Optimizing an Existing Theme for Liferay 7

Migrating your theme from Liferay 6.2 to Liferay 7 is easier than you may think, thanks to the themes generator. This tutorial assumes that you already have the themes generator installed and a 6.2 theme that was developed with the Plugins SDK.

This tutorial demonstrates how to:

- Upgrade your theme from Liferay 6.2 to Liferay 7

Go ahead and get started.

Importing Your 6.2 Theme

The first thing you will need to do is import your 6.2 theme to use the new Node.js theme development tools. Follow the steps below to get your theme set up:

1. Navigate to the directory you want to import your theme into and open the command line and run the following command:
2. `yo liferay-theme:import`

   This runs the import sub-generator for the themes generator.

3. Enter the direct path for your theme to import and press `Enter`.

   Note: the themes import sub-generator does not support relative paths.

   The theme's modified files are all copied and migrated to a newly created `src` directory. A `gulpfile.js`, `liferay-theme.json`, `package.json` file and a `node_modules` directory is added to the existing theme.

   Next, the `gulp init` task runs and prompts you with a couple questions.

4. Enter the path to your app server.
5. Enter the URL to your production or development site, or press `Enter` to accept the default `http://localhost:8080`.

Your theme is now set up to use the Node.js build tools and theme gulp tasks. However, the theme is still configured for Liferay 6.2. To upgrade your theme to Liferay 7, follow the steps in the next section.

Updating Your Theme with the Gulp Upgrade Task

Now that your theme is set up to use the Node.js build tools, you can use the `gulp upgrade` task to start the upgrade process. Follow the steps below to upgrade your theme:

1. Navigate to your newly imported theme's root directory and run the following command:
2. `gulp upgrade`

   A backup of your theme's files is added to a `_backup` folder. If you wish to revert your theme to 6.2, you can run the `gulp upgrade:revert` task at any time.

3. Press `Enter` to rename all of the existing `.css` files to `.scss`. All Sass files now use the `.scss` extension and Sass partials are indicated with a _ at the beginning of the file name.

   the upgrade task checks all of the theme's files and upgrades them, when possible, and leaves suggestions for other updates.

   As part of the gulp upgrade task, your theme's Bootstrap code is upgraded from version 2 to version 3, to comply with Liferay 7. The changes and suggestions are printed in the command line.

   Note: the Font Awesome icon imports have been relocated to the `_aui_variables.scss` file. If you had previously modified that file and currently have variables in it, you will need to add the imports below for the Font Awesome Icons to render properly in Liferay:

   ```
   // Icon paths
   $FontAwesomePath: "aui/lexicon/fonts/alloy-font-awesome/font";
   $font-awesome-path: "aui/lexicon/fonts/alloy-font-awesome/font";
   $icon-font-path: "aui/lexicon/fonts/";
   ```

   Once you've made any remaining changes to your theme files, you can move on to the next section.

Updating Your CSS Responsiveness

The `respond-to` media queries that were used in Liferay DXP 6.2 have been replaced with the explicit media queries. Follow the step below to update your responsiveness.

1. Update your 6.2 `respond-to` media queries with the matching ones below:

   Media Query Replacements

6.2	7.0
`@include respond-to(phone)`	`@include media-query(null, $screen-xs-max)`
`@include respond-to(tablet)`	`@include media-query(sm, $screen-sm-max)`
`@include respond-to(phone, tablet)`	`@include media-query(null, $breakpoint_tablet - 1)`
`@include respond-to(desktop, tablet)`	`@include sm`
`@include respond-to(desktop)`	`@include media-query($breakpoint_tablet, null)`

   Now that your responsiveness is updated, you can build your theme files.

2. Open the command line and run the `gulp build` task.

   This builds the base files for the theme. Your core theme files should be upgraded now. You can use the theme files in the `build` directory as a starting point for any future theme modifications.

   Now that your CSS and theme templates are updated, you can move on to the resources importer next.

Updating the Resources Importer

The resources importer has undergone several changes that affect the configuration files and directory structure. Follow the steps in the sections below to make the required changes.

Updating liferay-plugin-package.properties

1. Open the `liferay-plugin-package.properties` file and remove the `required-deployment-contexts` line.

   This property is no longer required, as the resources importer is now an OSGI module that is included in Liferay DXP.

   The class name has changed slightly, so you'll need to update it.

2. Update the value of the `resources-importer-target-class-name` property to the value below:

```
com.liferay.portal.kernel.model.Group
```

With your configuration files updated, you can move onto the directory structure and web content article changes next.

Updating the Web Content Articles

In versions prior to Liferay DXP 7, basic web content articles did not require a structure and template, and could be of type HTML. In Liferay DXP 7 all articles must have a structure and template and be of type XML. Follow the steps below to update your directory structure:

1. Open the `src/resources-importer/journal/articles/` directory and create a folder, for example `Basic Web Content`, to hold your articles.

1. Rename the extension for any `.html` articles you have to `.xml`.
2. Follow the pattern below to update your HTML articles to XML:

```
<?xml version="1.0"?>

<root available-locales="en_US" default-locale="en_US">
        <dynamic-element name="content" type="text_area"
        index-type="keyword" index="0">
                <dynamic-content language-id="en_US">
                        <![CDATA[
                          HTML CONTENT GOES HERE
                          ]]>
                  </dynamic-content>
          </dynamic-element>
 </root>
```

Once your articles are updated, you can create the template and structures next.

Before Liferay DXP 7 web content structures were of type XML. Now web content structures must be of type JSON. You can manually recreate your structure in Liferay DXP 7 to ensure that it is correct. For basic web content articles you can use the structure covered in the next step.

15. Create a `[structure-name].json` file in the `resources-importer/journal/structures/` directory that matches the name of the folder you created in step 1. For basic web content articles you can use the structure below:

```
 {
     "availableLanguageIds": [
         "en_US"
     ],
     "defaultLanguageId": "en_US",
     "fields": [
         {
             "label": {
                 "en_US": "Content"
             },
             "predefinedValue": {
                 "en_US": ""
             },
             "style": {
                 "en_US": ""
             },
             "tip": {
                 "en_US": ""
             },
             "dataType": "html",
             "fieldNamespace": "ddm",
             "indexType": "keyword",
             "localizable": true,
             "name": "content",
             "readOnly": false,
             "repeatable": false,
             "required": false,
             "showLabel": true,
             "type": "ddm-text-html"
         }
     ]
 }
```

The structure above defines some basic attributes for the web content, sets the input field data as `html`, and identifies the web content by the `name` `content`. You can use this `name` as reference in your template next.

48. Open the folder you created in step 1 in the `resources-importer/journal/templates/` directory and create a `[template-name].ftl` file with the same name as your structure. For basic web content you can add the following line to the template:

```
${content.getData()}
```

This simply renders the html from the the matching web content structure with the `name` `content`.

Your theme is ready to be deployed to Liferay DXP 7!