

How to Implement your Design System in Liferay

Evan Thibodeau

Team Lead, Liferay

It Depends...

Agenda

In order to understand how to implement your design system in Liferay we will discuss the following:

01

New Liferay Features for
your Implementation

02

How your Business
Requirements Affect
your Implementation

03

Example
Implementations
(Demo)

01

New Liferay Features for your Implementation

01 Liferay Features

Theme CSS Client Extension

Allows you to replace the CSS and the style book tokens provided by the theme.

Replaces

- Clay.css, Main.css, and frontend-token-definition.json from the theme

Documentation

- [Using a Theme CSS Client Extension](#)
- [Clay Sass API](#)

Examples

- [Liferay-sample-theme-css-3](#)
- [Dialect-theme-css](#)
- [noClay-theme-css-client-extension](#)

```
↳ liferay-sample-theme-css-3 git:(master) cat client-extension.yaml
assemble:
  - from: build/buildTheme/img
    into: static/img
assemble:
  - from: build/buildTheme/images
    into: static/images
liferay-sample-theme-css-3:
  clayRTLURL: css/clay rtl.css
  clayURL: css/clay.css
  frontendTokenDefinitionJSON: src/frontend-token-definition.json
  mainRTLURL: css/main rtl.css
  mainURL: css/main.css
  name: Liferay Sample Theme CSS 3
  type: themeCSS%
```

```
↳ liferay-sample-theme-css-3 git:(master) tree .
.
└── client-extension.yaml
    └── src
        └── css
            └── _clay_variables.scss
                └── frontend-token-definition.json
```

3 directories, 3 files

01 Liferay Features

CSS Client Extension

Allows you to add additional CSS on top of the CSS provided by the theme.

Replaces

- Theme Contributor CSS, or CSS added directly through a configuration in the UI

Documentation

- [Using a CSS Client Extension](#)

Examples

- [liferay-sample-global-css](#)

```
↳ liferay-sample-global-css git:(master) cat client-extension.yaml
assemble:
  - include: static/*.css
liferay-sample-global-css:
  name: Liferay Sample Global CSS
  type: globalCSS
  url: global.*.css%
↳ liferay-sample-global-css git:(master) tree .
.
└── assets
    └── global.css
    ├── client-extension.yaml
    ├── package.json
    └── webpack.config.js
2 directories, 4 files
```

01 Liferay Features

JavaScript Client Extension

Allows you to add a script tag to any page in Liferay. Can be applied broadly to different scopes, such as page, page template, site, or instance.

Replaces

- JS that would be provided by a theme, theme contributor, or added directly through a configuration in the UI

Documentation

- [Using a JavaScript Client Extension](#)

Examples

- [Liferay-sample-global-js-2](#)
- [gov-uk-global-js](#)

```
↳ liferay-sample-global-js-2 git:(master) cat client-extension.yaml
assemble:
  - from: build/static
    into: static
liferay-sample-global-js-2:
  name: Liferay Sample Global JS 2
  scriptElementAttributes:
    async: true
    data-attribute: value
    data-senna-track: permanent
    fetchpriority: low
  type: globalJS
  url: global.*.js%
```



```
↳ liferay-sample-global-js-2 git:(master) tree .
.
└── assets
    └── global.js
    ├── client-extension.yaml
    ├── package.json
    └── webpack.config.js
```

2 directories, 4 files

01 Liferay Features

JS Import Maps Entry Client

Extension

A module to add custom JS modules to Liferay's JS import map for easy sharing of modules across all of Liferay.

Replaces From Theme

- New Feature!

Documentation

- [Bundling Resources in a JavaScript Import Map Entry Client Extension](#)
- [Mozilla Import Map Docs](#)

Examples

- [Liferay-sample-js-import-maps-entry](#)
- [gov-uk-js-import-maps-entry](#)

```
↳ liferay-sample-js-import-maps-entry git:(master) cat client-extension.yaml
assemble:
  - from: build/static
    into: static
liferay-sample-js-import-maps-entry:
  bareSpecifier: jquery
  name: Liferay Sample JS Import Maps Entry
  type: jsImportMapsEntry
  url: jquery.*.js
```

```
↳ liferay-sample-js-import-maps-entry git:(master) tree .
.
└── assets
    └── index.js
└── client-extension.yaml
└── package.json
└── webpack.config.js
```

2 directories, 4 files

01 Liferay Features

Theme Sprite Map Client Extension

Allows you to override the default Clay sprite map used for icons on a page.

Replaces From Theme

- /images/clay/spritemap.svg

Documentation

- [Using a Theme Sprite Map Client Extension](#)

Examples

- [liferay-sample-theme-spritemap-2](#)

```
↳ liferay-sample-theme-spritemap-2 git:(master) cat client-extension.yaml
assemble:
  - from: build/static
    hashify: spritemap.svg
    into: static
liferay-sample-theme-spritemap-2:
  name: Liferay Sample Theme Spritemap 2
  type: themeSpritemap
  url: spritemap.*.svg

↳ liferay-sample-theme-spritemap-2 git:(master) tree .
.
└── buildSpritemap.js
└── client-extension.yaml
└── package.json
└── src
   └── cog.svg
   └── foo.svg

2 directories, 5 files
```

01 Liferay Features

Theme Favicon Client Extension

Allows you to override the theme's favicon on the selected page.

Replaces From Theme

- /images/favicon.ico

Documentation

- [Using a Theme Favicon Client Extension](#)

Examples

- [liferay-sample-theme-favicon](#)

```
↳ liferay-sample-theme-favicon git:(master) cat client-extension.yaml
assemble:
  - from: assets
    hashify: favicon.ico
    into: static
liferay-sample-theme-favicon:
  name: Liferay Sample Theme Favicon
  type: themeFavicon
  url: favicon.*.ico

↳ liferay-sample-theme-favicon git:(master) tree .
.
└── assets
    └── favicon.ico
    └── client-extension.yaml

2 directories, 2 files
```

01 Liferay Features

Master Pages

Master page templates provide a way to define elements common to every page using fragments (e.g. headers and footers previously provided by themes).

Replaces From Theme

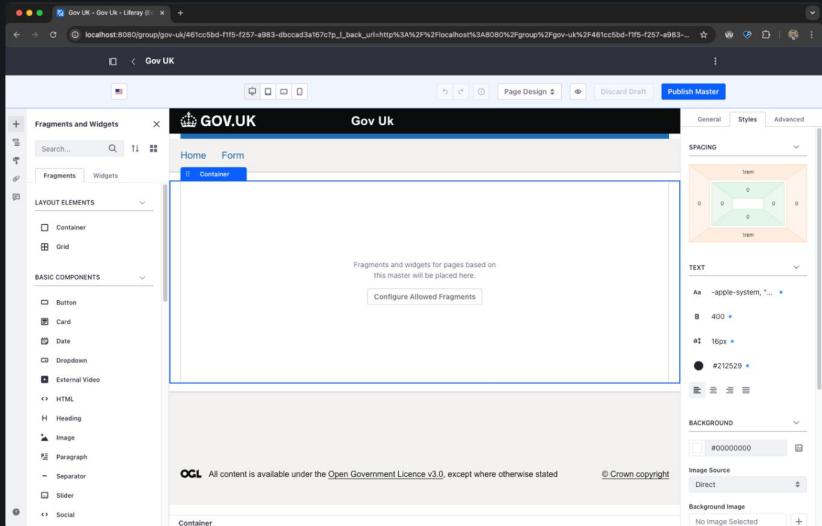
- Theme Template (.ftl) Files, Theme Settings

Documentation

- [Master Page Templates](#)
- [Using Content Pages](#)

Examples

- [gov-uk-master-page](#)



01 Liferay Features

Page Fragments

Page fragments are extensible, configurable, and reusable drag-and-drop elements for building content pages.

Replaces From Theme

- Theme Template (.ftl) Files, Theme Settings

Documentation

- [Using Fragments](#)
- [Developing Page Fragments](#)
- [Fragment Configuration Types Reference](#)

Examples

- <https://github.com/lfrsales/demo-fragment-collections/>
- govuk-fragments

```
HTML


<nav class="navbar navbar-collapse-relative navbar-expand-md navbar-light">
<button aria-active-descendant aria-controls="navigationBarCollapse" type="button">
<span class="navabar-text-truncate">Tab 1</span>
<span class="icon-caret-bottom"></span>
</button>
<div class="collapse navbar-collapse" id="navigationBarCollapse">
<ul class="nav navbar-nav" role="listbox">
#list 0..configuration.numberOfTabs-1 as i1
<li class="nav-item" role="presentation">
<button aria-controls="tabPanel${i1}-$fragmentEntryLinkId" type="button">
<span class="navabar-text-truncate" data-lfr-editable-id="Tab ${i1}">
</span>
</button>


```

```
CSS
.component-tabs .navigation-bar-light .navbar-toggler {
color: #272833;
}
.component-tabs .navigation-bar-light .icon-caret-bottom {
border-left: 5px solid transparent;
border-right: 5px solid transparent;
border-radius: 4px;
border-top: 5px solid #272833;
height: 0;
margin-left: 4px;
margin-top: 4px;
width: 0;
}
.component-tabs .navigation-bar-light .nav-link {
```

```
JavaScript
<script type="module">
const fragmentElement = ...;
const configuration = ...;
const dropdown = fragmentElement.querySelector('.navbar-collapse');
const dropdownButton = fragmentElement.querySelector('.navbar-toggler');
const editMode = layoutMode === 'edit';
const persistedTabKey = 'tabsFragment_' + fragmentNamespace + '_per';
const tabItems = [].slice.call(
fragmentElement.querySelectorAll(
`[data-fragment-namespace="${fragmentNamespace}"] .nav-link`
));
const tabs = document.querySelectorAll('.nav-item');
const tabsFragment = document.querySelector(`#${persistedTabKey}`);
const tabsFragmentValue = tabsFragment.value;
const tabsFragmentList = tabsFragmentList || tabsFragmentValue;
const tabsFragmentListLength = tabsFragmentList.length;
const tabsFragmentListIndex = tabsFragmentList.indexOf(tabsFragmentValue);
const tabsFragmentListIndexLength = tabsFragmentListIndex.length;
```

01 Liferay Features

Site Initializer Client Extension

Allows you to programmatically create or update a site, complete with configurations and content.

Replaces From Theme

- [Resources Importer](#) (sharing of theme resources)

Documentation

- [Using a Site Initializer Client Extension](#)

Examples

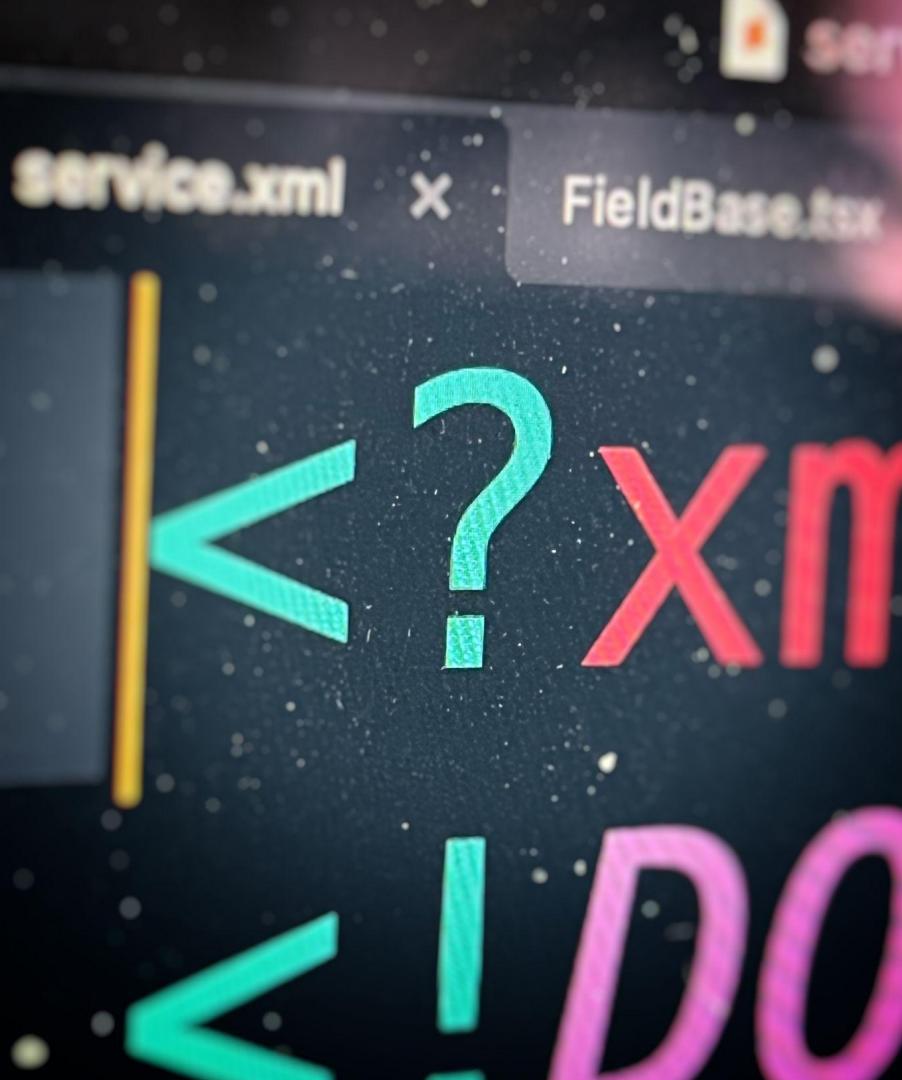
- [Site Initializer Extender Test Bundle](#)
- [gov-uk-site-initializer](#)
- [commerce-content-site-initializer](#)

```
↳ liferay-sample-site-initializer git:(master) cat client-extension.yaml
liferay-sample-site-initializer:
  name: Liferay Sample Site Initializer
  oAuthApplicationHeadlessServer: liferay-sample-site-initializer-oauth-application-headless-server
  siteExternalReferenceCode: LIFERAY_SAMPLE
  siteName: Liferay Sample
  type: siteInitializer
liferay-sample-site-initializer-oauth-application-headless-server:
  .serviceAddress: localhost:8080
  .serviceScheme: http
  name: Liferay Sample OAuth Application Headless Server
  scopes:
    - Liferay.Headless.Site.everything
  type: oAuthApplicationHeadlessServer
↳ liferay-sample-site-initializer git:(master) tree .
.
└── client-extension.yaml
    └── site-initializer
        ├── documents
        │   └── group
        │       └── sample.txt
        └── journal-articles
            ├── journal_article.json
            └── journal_article.xml
```

5 directories, 4 files

02

How your Business Requirements Affect your Implementation

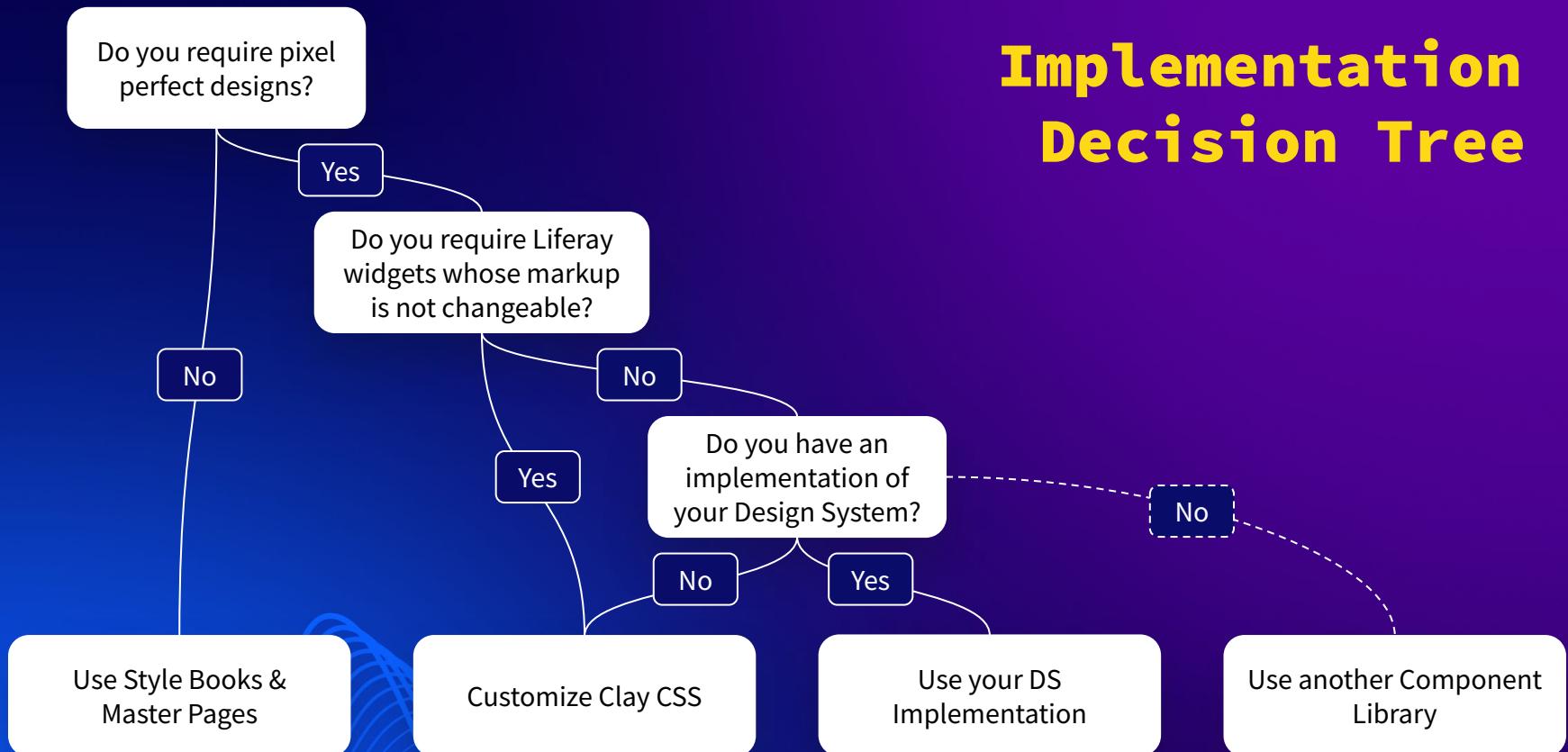


02 Business Requirements

Diagnostic Questions

- 1. What are your design requirements?**
 - Pixel Perfect VS Brand Colors
- 2. Do you have an existing design system?**
 - Does it have an implementation?
(Component Library)
 - Just designs, no markup?
- 3. What Liferay features does your project need?**
 - Which Liferay widgets will need to be restyled?
 - Is the markup of those widgets changeable?

Implementation Decision Tree



03

Example Implementations (Demo)

04

Questions



bit.ly/DevConDS