LIFERAY®

LIFERAY INTERNAL APIS

LIFERAY APIS

- The `Liferay` JavaScript object is populated with some helpful tools you're able to use.
- Below, you'll find some of the various APIs on the object.

LIFERAY.THEMEDISPLAY

```
Liferay.ThemeDisplay = {
    <c:if test="<%= layout != null %>">
        getLayoutId: function() {
            return '<%= layout.getLayoutId() %>';
        },
        getLayoutRelativeURL: function() {
            return '<%= PortalUtil.getLayoutRelativeURL(layout, themeDisplay) %>';
        },
        getLayoutURL: function() {
            return '<%= PortalUtil.getLayoutURL(layout, themeDisplay) %>';
        },
        getParentLayoutId: function() {
            return '<%= layout.getParentLayoutId() %>';
        },
        ...
    </c:if>
    ...
};
```

LIFERAY.AUI

```
Liferay.AUI = {
    Object {}
    getAvailableLangPath: function() {...}
    getCombine: function() {...}
    getComboPath: function() {...}
    getDateFormat: function() {...}
    getEditorCKEditorPath: function() {...}
    getFilter: function() {...}
    getFilterConfig: function() {...}
    getJavaScriptRootPath: function() {...}
    getLangPath: function() {...}
    getPortletRootPath: function() {...}
    getStaticResourceURLParams: function() {...}
}
```

LIFERAY.BROWSER

```
var Liferay = Liferay || {};
        Liferay.Browser = {
            acceptsGzip: function() {
                return <%= BrowserSnifferUtil.acceptsGzip(request) %>;
            },
            getMajorVersion: function() {
                return <%= BrowserSnifferUtil.getMajorVersion(request) %>;
            },
            getRevision: function() {
                return '<%= BrowserSnifferUtil.getRevision(request) %>';
            },
            getVersion: function() {
                return '<%= BrowserSnifferUtil.getVersion(request) %>';
            },
            isAir: function() {
                return <%= BrowserSnifferUtil.isAir(request) %>;
            },
            ...
        };
```

LIFERAY.PROPSVALUES

```
Liferay.PropsValues = {
    JAVASCRIPT_SINGLE_PAGE_APPLICATION_TIMEOUT: <%= PrefsPropsUtil.getInteger(
    themeDisplay.getCompanyId(), PropsKeys.
    JAVASCRIPT_SINGLE_PAGE_APPLICATION_TIMEOUT, PropsValues.
    JAVASCRIPT_SINGLE_PAGE_APPLICATION_TIMEOUT) %>,
    NTLM_AUTH_ENABLED: <%= PrefsPropsUtil.getBoolean(themeDisplay.getCompanyId(),
    PropsKeys.NTLM_AUTH_ENABLED, PropsValues.NTLM_AUTH_ENABLED) %>,
    UPLOAD_SERVLET_REQUEST_IMPL_MAX_SIZE: <%= PrefsPropsUtil.getLong(PropsKeys.
    UPLOAD_SERVLET_REQUEST_IMPL_MAX_SIZE, PropsValues.
    UPLOAD_SERVLET_REQUEST_IMPL_MAX_SIZE) %>
};
```

LIFERAY.PORTLETKEYS

```
Liferay.PortletKeys = {
            DOCUMENT_LIBRARY: '<%= PortletKeys.DOCUMENT_LIBRARY %>',
            DYNAMIC_DATA_MAPPING:
            'com_liferay_dynamic_data_mapping_web_portlet_DDMPortlet',
            ITEM_SELECTOR: '<%= PortletKeys.ITEM_SELECTOR %>'
};
```

LIFERAY.DATA

```
Liferay.Data = Liferay.Data || {};
Liferay.Data.NAV_SELECTOR = '#navigation';
Liferay.Data.NAV_SELECTOR_MOBILE = '#navigationCollapse';
Liferay.Data.isCustomizationView = function() {
    return <%= (layoutTypePortlet.isCustomizable() && LayoutPermissionUtil.contains
    (permissionChecker, layout, ActionKeys.CUSTOMIZE)) %>;
};
Liferay.Data.notices = [
    null
    <c:if test="<%= permissionChecker.isOmniadmin() && PortalUtil.isUpdateAvailable
    () %>">
        , {
            content: '<a class="update-available" href="<%= themeDisplay.
            getURLUpdateManager() %>"><liferay-ui:message key="updates-are-
            available-for-liferay" /></a>',
            toggleText: false
        }
    </c:if>
    ...
];
```

LIFERAY.AUTHTOKEN

```
Liferay.authToken = '<%= AuthTokenUtil.getToken(request) %>';
Liferay.currentURL = '<%= HtmlUtil.escapeJS(currentURL) %>';
Liferay.currentURLEncoded = '<%= HtmlUtil.escapeJS(HttpUtil.encodeURL(currentURL))
 %>';
```

LIFERAY.CURRENTURL

```
Liferay.currentURL = '<%= HtmlUtil.escapeJS(currentURL) %>';
Liferay.currentURLEncoded = '<%= HtmlUtil.escapeJS(HttpUtil.encodeURL(currentURL))
 %>';
```

LIFERAY.CURRENTURLENCODED

```
Liferay.currentURLEncoded = '<%= HtmlUtil.escapeJS(HttpUtil.encodeURL(currentURL))
 %>';
```

LIFERAY.PORTLET

- Custom applications that implement their own module will add to the pre-existing `Liferay.Portlet` object.
- An example of this can be seen in the `Google Maps Portlet` here: *https://github.com/liferay/liferay-portal/blob/master/modules/apps/google-maps/google-maps-web/src/main/resources/META-INF/resources/js/main.js*
- Adding the module `Liferay.Portlet` makes it available globally.

LIFERAY.PORTLETURL

- This provides a way to create Liferay PortletURLs such as `actionURL`, `renderURL`, and `resourceURL` through JavaScript.
- Here is an example:

```
var portletURL = Liferay.PortletURL.createURL(themeDisplay.
getURLControlPanel());

portletURL.setDoAsGroupId('true');
portletURL.setLifecycle(Liferay.PortletURL.ACTION_PHASE);
portletURL.setParameter('cmd', 'add_temp');
portletURL.setParameter('javax.portlet.action',
'/document_library/upload_file_entry');
portletURL.setParameter('p_auth', Liferay.authToken);
portletURL.setPortletId(Liferay.PortletKeys.DOCUMENT_LIBRARY);
```

LIFERAY.MAPS

- In DXP, the new Loader makes it possible to add some third-party modules and dependencies for your modules.
- These two objects make it possible to see what modules are being loaded into the application for each module.
- Here is an example of `Liferay.MAPS`:

```
Liferay.MAPS = {
    classic-theme: "classic-theme@1.0.2"
    com.liferay.announcements.web: "com.liferay.announcements.web@1.0.5"
    com.liferay.application.list.taglib: "com.liferay.application.list.
    taglib@2.0.4"
    com.liferay.asset.browser.web: "com.liferay.asset.browser.web@1.0.6"
    com.liferay.asset.categories.admin.web: "com.liferay.asset.categories.
    admin.web@1.0.9"
    com.liferay.asset.categories.navigation.web: "com.liferay.asset.categories
    navigation.web@1.0.4"
    com.liferay.asset.display.web: "com.liferay.asset.display.web@1.0.1"
    ...
}
```

LIFERAY.MODULES

- This shows all of the included module names, the module they belong to, and that module's version number.
- Here is an example of `Liferay.MODULES`:

```
Liferay.MODULES = {
    ...
    frontend-image-editor-capability-crop@1.0.2/CropHandles.es: {...}
    frontend-image-editor-capability-crop@1.0.2/CropHandles.soy: {
        conditionalMark: false,
        dependencies: ["exports", "metal-component/src/Component", "metal-
        soy/src/Soy"]
    }
    "frontend-image-editor-capability-crop@1.0.2/CropHandles.soy": {...}
    ...
}
```

- This object displays each module and its dependencies.

Notes: