



STYLING APPLICATIONS

Copyright ©2017 Liferay, Inc.

All Rights Reserved.

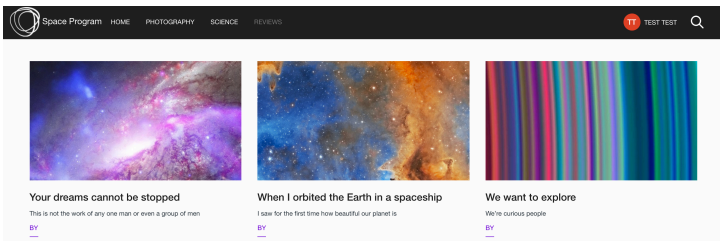
No material may be reproduced electronically or in print,
distributed, copied, sold, resold, or otherwise exploited
for any commercial purpose without express written
consent of Liferay, Inc.

WHAT YOU NEED

- ❖ This module uses:
 - ❖ **Installed** theme:
 - ❖ `space-program-theme.war`
- ❖ You can find the theme in *exercises/front-end-developer-exercises/05-front-end-developer/03-theme-development/solutions*.
- ❖ Our Theme includes some content and *Application Display Templates (ADTs)* for us to take a look at.

ASSET PUBLISHER ADT EXAMPLE

- ❖ Let's look at an example of an ADT bundled in our theme.
 - ❖ The theme let us create some useful assets in a sample site
- ❖ We have a number of Asset Publisher ADTs in use on the sample *Space Program* site generated by the Resources Importer.
- ❖ We'll look at the *Entry List 3 Item* ADT.



ADTS AND WEB CONTENT TEMPLATES

- ❖ *Web Content Templates* and ADTs use the same basic type of template.
- ❖ Since they both use FreeMarker, it's easy to write for both.
- ❖ The difficulty can be in remembering what makes them different.
- ❖ ADTs reinforce styling, but have the main purpose of displaying (*rendering*) different types of assets.
- ❖ Let's look at our example ADT in detail so you can see what that looks like.
- ❖ This ADT displays content articles as image tiles with information below with 3 columns.

EXERCISE: NAVIGATING TO ADT CREATION UI

- ❖ You'll want to make sure you're on the *Space Program* site to check out these examples.
1. **Open** the *Menu*.
 2. **Go to** *Site Administration*→*Configuration*→*Application Display Templates* in the Space Program Site.
 3. **Open** the *Entry List 3 items* ADT.

Name

Entry List 3 Items ADT

Details

Script

Search

Fields

- Asset Entries *
- Asset Entry

```

1 <div if entries.has_content>
2   ${request.setAttribute("displayMode", 2)}
3   ${request.setAttribute("colMd", "col-md-4")}
4
5   <div class="blog-list container-fluid-1280">
6     <div class="row">
7       <div list entries as curEntry>
8         <div assign
9           assetRenderer = curEntry.getAssetRenderer()
10          viewURL = (assetLinkBehavior != "showFullContent")?then(assetPub
11            lisherHelper.getAssetViewURL(renderRequest, renderResponse, curEntry, true),
12            assetPublisher.getAssetViewURL(renderRequest, renderResponse, curEntry))

```

ADTS HAVE ONLY ONE JOB

- ❖ The main job of ADTs is to *display dynamic data in markup*.
- ❖ This ADT is for the *Asset Publisher*, whose main content is *assets*.
- ❖ You'll notice the ADT gets a list of assets to display.
- ❖ Let's look more closely at how the assets are displayed and styled consistently.

MAKING ADT DESIGN EASIER

- ❖ The built-in editor makes it easier to reference variables.
- ❖ You can see an example of that in the *Entry List 3 Item* ADT.
- ❖ The ADT takes the default `Asset Entries *` field and expands upon it.
- ❖ Here is what is generated when you select `Asset Entries *` in the editor:

```
<#if entries?has_content>
  <#list entries as curEntry>
    ${curEntry.getTitle(locale)}
  </#list>
</#if>
```

ADDING CUSTOM ADT CODE

- ❖ Within the `<#if>` Structure, there are attributes set on the displayed content:

```
<#if entries?has_content>
    ${request.setAttribute("displayMode", 2)}
    ${request.setAttribute("colMd", "col-md-4")}
    ...
    ${request.setAttribute("author", "" )}
    ${request.setAttribute("displayMode", 0)}
    ${request.setAttribute("viewURL", "" )}
</#if>
```

- ❖ We also have this markup added:

```
<div class="blog-list container-fluid-1280">
    <div class="row">
        <#list entries as curEntry>
        ...
        </#list>
    </div>
</div>
```


EXERCISE: CREATING A NEWS FEED ADT

❖ Now that we've seen an ADT breakdown, let's add our own ADT for the S.P.A.C.E. front page *News* and *Events*.

1. **Open** the *Menu*.
2. **Click** on the *Site Selector* in *Site Administration*.
3. **Choose** the S.P.A.C.E. Site.
4. **Go to** *Configuration*→*Application Display Templates*.
5. **Click** *Add*→*Asset Publisher Template*.

EXERCISE: NAMING THE ADT

1. **Type** to name the ADT *Card List - Feed*.
2. **Open** the `card-list-feed-ap-adt.ftl` from *exercises/front-end-developer-exercises/o8-application-display-templates*.
3. **Copy** the contents.
4. **Paste** the contents in the template editor.
5. **Click** *Save*.

CARD LIST - FEED: EXPLANATION

1. In the Card List - Feed ADT, we check to see if there are any entries, and then iterate through the list.
 - You can hover over Asset Entries * for more information on entries.
2. Next, we use Lexicon CSS Cards to create the card layout.
 - <http://liferay.github.io/clay/content/cards/>
3. Finally, we'll display the article title, URL, and summary from our Asset Renderer.
 - As we're displaying Web Content Articles, see JournalArticleAssetRenderer for more information.
4. As front-end developers, we can collaborate with our back-end team to display dynamic content.

ASSET URLS

- ❖ We're using a URL for the article called *View in Context*.
- ❖ *View in Context* needs the article to be displayed on a page in order to work.
- ❖ If you'd like to test the URL, you'll want to add the Web Content Article to a page somewhere before using the link.

EXERCISE: CREATING AN EVENT FEED ADT

1. **Click** *Add*→*Asset Publisher Template*.
2. **Type** to name the ADT *Card List - Events*.
3. **Open** the `card-list-events-ap-adt.ftl` from *exercises/front-end-developer-exercises/o8-application-display-templates*.
4. **Copy** the contents.
5. **Paste** the contents into the template editor.
6. **Click** *Save*.

CARD LIST - EVENTS: EXPLANATION

1. In the *Card List - Events* ADT, the code is almost identical except for a few CSS changes.
 - Instead of an rss icon, we displayed a calendar icon and changed the font.
2. The main difference is that we configured *Asset Publisher* to pull *Web Content Articles* that use the *Events Web Content Structure*.
3. We'll see the results of these changes in a few slides.
4. *Asset Publisher*, *Application Display Templates*, and *Web Content Templates* gives us precise control to create consistent branding and work better with our content marketing team in a way that allows both teams to focus on our areas of our expertise.

EXERCISE: CONFIGURING ASSET PUBLISHERS TO USE ADTS

- ❖ Once we have created our ADTs, we can use them with our Asset Publishers.
- 1. **Click** *Go to Site* under *Site Administration* in the *Menu*.
- 2. **Click** on the *Add* menu near the top right.
- 3. **Open** the *Applications*→*Highlighted* section.
 - ❖ The Asset Publisher can also be found in the *Applications*→*Content Management* section.
- 4. **Add** an Asset Publisher above the *About S.P.A.C.E.* article.
- 5. **Add** an Asset Publisher below the *About S.P.A.C.E.* article.
- ❖ The Asset Publishers can be configured to display Web Content Structure, allowing our content team to make sure our News and Events are displaying on the front page.

EXERCISE: CONFIGURING THE FEED ADT

1. **Open** the *Options* menu of the first Asset Publisher.
2. **Click** *Configuration*.
3. **Click** the *Display Settings* tab.
4. **Choose** *Card List - Feed* under *Display Template*.
5. **Save** the changes.
6. **Close** the *Configuration* pop-up.

EXERCISE: CONFIGURING THE EVENTS TEMPLATE

1. **Open** the *Options* menu of the second Asset Publisher.
2. **Click** *Configuration*.
3. **Click** the *Display Settings* tab.
4. **Choose** *Card List - Events* under *Display Template*.
5. **Save** the changes.
6. **Close** the *Configuration* pop-up.

BRINGING IT ALL TOGETHER

- ❖ Let's import some content created by our content team in order to see everything in action.
- ❖ The content team has already configured the Asset Publishers to pull in the content they want to display.
- ❖ For the Asset Publisher at the top, they've configured it to pull in Web Content Articles that use the *News Structure*.
- ❖ For the second Asset Publisher, they've configured it to pull in Web Content Articles that use the *Events Structure*.
- ❖ The front-end team provided consistent styling for the Web Content using Web Content Templates.
- ❖ The front-end team will be responsible for applying the design, which we've completed above with ADTs.

EXERCISE: IMPORTING CONTENT USING ADTS

1. **Go to** *Menu*→*Site Administration*→*Publishing*.
 2. **Click** on *Import*.
 3. **Click** on the *Add* button at the bottom right.
 4. **Choose** the `Landing_Page_Redesign.lar` from *exercises/front-end-developer-exercises/o8-application-display-templates/space-front-page-content*.
 5. **Click** *Continue*→*Import*.
- ✓ Now we have content on display with our templates!

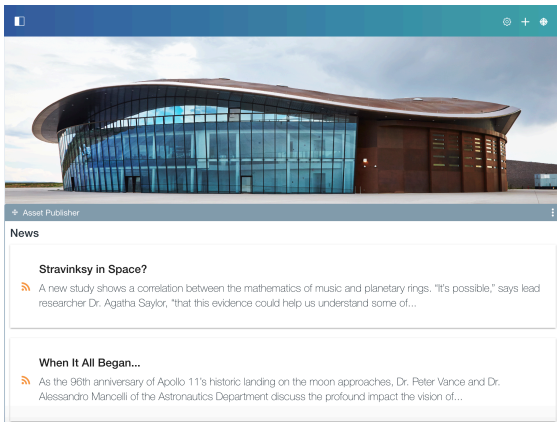
EXERCISE: APPLYING THE THEME

- ❖ When we imported our content changes, it reverted the theme selection.
- ❖ Let's go ahead and re-apply the Space Program theme.

1. **Go to** *Menu*→*Site Administration*→*Navigation*.
 2. **Open** the *Options* menu next to *Public Pages*.
 3. **Click** *Configure*.
 4. **Click** *Change Current Theme*.
 5. **Choose** the *Space Program Theme*
 6. **Click** *Save*.
- ✓ Now our theme is applied!

READY FOR LIFT OFF!

- Now we've changed the overall look-and-feel of the S.P.A.C.E. site and its content.



Notes: