



THEMELETS

Copyright ©2017 Liferay, Inc.

All Rights Reserved.

No material may be reproduced electronically or in print,
distributed, copied, sold, resold, or otherwise exploited
for any commercial purpose without express written
consent of Liferay, Inc.

S.P.A.C.E. SITE REQUIREMENTS

- ❖ So far, the Theme covers all of our global styles and responsive design.
- ❖ S.P.A.C.E. has some additional themes they want to create for their College and Department pages in the future.
- ❖ There are some styles and features they want to include in all of their future themes, including:
 - ❖ Making application moving on a page easier to visualize
 - ❖ Providing modern-style animation that emphasizes the menu when it's opened
- ❖ With this, they're looking for a way to provide a UI component that can be shared across themes and updated individually.
- ❖ We can take advantage of Liferay *Themelets* to accomplish this goal.

WHAT IS A THEMELET?

- ❖ Themelets are small, extendable, and reusable pieces of code that are implemented by a theme.
- ❖ They can consist of CSS, templates, images, and JavaScript just like a theme.
- ❖ They exist as *npm* packages and can be published to the npm registry for easy sharing and reuse.
- ❖ Themelets allow you to share pieces of code between themes, cutting down on boilerplate and repetition.
- ❖ You can either create new themelets or take advantage of existing themelets.

EXERCISE: VISUALIZING LAYOUT WITH THEMELETS

- ❖ The themelet generator is packaged within the `generator-liferay-theme` npm package.
- 1. **Run** `yo liferay-theme:themelet` in the command line or Terminal from your *liferay* folder.
- 2. **Name** the themelet *Application Drag Indicator*.
- 3. **Press** *Enter* to accept the default Id.
- 4. **Choose** *7.0* for the version.
- ✓ Now we have a Themelet with an `src` folder and a `package.json`.

THEMELET STRUCTURE

- ❖ The `src` directory is where all themelet files exist that will eventually be used by a theme.
 - ❖ *Note:* A themelet can have multiple JS files. Any file with the extension `.js`, in `src/js` will automatically be injected into a theme during build.
- ❖ For files not CSS or JS, they will be made available in the built theme's files.

File	Description
<code>src</code>	Contains src files of themelet.
<code>src/css/_custom.scss</code>	Contains themelet styles that get automatically injected to theme on theme build.
<code>package.json</code>	Where themelet meta-data is defined.

EXERCISE: MAKING OUR THEMELET AVAILABLE

- ❖ In order to use your themelet in your theme, you must install it globally.
 - ❖ There are two ways to make a themelet available to a theme:
 1. **npm link**: Creates a symlink in the global npm module directory
 2. **npm install -g**: Copies the files. If you use this method, you will have to re-run `npm install -g` to make any changes to your themelet available.
 - ❖ To install your themelet as a global npm module, you can run either of these options from the root of the themelet.
 1. **Go to** the *application-drag-indicator-themelet* directory in the command line or the Terminal.
 2. **Run** the `npm link` command.
 - ❖ You will need admin access in the directory to run this command.
 - ❖ You may need to type `sudo npm link` on unix-based systems.
- ✓ Now we can add our dropzone code.

EXERCISE: ADDING APPLICATION DROPZONE CSS

- ❖ We can use this themelet to provide CSS that will allow us to visualize a page layout.
- 1. **Copy** the contents of the `application-drag-indicator.scss` file from *exercises/front-end-developer-exercises/03-theme-development/07-themelets*.
- 2. **Paste** it into the `_custom.scss` file in your *application-drag-indicator-themelet/src/css*.
- 3. **Save** the file.

EXERCISE: MENU ANIMATION THEMELET

- ❖ Now we can create the Menu Animation Themelet that we can reuse for any new theme.
- 1. **Run** `yo liferay-theme:themelet` in the command line or Terminal from your *liferay* folder.
- 2. **Name** the themelet *Product Menu Animation*.
- 3. **Press** *Enter* to accept the default Id.
- 4. **Choose** *7.0* for the version.
- ✓ Now we have our new Themelet with an `src` folder and a `package.json`.

EXERCISE: TWO IS BETTER THAN ONE

- ❖ Just as before, we'll need to make this new Themelet available to our Theme.
- 1. **Go to** the *product-menu-animation-themelet* directory in the command line or the Terminal.
- 2. **Run** the `npm link` command.
 - ❖ You will need admin access in the directory to run this command.
 - ❖ You may need to type `sudo npm link` on unix-based systems.
- ✓ Now we can add the animation code!

EXERCISE: ADDING ANIMATION CSS

1. **Copy** the contents of the `product-menu-animation.scss` file from *exercises/front-end-developer-exercises/03-theme-development/07-themelets*.
 2. **Paste** it into the `_custom.scss` file in your *product-menu-animation-themelet/src/css*.
 3. **Save** the file.
- ✓ Now we can add our new themelets to the theme to deploy!

EXERCISE: INSTALLING OUR THEMELETS

1. **Open** the command line or Terminal.
2. **Go to** the root folder for the `space-program-theme`.
3. **Run** the *gulp extend* command.
 - ▶ There may be a brief pause while the command processes.
4. **Choose** *Themelet*.

EXERCISE: SELECTING BOTH THEMELETS

1. **Choose** *Search globally installed npm modules.*
 - Any globally-installed themelets should show up in the list.
 2. **Press** *Space* to select the `application-drag-indicator-themelet`.
 3. **Press** *Space* to select the `product-menu-animation-themelet`.
 4. **Press** *Enter* to apply the themelet.
 5. **Run** *gulp deploy* in your theme.
- ❖ The themelet will now be added as an npm dependency to your theme.
- ❖ Once deployed, this themelet will only display when changing where applications are placed on your page.

```
? Where would you like to search for themelets? Search globally installed npm modules (development purposes only)
? Select a themelet
> ● application-drag-indicator-themelet
  ● product-menu-animation-themelet
```

THEMELET CSS

- ❖ Themelet CSS and JS files will automatically be injected into your theme during the build task.
- ❖ Themelet CSS will be injected via Sass imports into the theme's `_custom.scss` file wherever the following CSS comment exists:

```
/* inject:imports */  
/* endinject */
```

THEMELET JAVASCRIPT

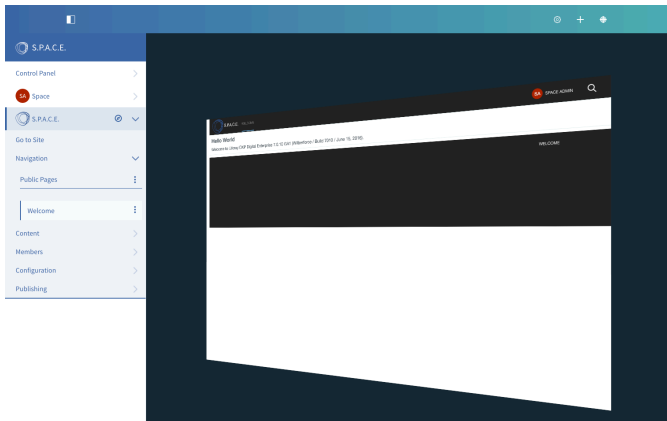
- ❖ Themelet JS will be injected via `<script>` tags into the theme's `portal_normal` template wherever the following HTML comment exists:

```
<!-- inject:js -->  
<!-- endinject -->
```

- ❖ These comments can be moved anywhere in the files they reside in. If you remove these comments, the themelet resources in question will not be added during the build.
 - *Note:* the build task is also run during the deploy and `deploy:gogo` tasks, so themelet files will also be injected during deploy.

SUCCESSFULLY ADDED

- ❖ Once deployed, we will see both the clear outline when moving applications around as well as the animation when we click on the Menu.



INSTALLING FROM THE NPM REGISTRY

- ❖ You can also choose from existing themelets on the npm registry.
- ❖ The list of themelets can be found here:
<https://www.npmjs.com/search?q=Themelet>
- ❖ Publishing to or extending from this list allows you to collaborate with other developers creating reusable pieces of code.

Notes: