need. In this fashion we're avoiding a common pitfall of template isomorphism: having to maintain filters with exactly the same functionality in multiple languages.

How extendability works

With the Liveblog 3 default theme we're leveraging the concept of templating. Our default templates are baked into a single js bundle and referenced in code through global variables. The template logic contains a switch that decides whether to use the prebaked templates or override them based on global variables referencing individual templates/*.nunj files. As we're replacing one at a time, baked-in and custom templates can be mixed.

We reference our pre-built default Liveblog library file in all the inheriting themes and circumvent the problem of having clients serve stale versions of the library. The pre-built and minified library lives at https://platform.sourcefabric.com/theme.js. As for styles, we're leveraging the cascading part of cascading stylesheets, effectively overriding classes in our topmost css bundle.

Basic theme customization

Using our default theme, open up either the global or per-blog theme settings dialog. Here you will find basic style and layout options that can easily be manipulated through the interface.

Expert theme customization

This describes the process of inheriting from the Liveblog 3 default theme and changing both a bit of template code and the corresponding style information. In this example, we're moving post timestamps to the end of each post and replace the default color branding. The bare minimum you will need is a directory structure as follows: theme.json, templates/index.html, templates/post.html, css/globals.css. Just clone our starter kit on Github.

Let's open our index.html. We won't touch anything here, but you will see the default library platform.sourcefabric.com/theme.js referenced before the closing body tag as well as our empty less/globals.less referenced in the head section after the precompiled style file that we're cascading down from. Next, open the less/globals.less and edit the @secondary_color value to taste, for demonstration purposes we're using „hotpink" of course. That's it for the styles. Now let's push that timestamp around. Open up the templates/post.html and move the timestamp element from line 12 to line 27. Boom, done.

Before shipping your theme, you should have a) a quick look at the theme.json and edit it to taste. Should be self-explanatory, as it's conceptually based off of npm's `package.json`. Also: This step is entirely optional. Now, we need tob) package our theme.

To do that, you first need to install a few dependencies via `npm i` inside your theme directory. Then, to fire up the actual build process, run `make`. The result is a your-name-here.zip placed in your working directory that's ready to be uploaded to your Liveblog instance.

The Live Blog theme generator

We introduced a theme generator in order to make sure everything is in place when extending a theme. Please check it out on Github. The Live Blog theme generator will help you to easily create and maintain theme extensions to our default theme: https://github.com/liveblog/generator-liveblog-theme

Handle the SEO friendly Live Blog output with ESI technology

In order to make your content visible to search engines you will have to set up a technical infratsructure inside of your CMS that helps you retrieving the static Live Blog output and attach it as native content inside your articles. Here is how you can handle to integrate the SEO friendly output of Live Blog into the Superdesk Publisher for example:

Read more about ESI and Varnish here: https://varnish-cache.org/docs/4.0/users-guide/esi.html

General idea about ESI on example: https://www.fastly.com/blog/using-esi-part-1-simple-edge-side-include/

Edge-Side Include (ESI) is a web standard originally proposed by Akamai and Oracle, among other companies. It allows an Edge Server to "mix and match" content from multiple URLs.