

## 1 Difference between Liferay 6.2 and Liferay Dxp?

	DXP	6.2
<b>BOOT STRAP, AUI Version</b>	Liferay 7 support Bootstrap 3 UI framework. AUI 3	Liferay 6.2 support Bootstrap 2 UI framework. AUI 2
<b>DEPLOYMENT FILE FORMAT</b>	Support war and Jar	Support War
<b>OSGI VS TRADITIONAL</b>	Traditional Application Container	OSGI Container
<b>SEARCH ENGINE ELASTIC VS LUCENE SOLR</b>	Liferay 7 inbuilt search engine Elastic Search. It supports SOLR search as well.	Liferay 6.2 in built search engine Apache Lucene. Liferay 6.2 have SOLR web to enable SOLR search.
<b>SINGLE PAGE APP BUILT IN SUPPORT</b>	Liferay 7 have Single Page Application inbuilt support using Senna.js	Liferay 6.2 does not have inbuilt Single Page Application support but we can integrate SPA framework to achieve it.
<b>DESIGN SUPPORT</b>	Clay and Lexicon design	Do not have
<b>GOGO SHELL ACTIVE DEACTIVE VS INSTALL UNINSTALL</b>	Activate / Deactivate from Gogo shell	Require to remove

## 2 What is OSGI?

- **Open Services Gateway Initiative:** - OSGI stands for Open Services Gateway Initiative.
- **Modular Application:** - OSGI provides an architecture for developing and deploying modular applications and libraries.
- **Dynamic Module System:** - Sometimes OSGI is also referred as Dynamic Module System for Java. In this article we will explore the basics of OSGI.

## 3 OSGI BENEFITS

<b>Lightweight, Fast and Secure</b>	Liferay DXP's runtime framework is lightweight, fast, and secure.
<b>No Restart</b>	Modules' dependencies are managed automatically by the container, dynamically (no restart required).
<b>Dynamic Module Management</b>	The container manages module life cycles dynamically. Modules can be installed, started, updated, stopped, and uninstalled while Liferay DXP is running, making deployment a snap.
<b>Different Module Can have different dependency</b>	Modules and packages are semantically versioned and declare dependencies on specific versions of other packages.
<b>Parallel Work Possible</b>	Team members can develop, test, and improve modules in parallel.

--	--

- void init(PortletConfig config)
- void render(RenderRequest request, RenderResponse response)
- void processAction(ActionRequest request, ActionResponse response)
- void processEvent(EventRequest request, EventResponse response)
- void serveResource(ResourceRequest request, ResourceResponse response)
- void destroy()

## 6 Portlet Life Cycle Methods

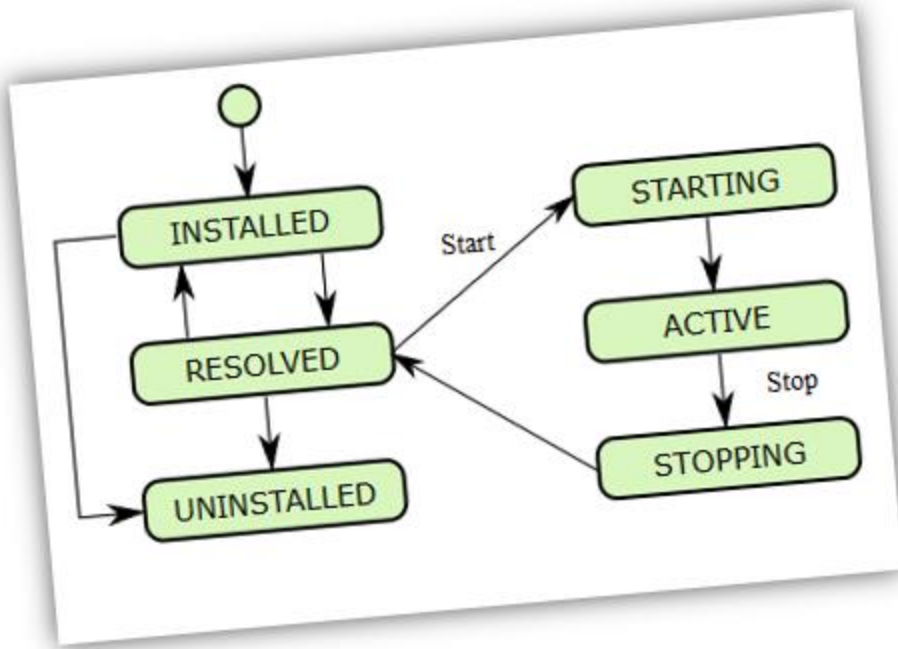
Init	<ul style="list-style-type: none"> <li>• Portlet Container call it</li> <li>• Useful to initialize</li> <li>• Only one time during life cycle</li> </ul>
Render	<ul style="list-style-type: none"> <li>• When portlet display on page</li> <li>• Useful to generate dynamic page.</li> <li>• We can create render URL to invoke render method. For example</li> <li>• <b>&lt;portlet:renderURL var="myURL"/&gt;</b></li> </ul>
processAction	<ul style="list-style-type: none"> <li>• When we invoke <b>action URL</b> the <b>processAction</b> method is called.</li> <li>• A process action is generally used to process submitted form or writing data to database.</li> <li>• <b>&lt;portlet:actionURL var="myURL" name="process"/&gt;</b></li> <li>• As soon as a <b>process action</b> is invoked it invoke <b>render method</b>.</li> </ul>
processEvent	<ul style="list-style-type: none"> <li>• This method useful in Inter portlet Communication</li> <li>• Introduce in 286 standard</li> <li>• As soon as process event method is completed the portlet container invokes <b>render method</b>.</li> </ul>
serveResource	<ul style="list-style-type: none"> <li>• Called by the portlet container to allow the portlet to generate the resource content based on its current state.</li> </ul>

	<ul style="list-style-type: none"> <li>• The portal / portlet container must not render any output in addition to the content returned by the portlet.</li> <li>• The portal / portlet container should expect that the portlet may return binary content for a renderResource call.</li> <li>• This portlet life cycle method is added in JSR 286.</li> <li>• This method is basically required to generate non HTML based content like JSON, XML etc. We can implement AJAX using this method. A serveResource method can be invoked by creating resource URL. For example</li> <li>• <code>&lt;portlet:resourceURL var="myResourceUrl"/&gt;</code></li> </ul>
destroy	<ul style="list-style-type: none"> <li>• Called by the portlet container to indicate to a portlet that the portlet is being taken out of service.</li> <li>• The default implementation does nothing.</li> <li>• When a portlet is undeployed this method is called by the portlet container.</li> </ul>

**Which JSR-286 portlet class extends?**

**Ans)** GenericPortlet

**OSGI bundle life cycle**



**INSTALLED:** The bundle has been successfully installed.

**RESOLVED:** All Java classes that the bundle needs are available. This state indicates that the bundle is either ready to be started or has stopped.

**STARTING:** The bundle is being started, the **BundleActivator.start** method has been called but the start method has not yet returned. When the bundle has an activation policy, the bundle will remain in the **STARTING** state until the bundle is activated according to its activation policy.

**ACTIVE:** The bundle has been successfully activated and is running. Its Bundle Activator start method has been called and returned.

**STOPPING:** The bundle is being stopped. The **BundleActivator.stop** method has been called but the stop method has not yet returned.

### Application Display Template

- **Customize Display:** - It allows to customize Portlet display dynamically
  - example, you want to show assets horizontally instead of vertically
  - in various sizes in asset publisher, you can achieve it using ADT.
  - We can apply it on asset publisher. E.g., News portlet where we show only date and headline, small
  - News Link, Headline, Date Click on It redirect to another page with full shows.

### What is Expando in liferay?

- **FEATURE:** - Liferay Custom Fields or Liferay Custom Attribute is a feature in Liferay.
- **UI AND API:** -Liferay provides UI and API to allow for the creation of customized fields associated with any entity
- **Extensibility:** - It provide extensibility without extending schema.

- ExpandoTableLocalServiceUtil
- ExpandoValueLocalServiceUtil
- ExpandoValue
- ExpandoTable

### HOOK VS EXT

	HOOK	EXT
WHAT IT DOES	CUSTOMIZE OVERRIDE DEFAULT BEHAVIOUR	EXTEND AND CUSTOMIZE EVERYTHING
Types	<ol style="list-style-type: none"><li>1 .Event Handlers<ul style="list-style-type: none"><li>• Application startup events</li><li>• Login events</li><li>• Service Events</li></ul></li><li>2 Language Bundles.</li><li>3 Jsp files(either portal or portlet folder).</li><li>4. Model Listeners.</li></ol>	Using Exts we can customize <ol style="list-style-type: none"><li>a).portal-impl as ext-portal-impl.</li><li>b).portal-service as ext-portal-service.</li><li>c).util-bridges as ext-portal-utilbridges..</li><li>d).util-java as ext-util-java.</li><li>e).util-taglib as ext-taglib.</li><li>f).web as ext-portal-web.</li></ol>

	5. Services.	
<b>Maintenances</b>	These are recommended way of extension and maintains easy.	These are recommended way of extension and maintains easy., These are recommended when there is no way of implementing required functionality using hooks.
<b>HOT DEPLOY</b>	Hot deployable	Not hot deployable and server must restart
<b>COMBINE WITH OTHER PLUGIN</b>	Combine with other plugins like portlets.	These are some difficulties and customize almost everything.

## Structure

- **Useful for Web content:** - Structures are the foundation for web content.
- **Provide Data Fields:** - They determine which fields are available to users as they create new items for display.
- **Improve Manageability:** - Structures not only improve manageability for the administrator, they also make it much easier for users to quickly add content.

**Structure and Template:** - Liferay we can create structure and templates.

- **Types of Items in your content** It defines items such as text, image, checkbox, calendar item etc.
- **Fill Data while creating content:** - User can fill data while creating content

**Template:** -

- **Display Item on Page:** - **template** uses a templating language to display a structure's items
- **Apply Style and Logic:** - you can apply styles and logic to create complex or interactive content.
- **Free marker & Velocity:** - you can create templates in [FreeMarker](#) or [Velocity](#).
- **CSS, JavaScript and HTML:** -Templates can contain CSS, HTML, JavaScript, and elements of the templating language it uses.

## Themes and Layout Templates

- **Look and Feel:** - A Theme provides the overall look and feel for a site.
- [Understanding the page layout](#) is crucial to targeting the correct markup for styling, organizing your content, and creating your site.
- Once you understand how the page is organized, you can develop your theme.

### **Understanding ServiceContext**

- The ServiceContext class holds contextual information for a service.
- It aggregates information necessary for features used throughout Liferay's portlets, such as permissions, tagging, categorization, and more.

### **What is Search container in Liferay**

- Liferay have very build in UI components and we can simply use those UI components when we develop portlets.
- Liferay Search Container is Liferay UI component to display data in Grid format

### **What are the different module types supported in Liferay 7?**

## Service Builder Liferay

### What is service builder?

**Generate Model, Service Dao layer:** - Liferay provides Service Builder tool for generating modal objects, Service layer and DAO layer

**Internally use spring and hibernate:** - Service builder internally uses spring and hibernate and saves lot of development time. below are pros and cons:

#### Advantages:

1. Service Builder uses Hibernate as underlying framework. you can see lot spring context files in META-INF folder
2. **Default Cache:** -Provides default cache utils
3. **Easy To develop:** - Easy to develop and saves lot of development time
4. Provides default interfaces to
5. **Database connection pool:** - Manages Database connection pool internally
6. Automatically generates required SQL scripts

#### Drawbacks:

7. **Not support foreign key:** - Liferay does not support entity relationship – Foreign key (One to Many and Many -Many relationship)
8. Platform dependency. you cannot use portlet in another environments

### Finder Method

- **Find column:** -
- **Return type:** - Collection etc.
- **Implement in impl class:** -
- **How do you enable auto generation of primary key in the service builder?**

**primary=" true"**

- *This specify that the given column is a Primary key.*

**iv. id-type=" increment"**

- *This specify that the given column is an auto-generated type.*

```
<column name="sectionId" type="long" primary="true" id-type="increment" />
```



### What are the steps to create finder method?

- Define finder tag and finder columns in Entity tag
- Run service builder
- Implement custom method in XXXLocalServiceImpl class
- Run service builder
- Call implemented method using XXXLocalServiceUtil class

### How to inject GroupLocalService in ServiceBuilder generated services?

- The reference element allows you to inject services from another service.xml within the same class loader. `<reference entity="Group" package-path="com.liferay.portal"></reference>`

### How to increase the column size? Liferay service builder

- In Liferay you can easily modify DB column size. Here is the example for Column with type String.
- By default, if we use type as a string in service.xml then size will be 75.
- By doing few changes inside portlet-model-hints.xml. This configuration file is available inside src/META-INF directory. To increase column size do something like this.

```
<field name="columnName" type="String">
```

```
<hint name="max-length">4000</hint>
```

```
</field>
```

- If max-length is less-than 4000 then column type will be Varchar.
- If max-length is equal to 4000 then column type will be STRING.
- If max-length is greater than 4000 then column type will be TEXT.

### How to Map Existing Table in DB from Liferay?

- service builder provides “table” and “db-name” elements to map existing tables like shown below:

### Custom Query: -

Custom SQL in liferay is robust and easy to use and below are steps to write Custom SQL queries.

- **Default.xml** :- Define custom SQL queries in default.xml
- **Custom Finder** :- Create custom Finder Util with {EntityName}FinderImpl.java in persistence package
- **Access CustomFinderUtil in ServiceImpl Class**:- Access Custom Finder Util in LocalServiceImpl class

### Dynamic Query

- Liferay Dynamic Query API uses Hibernate Criteria API, in which you no need to build queries SQL and all you need to know is Objects and its variables

### How to implement custom-sql in Liferay?

#### Step 1: Specify your Custom Sql

Create a folder custom-sql inside docroot/WEB-INF/src/custom-sql. Now create a default.xml inside custom-sql folder.

#### Step 2: Implement your Finder Method=

- Create a class XXXFinderImpl in the persistence package which extends BasePersistenceImpl class
- Run the service builder to generate the Finder interface and Util for Finder(XXXFinder and XXXFinderUtil)
- Modify XXXFinderImpl class to have it implements XXXFinder interface you just generated.

**Step 3:** Create a finder method inside XXXFinderImpl class

**Step 4:** Access your finder method from your service

Add the method inside XXXLocalServiceImpl class

Call the finder method by XXXFinderUtil.MethodName()

Run the service builder

Step 5: Access the method in action class by using XXXLocalServiceUtil

### Clustering

- cluster.link.enabled=true

- cluster.link.autodetect.address=www.google.com:80
- cluster.link.channel.name.control=[your control channel name]
- cluster.link.channel.properties.control=[your control channel properties]  
cluster.link.enabled=truecluster.link.channel.properties.control=/jgroups/tcp.xmlcluster.link.channel.properties.transport.0=/jgroups/tcp.xml

**Add a parameter to your app server's JVM on each node:**

-Djgroups.bind\_addr=[node\_ip\_address]

Extract the tcp.xml file from \$LIFERAY.HOME/osgi/marketplace/Liferay Foundation - Liferay Portal - Impl.lpkg/com . liferay . portal . cluster . multiple - [version]. jar/lib / jgroups -

9. WEB-INF/classes/jgroups/tcp.xml

**What is GroupId and CompanyId?**

- **GroupId is organizational or site id.** It's used if you want to separate your data between organization or sites.
- **CompanyId is the liferay portal instance id.** If you setup liferay to be multi-tenant, = this id is used to separate data between the virtual portal.

**2.What are the portlet Lifecycle methods?**

**init()**

- It is called when portlet is deployed

**render()(Render Phase)**

- Called to render the content. It represents Render Phase

### **processAction() (Action Phase)**

- Called when any action performed. It represents Action Phase

### **processEvent() (Event Phase)**

- Called when any Event is triggered. It represents Event Phase

### **serveResource() (Resource Phase)**

- Called when any Resource is served with resource URL. It represents Resource Serving Phase

### **destroy ()**

- When portlet is un-deployed

### **3. If we perform action or operation on on portlet in a page then what happens to other portlets in a same page?**

- if action() method is completed for one portlet then its' render() method is called along with render()
- method of all portlet on that page.

### **4. When init() method of portlet called?**

- Initialization phase of portlet represented by init() method.
- When we deploy the portlet in liferay server portlet container calls init() method of portlet.

### **5.What is window state?**

- window State represents current state of portlet.

### **6.What are the types of window States?**

- Maximize: It shows the portlet in Whole page
- Minimize:It shows the portlet in invisible stage.User explicitly need to make it either normal
- Normal: It shows the portlet in normal as it is default
- POP\_UP: Used to show portlet in modal popup

### **7. What are the default variables in a theme or Name some default velocity variables in a theme?**

Variable	Description
theme_display	Returns the themeDisplay Java Object and all its methods
portlet_display	Returns the portletDisplay Java Object and all its methods
layoutSet	Returns the page set

theme_timestamp	Prints the date in the current locale with the given format
theme_settings	Retrieves theme settings. See <a href="#">configurable theme settings</a> for more information.
root_css_class	Returns the root CSS class which indicates the direction of the page (ltr (left-to-right) by default)
css_class	Returns a string of the current classes applied to the body of the page
page_group	Retrieves the page group
css_folder	Returns the path to the theme's css folder
images_folder	Returns the path to the theme's images folder
javascript_folder	Returns the path to the theme's javascript folder
templates_folder	Returns the path to the theme's templates folder
full_css_path	Returns the full path, which includes the servlet context, to the theme's css
full_templates_path	returns the full path, which includes the servlet context, to the theme's templates
css_main_file	Returns the path to main.css
js_main_file	Returns the path to main.js
company_id	Returns the company ID
company_name	Returns the company name
company_logo	Returns the company logo's URL
company_logo_height	Returns the company logo's height
company_logo_width	Returns the company logo's width
company_url	Returns the URL of the home page for the company
time_zone	Returns the time zone for the current user
is_login_redirect_required	Returns whether a login redirect is required for the user
is_signed_in	Returns whether the user is signed in
group_id	Returns the group ID for the current user
time_zone	Returns the time zone for the current user
is_default_user	Returns if the user has a default role
is_female	Returns if the current user is Female
is_male	Returns if the current user is Male
is_setup_complete	Returns whether the user has configured their profile
language	Returns the native language for the current user
language_id	Returns the ID of the current locale

user_birthday	Returns the current user's birthday
user_comments	Returns comments from the user's profile
user_email_address	Returns the user's email address
user_first_name	Returns the user's first name
user_greeting	Returns the user's greeting
user_id	Returns the ID of the current user
user_last_login_ip	Returns the IP address that the user last logged in from
user_last_name	Returns the last name of the current user
user_login_ip	Returns the current user's current IP address
user_middle_name	Returns the user's middle name
user_name	Returns the current user's username
w3c_language_id	Returns the W3C language code of the current language

## URLs

Variable	Description
show_control_panel	Returns whether the current user has permission to view the Control Panel
control_panel_text	Returns the "control-panel" language key in the current user's locale, if they have permission to view the Control Panel
control_panel_url	Returns the URL to the Control Panel, if the current user has permission to view the Control Panel
show_home	Returns whether the current user is on a page
home_text	Returns the "home" language key in the current user's locale
home_url	Returns the URL to the home page
show_my_account	Returns whether the current user's account icon is visible
my_account_text	Returns the "my-account" language key in the current user's locale, if the user's account icon is visible
my_account_url	Returns the URL to the user's Account Settings page if the user's account icon is visible
show_sign_in	Returns whether the sign in link is visible
sign_in_text	Returns the "sign-in" language key in the current user's locale, if they are signed out
sign_in_url	Returns the sign in URL, if the current user is signed out
show_sign_out	Returns whether the sign out link is visible

sign_out_text	Returns the “sign-out” language key in the current user’s locale, if they are signed in
sign_out_url	Returns the sign out URL, if the current user is signed in

## Page

Variable	Description
the_title	Returns the current page’s title
selectable	Returns whether the current page is selectable
is_maximized	Returns whether the page is maximized
page	Returns the current page (layout)
is_first_child	Returns whether the current page is the first child page in the navigation
is_first_parent	Returns whether the current page is the first parent page in the navigation
is_portlet_page	Returns whether the current page is a widget page (portlet)
site_name	Returns the site’s name
is_guest_group	Returns whether the current page group is for guests
site_type	Returns the type of the current site: site, company site, organization site, or user site
site_default_url	Returns the default URL for the site
layout_friendly_url	Returns the friendly URL of the current page
portlet_id	Returns the portlet ID for the specified portlet

## Logo

Variable	Description
logo_css_class	Returns a string of the current classes applied to the logo.
use_company_logo	Returns whether the logo is displayed
site_logo_height	Returns the logo’s height
site_logo_width	Returns the logo’s width
show_site_name_supported	Returns whether the logo is configured to show the site name. The value is true if show_site_name_default is true.
show_site_name_default	Returns whether the Show Site Name Default theme setting is enabled
show_site_name	Returns whether the showSiteName property for the current pageset is enabled

logo_description	Returns the Site's name or nothing if show_site_name is enabled. It is used for alternate text for the logo by default.
------------------	---

## Navigation

Variable	Description
has_navigation	Returns whether navigation exists (i.e. at least one page exists)
nav_items	Returns the current pages as list
nav_css_class	Returns a string of the current classes applied to the page's navigation

## My Sites

Variable	Description
show_my_sites	Returns whether the current user has a My Sites page
show_my_places	Returns whether the current user has a My Sites page
my_sites_text	Returns the "my-sites" language key in the current user's locale
my_places_text	Returns whether the current user has a My Sites page

## Includes

Variable	Description
dir_include	Returns "/html"
body_bottom_include	Returns "\$/common/themes/body_bottom.jsp"
body_top_include	Returns "\$/common/themes/body_top.jsp"
bottom_include	Returns "\$/common/themes/bottom.jsp"
top_head_include	Returns "\$/common/themes/top_head.jsp"
top_messages_include	Returns "\$/common/themes/top_messages.jsp"

## Date

Variable	Description
date	Gives access to the dateUtil Java Object and all its methods
current_time	Returns the current time
the_year	Returns the current year



## 9. How to get a variable present in a portlet into theme?

- Theme can access the request and thus any query parameter given to the page. \$request will give access to the request... thus you can do \$request.getParameter("parameter").
- Note that the parameter will not be namespace so you should make sure that you don't namespace it also in your form.

## 10. What is custom attributes/fields

- It is the way to add additional attributes/fields to the existed portal entities so that in our real requirement if any fields you want then you can add.

## 12. What is structure and template?

- Structure will decide what kind of data we are going to use and template will decide look and feel for data.

## 13. What is singleton and can you name singleton classes in Liferay

- A class which we can create one and only one instance of class is called singleton.
  - To make class as a singleton we need to make following three things
  - Constructor of that class must be private
  - Create one private static variable of type that class only
  - Create one public static method which will actually create instance of that particular class and assign to this static variable.
  - And the following are singleton classes in Liferay (As per my observations)
    - SessionContext
    - LiferayServerContext
- 
- What is the similarities between Servlet and Portlet
    - Both are build on java technology
    - Both are managed by containers (Portlet container and Servlet container)
    - Client/server model
    - Life cycle is managed by container

## **20. What is the difference between XXXLocalService and XXXService**

- LocalService: When we call methods from LocalService classes there is no security check applicable or there is no permission checking available.
- Service: But here there is some security and permission checking applicable when we call methods from these classes. If user do not have permission then it throws Principle Exception

## **21. What is the use of XXXUtil class? :**

- All finder methods will be available in this class. By using this class only we can call finder methods in the XXXLocalServiceImpl or XXXServiceImpl.

## **22. What are the steps to create a custom method?**

- Implement Custom method in XXXLocalServiceImpl class
- Run service builder
- Call implemented method using XXXLocalServiceUtil

## **23. What will happen if i call finder method in Jsp or Controller Class by using XXXUtil class?**

- We will get Exception like as follows
- org.hibernate.HibernateException: No Hibernate Session bound to thread

## **What are the files will be generated when you run service builder?**

- Persistence Layer
- Service Layer
- Model Layer
- WebServices if you put remote attribute as true
- Hibernate/Spring Config files

## **Which javascript method will be execute on each portlet ready?**

- Liferay.Portlet.ready(fn) : Executed after each portlet on the page has loaded.
- Aui.ready(fn) : Executed HTML in the page has finished loading
- Liferay.on(fn) : Executed everything ready.

**How to get UserId?(It seems to be simple but in interview you can't get idea suddenly some times)**

```
User currentUser;  
currentUser = themeDisplay.getUserId()  
currentUser = PortalUtil.getUser(request)  
currentUser = (User) request.getAttribute(WebKeys.USER)
```

**29. What is meant by instanceable-portlet?**

- Allow multiple instances on the page
- Have different preferences

**30. Liferay core portlets main configuration defined in which file?**

- portlet-custom.xml

**31. What developer mode do?**

- Doesn't minify CSS and JS
- Caching disabled for Templates and webcontents
- Server does not launch browser when startup
- For above all respective properties are there please go through in detail.

**32. By which type of hook you can modify the method of service(UserLocalService)?**

- Service Wrapper hook

**33. Which classes will be create when we run service build?**

- Persistence:
- XXXPersistence
- XXXPersistenceImpl
- XXXUtil

**Service:**

- XXXLocalService
- XXXLocalServiceImpl
- XXXLocalServiceBaseImpl

- XXXLocalServiceUtil
- XXXLocalServiceWrapper

if you put remote= -service is true then

- XXXServiceSoap
- XXXSoap

Model:

- XXXModel
- XXXModelImpl
- XXX=
- XXXImpl
- XXXWrapper <= /span>

34. Which class you will write logic for modify= ing the database column width only once?

- GlobalStartupAction

35. What is the difference between application.= startup.events and global.startup.events?

- global.startup.ev= ents - Run once for global server
- application.staru= p.events - Run for every portal instance at startup

36. Which class Liferay MVC portlet extends?

- LiferayPortlet

37. Which class JSR-286 portlet = extends?

- GenericPortlet

38. How you will enable developer mode in Lifer= ay?

- By adding portal-developer.properties file to application server's configuration file. =

39. How can you generate remote = services using liferay service builder?

- By enabling remot= e-service=true in service.xml

<https://www.surekhatech.com/blog/kaleo-workflow-in-liferay>

### What is workflow?

- **Sequence Of Small Task:-** Workflow is a sequence of small tasks.
- Each task will be processed by different people.
- In the end, all tasks join together and will complete the big task.

### Introduction of workflow in Liferay.

- Liferay DXP provides the workflow engine called Kaleo.
- Kaleo workflow allows a user to define simple to complex business processes/workflows. In Liferay CE, workflow is written in an XML document and if you are Liferay EE customer then they provide a graphical view to create a workflow.

**Kaleo workflow custom entity** <https://www.javasavvy.com/liferay-dxp-custom-entity-kaleo-workflow-tutorial/>

### What Is a Liferay Portal?

**Single point of access:** - In general, a web portal represents a web application that provides a single point of access to variety of information.

**Portlet Custom application which can run inside portal:** - Functionality of a portal can extended very easily by creating custom applications that can run inside a portal. These applications are called portlets.

**Application server that can run a portlet container spec:** - For running portlets you need application server that can run a portlet container spec (JSR168 or JSR286). There are dozens of free and enterprise portal products available.

### Benefits of portal

- Content management system (Documents, Articles).
- 2-stage,3-stage workflows and publishing.
- Social networking. 4) Sites and organization basis pages.
- Integration with third party like LDAP, Solr, SSO (CAS, SiteMinder).
- Portlets related to Wiki's, News, Finance, Social, Workflow, Collaboration.
- Compatible with different Operating systems, databases and APP servers.

- Compatible with UI technologies and web services.
- Compatible with different scripts like PHP, Ruby, Grails, Python.
- Support of hot deployment feature.

## **Q.2) What is a Portlet?**

**Ans)** Liferay portal comes with different portlets by default. These portlets are nothing but different small applications running in the instance (portal). Portlets are pluggable software components that are managed in a web portal.

## **) is Liferay is JSR-168 compliant or JSR-286?**

**Ans)** JSR – 286

## **Q.7) What class does LiferayMVCPortlet extend?**

**Ans)** LiferayPortlet -> GenericPortlet

## **What is service.xml and how will you use the Liferay service builder?**

- **Database Entities:** - service.xml is a configuration file that consists of the details for database entities that are used in a portlet.
- **Service.xml :-** The service builder uses the service.xml to generate the DAO/DTO classes for a portlet.

## **What Is the Difference Between View Mode and Edit Mode of Portlets?**

- **Type Of Operation:** - The modes of a portlet are typically related to the types of operations a user can perform on the portlet.
- **View Read Only:** - A view mode is where users are allowed to do read only operations,
- **Edit Add/Update:** - edit mode user can add/update the data in the portlet.

## **Each portlet mode has a method which handles it:**

- doView()
- doEdit()
- doHelp()

These methods will contain mainly our "traffic directing" logic, which are responsible to decide what view the user wants to see.

Once the view has been determined, we will forward on to methods which actually implement that view.

### What Is a Liferay Theme? What Are the Advantages of Using Them?

- **Look and Feel:** - Liferay themes are the way to customize look and feel of a liferay portal page.
- **Extensible Component:** - The themes are extendable components
- **Deploy Separately:** - Theme can be deployed separately on a portal
- **Separate Theme for Each Page:** - Each page on the portal can choose a different theme (if required).
- **Multiple Website for Single Portal:** - Themes are really powerful way to run multiple websites from a single portal. You can host multiple websites on a single liferay portal with totally different look and feel. This allows you to be unique at the same time save a lot resources on website hosting.
- **Multiple Theme Can be Deployed:** - Liferay provides the feature to deploy the different themes into the portal same like portlets war files. So, number of themes we can be deployed into Liferay portal and we can further select those themes for user interface as per the requirement of end user for different site or organization pages.
- Combination of Free marker Template ,CSS and JS

### What is meant by Liferay?

- **Web Application:** - Portal is platform to create web applications
- **Common Feature:** - portal provides commonly used features such as user management, authentication, sites, web content management to avoid creating from scratch.
- **Multi Portlet on Page:** - Portal assembles multiple portlet applications to form a page

### JSR 168 and JSR 268 Difference

Topic	JSR 168	JSR 268
Specification	1.0	2.0
IPC	Does not support IPC	Support IPC
Serve Resource	No support of serveResource() method	Support Ajax and serveResource() method
Filters	No Support of Portlet filters	Support Portlet Filters
Namespace		GetNameSpace() Method

## Portlet and Servlet

Topic	Portlet	Servlet
Render	Part of Html	Complete Html
Container	Portlet Container	Servlet Container
Type Of Request	Has Three Type of Request Render, Action, Resource	Has Single type of action called: HttpServletRequest
Windows States and Modes	Portlets can define window states and modes/td>	Servlet can not define window states and modes
Access	Can't Access Through URL	Can Access Through URL

## How to Add Portlet in Theme?

- `$theme.runtime($portletId, $queryString, $velocityPortletPreferences.toString())`
- `<@liferay_portlet["runtime"]`  
`defaultPreferences="{freeMarkerPortletPreferences}"`  
`portletProviderAction=portletProviderAction.VIEW`  
`portletName="com_custom_portlet_themesearch_ThemeSearchPortlet" />`

## What is portlet namespace?

**Uniquely Identify Each Portlet** namespace is used to uniquely identify each portlet. Its unique identifier associated to each portlet by portal. **Suppose**

- As you already know you can put many portlets on the same page in Liferay.
- Let if there are 2 portlets on the same page and both have an input box with a same name like username what will happen?
- Answer is Name conflict. To avoid this name conflict, we use portlet name space.
- Portlet Name Space will give unique name to the elements in the page and these elements are associated to respective portlet. This will avoid name conflicts in the page
- 

## Database sharding

- Liferay Sharding allows you to split up the database by portal instance only.



- Database sharding allows system more scalable.

#### What is Sharding?

- Sharding is used to split up the database by filtering data. If database/table is loaded with huge data then system will get slow.
- In such case, sharding technique is used to distribute the data to different database and select the data based on that
- For example, some implementations of sharding a database split up the users: User names beginning with A to D go in one database; E to I go in another database;

#### How Liferay implemented Sharding?

- Liferay allows you to support sharding through different portal instances and we can create separate database for each portal instance
- default algorithm use round robin shard selector.
- As we are creating separate database for each portal instance and it makes system more scalable

#### How To Support Internationalization (i18n) On Liferay Portlets? How Do You Support Multiple Languages Without Writing Java Code In Liferay?

locales=en\_US,zh\_CN,zh\_TW,ja\_JP,ko\_KR

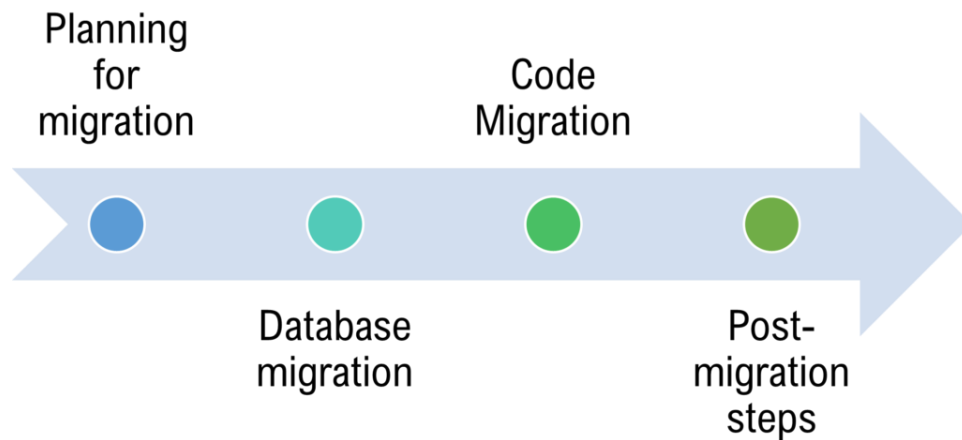
- Language\_en\_US.properties
- Language\_zh\_CN.properties
- Language\_zh\_TW.properties

#### How Will You Configure A Portlet That Can Be Added Multiple Times On A Portal Page?

Instanciable = true

session.store.password=true

Migration steps <https://aixtor.com/liferay-migration-from-6-2-to-liferay-dxp-7/>



### 1. Prerequisite for migration

**Latest Patch and Service Pack:** - Make sure that Liferay is patched with the most recent service pack.

**Test Environment:** - Create a test environment and use copies of the database and server.

**Clean Unnecessary Data:** - Remove unnecessary instance, web content, sites, Liferay provide groovy script, data cleanup, old version of journal article and document, inactive users and its data, unwanted data layout

### 2. Database migration

In this step, we will migrate our database from Liferay 6.2 to Liferay 7.x

- Tune database as it executes many update statements.
- Deactivate the data integrity
- Turn off transaction logging
- Increase the interval to flush commit

Liferay provide database migration tool.

`indexReadOnly="true"`

### 3. Liferay Code Migration

- code migration tool in Liferay IDE.

- These are some problems that cannot be auto corrected you have to update them manually according to the breaking changes documentation.
- Upgrade theme ,ext , service builder.
- Manual JSP hook migration

#### **4. Post Migration Steps**

- indexReadOnly="false"
- Regenerate the search index  
delete license folder
- Tune Liferay memory setenv.bat,setenv.sh
- Deploy license in deploy folder

#### **5. Common issues/error Occurred after Migration**

- NullPointerException at ThemeServicePreAction –
- PwdEncryptorException: invalid keyLength value – This exception will occur when the specified password-encryption algorithm is incorrect. We need to specify the correct encryption algorithm in portal-ext.properties i.e passwords.encryption.algorithm.legacy=.  
NullPointerException
- while upgrading DDM – When there is any orphaned template is there, and you run migration at that time, this exception will occur. To resolve this issue, delete the orphaned template.

Liferay Modules	Liferay Plugins
Introduced in Liferay 7	Introduced since Liferay Started. From Liferay 4.x to Liferay 7/DXP
Liferay Modules are based on OSGi framework	Liferay Plugins based traditional Java web application framework. Like we use more J2EE web application Frameworks.
Liferay Module will run in OSGi Run Time Environment.	Liferay Plugins run in Traditional Application Servers such as Tomcat, JBoss , Glassfish
Liferay Module we can call it as OSGi bundles and packaged as JAR file.	Liferay Plugins are web applications and packaged as WAR files.
Liferay Module provides the dynamic component injection with help OSGi so that we can make most of the features as independent components and can easily inject into the other components when it required.	Liferay plugins are like independent web applications we cannot share anything between two web applications. It is more coupled. If we want share some classes we have to put these jar files in some global class loaders then only other web application can use that classes.
We can avoid all the problems of class loader issues.	We can see more class loader issues.
No Class Definition Found, Class Not Found Run time errors completely vanished in the runtime.	We frequently experience No Class Definition Found, Class Not Found Run time errors in the runtime.

<p>Liferay Modules have different types like Portlet Modules, Hook Modules.</p> <p>Themes and Layouts partially support modularity.</p>	<p>Liferay plugins have different types such as portlets, themes, ext, layout and hooks.</p>
<p>Liferay Modules developed based on OSGi Framework and it used the several framework to develop modules such as Declarative Services (DS), Apache Blueprint and OSGi API.</p>	<p>Liferay Plugins uses the J2EE web application frameworks such as Struts, Spring, Liferay MVC, JSF and other web application frameworks.</p>
<p>Liferay Modules followed 3-tier architecture application development such as Persistence, Service and Presentation layered architecture.</p>	<p>Liferay Plugins uses the same 3-tier architecture application development such as Persistence, Service and Presentation layers architecture.</p>
<p>Usually Liferay Module development we separated each Layer as Independent Modules</p> <p>Persistence Tier Module</p> <p>Service Tier Module</p> <p>Web Tier Module</p>	<p>Liferay Plugins Development all 3 tiers packaged as single WAR file.</p>
<p>Liferay Modules support the dynamic deployment feature and we can stop and start module without un-deploy the module.</p>	<p>Liferay Plugins we must un-deploy the war file from server to remove features.</p>
<p>Liferay Modules uses the Apache Gogo shell Command Line Interface (CLI) to manage the Liferay Modules in the OSGi run time environment. Like stop, start, activate and deactivate.</p>	<p>Liferay plugins does not have any Command Line Interface to un-deploy the bundles and we have to removed plugin war files from servers.</p>

Liferay Portal Admin screen also have to manage these Liferay modules.	<p>Some Enterprise Application Servers have administrative feature to do these things.</p> <p>Liferay Portal Admin screen also have option to manage these Liferay plugins.</p>
Liferay modules are available in Liferay Portal Server Bundle inside <b>OSGI</b> directory once the module is deployed.	All Liferay Plugins are available in Liferay Portal Server webapps directory.
Liferay Module have important configuration file <b>META-INF</b> which is like module deployment configuration file have more information used by OSGi run time environment.	Liferay Plugins have web.xml file and used by Portlet/Servlet containers.
Liferay portlet module all xml configuration files information turn into <b>Declarative Services (DS) Component properties</b> . Like portlet.xml, liferay-portlet.xml and liferay-display.xml files tags and values.	Liferay Plugins used the XML configuration such as portlet.xml, liferay-portlet.xml and liferay-display.xml
If the Liferay Application have more features then we divided into multiple independent modules and these modules can communicate each other very easy in OSGi environment.	Modularity is not that easy in Liferay Plugins development so we have to keep all the files in single WAR file.

Liferay Module introduced new tool called <b>BLADE CLI</b> to develop Liferay Modules.	Liferay Plugins have Plugins SDK to develop Liferay Applications.
Liferay Modules have several tools such as Liferay IDE, Liferay Studio, BLADE CLI	Liferay Plugins have tools such as Liferay IDE, Liferay Studio.
Liferay Module support GRADLE and MAVEN build environment.	Liferay Plugins support ANT and MAVEN build environment.
Liferay Modules used the Liferay Workspace concept to manage the application and its configuration across the platforms.	Liferay Plugins uses the Plugins SDK to develop and manage the plugins.
Liferay Modules used the OSGi API, Declarative Services and Apache Blueprint framework. So we have limited choice to use popular we application framework like Spring. Liferay still not fully supporting Spring Modules, it means we have to use the WAR deployment.	We can use any J2EE popular web application frameworks such as Spring, Struts and JSF.
Liferay Modules provides the Micro services Architecture based applications.	To bring Micro services Architecture we have to depend on other frameworks such as Spring Micro Services.
Liferay Modules have dynamic component development feature so we can easily turn on	Liferay Plugins not like Dynamic Component Development so bunch of feature in one plugin

and off the feature at any point of time with minimal effort.	application. If we remove one plugin, it can turn off many features.
Liferay Modules can provide easy way deployment process and we can move easily from one environment to other with minimal effort. Liferay Workspace provided this capability.	Liferay Plugin Deployment process and moving from one environment to other tedious. We do not have any special mechanism like Liferay Workspace.
Liferay Modules uses the Bndtools to develop and build the modules. We have BLADE CLI, Liferay IDE, GRADLE and MAVEN are used the Bndtools internally.	Liferay Plugins follow the J2EE web application development so MAVEN Liferay Plugin or Liferay IDE handle the Development and Deployment.

**What are the different module types supported in Liferay 7?**



```
E:\workspaces\my-liferay-workspace\modules>blade create -l
activator
api
content-targeting-report
content-targeting-rule
content-targeting-tracking-action
control-menu-entry
fragment
mvc-portlet
panel-app
portlet
portlet-configuration-icon
portlet-provider
portlet-toolbar-contributor
service
service-builder
service-wrapper
simulation-panel-entry
template-context-contributor
theme
```

### **What is IPC, What are the types of IPC? Explain Different IPC's In detail?**

Inter portlet communication is a mechanism to communicate between the portlets. This is a new feature brought up in JSR 286 (Java Specification Request). There are four different ways of IPC, let's discuss each one of them one by one,

- **Portlet Sessions IPC**
- **Public Render Parameters IPC**
- **IPC Events**
- **Client Side IPC with AJAX**
- **Client Side IPC with Cookies**

#### **PortletSessions IPC**

- From JSR 168/Portlet 1.0 we have feature of Inter Portlet Communication with Portlet Sessions.
- Each Portlet have its own session that is called Portlet Session and session data won't be shared with other Portlets.
- To make Inter Portlet Communication with help of Portlet session we will make Portlet session as public so that data will be shared in other Portlets.

[http://www.liferay savvy.com/2015/04/liferay-inter-portlet-communication\\_15.html](http://www.liferay savvy.com/2015/04/liferay-inter-portlet-communication_15.html)

- Available from JSR 168 and carried to JSR 286
- Communication can be possible with in the page and across the portal

### PublicRender Parameters IPC

- Public Render Parameters are the one of the ways to make Inter Portlet Communication among the Portlets and these Portlets may be in same page or different pages in the portal.
- Public Render Parameters we will define specific parameter in the Portlet with help of that data will be carried from one Portlet to other Portlet and these Public Render Parameters accessible by specific Portlet where we already defined these parameters.
- IPC we have two categories of Portlets Sender Portlet and Receiver Portlets. Sender Portlet will send some data or push some data then receiver Portlet will receive the data and will process subsequent steps.

Available from JSR 286

Communication can be possible with in the page and across the portal

<http://www.liferay savvy.com/2015/04/liferay-inter-portlet-communication.html>

### IPCEvents

- IPC Events is one of the ways to make Inter Portlet Communication among the Portlets and these Portlets may be in same page or different pages in the portal.
- IPC Events we have two categories of Portlets Event Producer Portlet and Event Listener Portlets. Producer Portlet will send some data or push some data then Listener Portlet will receive the data and will process subsequent steps.
- We can have one Producer and Multiple Consumer Portlet will listen the event and process subsequent steps.
- Available from JSR 286
- Communication can be possible with in the page and across the portal

[http://www.liferay savvy.com/2015/04/liferay-inter-portlet-communication\\_16.html](http://www.liferay savvy.com/2015/04/liferay-inter-portlet-communication_16.html)

### Client Side IPC with AJAX

- Client Side Inter Portlet Communication is one of the ways in IPC this can apply the Portlet which reside in same page. Because Client side IPC will uses java script to communicate that's why all Portlet which participated in IPC need to be in same page.
- Liferay already have some java script library to implement client side IPC in Portlet Development.
- Available from JSR 286
- Communication can be possible with in the page only

<http://www.liferay savvy.com/2014/01/liferay-client-side-inter-portlet.html>

### Client Side IPC with Cookies

- Same Like Client Side IPC with AJAX we can also achieve IPC through browser cookies and data will be stored in cookies and this will be accessed by other portlets and this have some limitation with respect data storage and security wise means I can store very limited data and sometime browsers may disable cookies then communication will be failed among Portlets.

Available from JSR 286

- Communication can be possible with in the page only

### Suggestible priority order as per my knowledge

- *IPC Events*
- *Public Render Parameters IPC*
- *Portlet Sessions IPC*
- *Client Side IPC with AJAX*
- *Client Side IPC with Cookies*

Explain The Permission System in Liferay ?

- Page Level.
- Portlet Level.
- Model Resource level.

#### Page Level Permissions :

At page level, we can define the permissions(VIEW,UPDATE,PERMISSIONS etc) for a particular page on a particular role(Guest,Organization Member etc)

- Resource Action.
- Resource Permission.

**primaryKey** -> It is the current PageId.

**RoleId** -> It is nothing but roles like Guest (10139),Organization Member(10147),Owner (10140).

**ActionIds** -> 1(VIEW), 3 (VIEW and ADD\_DISCUSSION), 127 (All Permissions).

#### Portlet Level Permissions :

- At portlet level, we can define the permissions(Add to Page,View,Configuration etc) of a portlet on a particular role(Guest,Organization Member etc).

#### Model Resource(Calendar Event, Manage Pages,Manage Teams) Level Permissions :

- At Model Resource(Calendar Event,Manage teams etc) level, we can define the permissions(View,Add Discussion,Permissions etc) for a model resource (Calendar Event)on a particular role(Guest,Organization Member etc).
- At portlet level, we can define the permissions(Add to Page,View,Configuration etc) of a portlet on a particular role(Guest,Organization Member etc).

**What is the difference between Site and an Organization?**

- Organization is similar to site but it can used to only group the people and manage them. In organization also we can have pages, web contents and users. In the organization users only can be members. In the organization we can create users and assign users to specific organization by admin

## What are the types of Roles available in Liferay?

- [Regular Role Or Portal Role](#)

When these roles are assigned to the user, the role's permissions are available to the User across the Portal Instance, the sub hierarchies such as Sites, Organization and User Groups.

- [Site Role](#)

When a Site Role is assigned to an Site User, The permissions of the Role to the User Will be applicable only inside the Site and will not applicable for the User in other sites.

- [Organization Role](#)

When an Organization Role is assigned to an Organization User, The permissions of the Role to the User Will be applicable only inside the Organization and will not applicable for the User in other Organizations.

## Regular Roles Available In Liferay

- [Guest](#)

Guest User will be having very minimal set of permissions in the portal and are non authenticated users

- [Owner](#)

This is can be applied to the User who is the owner of certain entities in Liferay

- [Administrator](#)

Administrator User can access almost anything in the portal and any action in the portal and can access Sites organization and any object without any constarints

- [User](#)

This is the basic Role Given to an User when the user is authenticated into the Portal.

- [Power User](#)

This Role have the same permissions as that of the User Role, we can use this Role Just to distinguish the normal user from this previllaged role and give more access.

### Site Roles Available In Liferay

- **Site Administrator**

It enables the Site Administrator User the ability to manage different assets of the sites such Site Pages, Site Membership, Site Contents and Site Settings.

- **Site Member**

This Role grants permissions to visit the privates pages of the sites and its contents including public pages

- **Site Owner**

This Role is same as that of the Site Administrator Role except that it provides the ability to manage the different aspects of the sites such as deleting the site membership or roles including Site Administrator

### Organization Roles Available in Liferay

#### Organization Administrator

It enables the Organization Administrator User the ability to manage different assets of the Organization such Site Pages, Organization Membership, Organization Site Contents and Organization Site Settings.

#### **Organization Owner**

This Role is same as that of the Organization Administrator Role except that it provides the ability to manage the different aspects of the Organization such as deleting the Organization membership or roles including Organization Administrator

#### **Organization User**

This Role grants user the ability to visit the Organization Site Pages when the user belongs to the Organization

Apart from all the above roles, we can create our own Regular, Site and Organization Roles with our own permissions in the control panel.

## What are the New Features Available in Liferay 7?

### Liferay 7 Features

- **OSGI AND MODULARITY**

- **OSGi**

- Any framework that implements the OSGi standard provides an environment for the modularization of applications into smaller bundles. Each bundle is a tightly coupled, dynamically loadable collection of classes, jars, and configuration files that explicitly declare their external dependencies (if any).

- **MODULARITY**

- **Example of modularity:** You can plug any number of available devices into a computer and they work because they are designed to integrate with the system in a standard way. It doesn't matter who manufactured the computer, because they are all made up of inter-operating modules that individually implement specific functionality, but when put together comprise an entire working system.
    - **Benefits of modularity:**
      - The framework is lightweight, fast, and secure.
      - The framework uses the OSGi standard. If you have experience using this with another project, you can apply your existing knowledge to Liferay.
      - Modules publish services to and consume services from a service registry. Service contracts are loosely coupled from service providers and consumers.
      - Modules export code at the package level for use by other modules, enabling the registry to manage dependencies automatically.
      - The container manages module lifecycles dynamically. Modules can be installed, started, updated, stopped, and uninstalled while Liferay is running, making deployment a snap.
      - Modules hide their internals from other modules, giving developers free reign to change the internals whenever needed.
      - Modules are semantically versioned and declare dependencies on specific versions of other modules. This allows two applications that depend on different versions of the same library to each depend on their own versions of the library.
      - You can use your existing developer tools and environment to develop modules.

- **TOOLING**

- Liferay is very flexible when it comes to supporting different development tooling. Instead of being pigeonholed into using a specific tool, Liferay allows you to choose whatever tool you feel most comfortable with, and use it to develop for your Liferay instance.
- **BLADE CLI**
  - The Blade CLI is the easiest way for Liferay developers to create new Liferay modules. Although the Plugins SDK is also supported, Blade CLI lets you create projects that can be used with any IDE or development environment.
  - Blade CLI is a command line tool bootstrapped on to a Gradle based environment that is used to build Liferay 7.0 modules. This tool set provides a host of sub-commands that help Liferay developers create and deploy modules to a Liferay instance.
- **LIFERAY WORKSPACE**
  - A Liferay Workspace is a generated environment that is built to hold and manage your Liferay projects.
  - This workspace is intended to aid in the management of Liferay projects by providing various Gradle build scripts and configured properties. This is the official way to create Liferay 7.0 modules using Gradle.
- **Internationalization**
  - Automatically generating language files
    - If you already have a default Language.properties file that holds language keys for your user interface messages, or even a language module that holds these keys, you're in the right place.
    - Instead of manually creating a language properties file for each locale that's supported by Liferay, you can get them all **automatically generated for you with one command**. The same command also propagates the keys from the default language file to all of the translation files.
    - You can take a few additional steps and get automatic translations using **Microsoft's Translator API**.
- **WYSIWYG Editors**
  - **Adding new behavior to an editor**
  - Liferay provides a way to programatically access the editor instance to create the editor experience you want.
  - This can be done by using the **liferay-util:dynamic-include JavaScript** extension point. This allows anyone to inject JavaScript code right after the editor instantiation to configure/change the editor.
  - Liferay supports several different types of WYSIWYG editors, which include (among others):
    - AlloyEditor



- CKEditor
  - TinyMCE
- **Themes generator**
  - **Themes generator**
    - The Liferay Themes Generator is an easy-to-use command-line wizard that streamlines the theme creation process. It is independent of the Liferay Plugins SDK, and works for Liferay versions 6.2 and up.
  - **Themelets**
    - Themelets are small, extendable, and reusable pieces of code. Whereas themes require multiple components, a themelet only requires the files you wish to extend. This creates a more modular approach to theme design, that lends itself well to collaboration, and reduces the need for duplicated code in your theme.
    - Themelets allow developers to easily share code snippets across their themes and with other developers. A themelet can consist of CSS, JavaScript, and templates, just like a theme.
- **Application Display Templates Support for Login, Language and Breadcrumb portlets.**
  - ADT is great way of customize the look and feel of existed portlets in Liferay through Free Marker Templates. In Liferay 6.2 have given ADT support for few portlets such as Asset Publisher, Blogs, Wikis, Navigation and Category Navigation and it added few more portlet in the list such as Login, Language and Breadcrumb portlets.
- **ElasticSearch**
  - Liferay before it's used the Lucene search engine to search content in Liferay and Liferay 7 integrated Elastic Search as default search.
- **Java 8 Support**
  - Liferay 8 completely computable with Java 8. Oracle have stopped updates for Java 7 and Java 6. So now Liferay 7 completely considered only Java 8 and they implemented Liferay 7 to run on Java 8.
  - Java 8 have many features so it will be valued added to Liferay Application development like Lambdas and Functional Interfaces, Default and Static Methods in interfaces ,Stream API .Date Time API Improvements for Time Zones and Nashorn, JavaScript Engine are really pretty cool features in Java 8. We cannot see java.lang.OutOfMemoryError: PermGen space in Liferay 7 any more.
- **Single Page Application (SPA) Development**

- Now trends is Single Page Application using MVC based java script libraries with awesome look and feel with bootstrap. Liferay 7 have bring support for SPA application using SennaJs. Now all portlet become as Single Page Application so that user can navigate to anywhere without reloading entire portal page.
- **Service Builder code now uses Declarative Services instead of Spring for dependency injection**
  - Declarative Services is one of runtime service provided by OSGi container to specify dependency services to other components and it make available these services to consumer bundle/component.
  - These dependency services declared as XML configuration so that when we deploy bundle into OSGi container then its resolves the all the dependency services otherwise bundle won't be activate and its responsibility taken care by Service Component Runtime, we can use either Equinox Declarative Service or Apache Felix SCR to provide feature in OSGi container. Liferay 7 have used default OSGi Declarative Service model than Spring dependency. Generally Declarative Service model service will be available or activates when it started using otherwise it won't be start but Spring Decency, the spring container will prepare the object based on XML configuration it will inject all objects and make it available complex Object in the spring container even we use it or not.
- **ECMAScript 2015 support for JavaScript lovers**
  - For those of you who like and follow front-end development trends, we're certain that you've heard about [ECMAScript 2015](#); which brings many significant improvements to JavaScript. We are pleased to announce that, now in Liferay 7, you will be able to:
    - Write ES2015 code which runs on IE9+, Safari, Chrome and Firefox (thanks to Babel).
    - Install 3rd party modules via Bower and use them in your Portlets from both JSP and JS files.
  - If you want more details to explore all the possibilities, don't miss Iliyan Peychev's blog entry [Introducing ECMAScript 2015 support in Liferay Portal 7](#)—it contains a video demo!!!
- **Bootstrap 3.0 Look and Feel**
  - Liferay 6.2 have used Bootstrap 2.x and Now Liferay 7 have used Bootstrap 3.0 so that look and feel for portal and its applications are really going to be good. All Liferay AUI components have used Bootstrap 2.x before and using Bootstrap 3.0 in Liferay 6.2 is really challenging and once we use bootstrap 3.0 UI developer need to put lot effort to adjust the things in theme level or AUI components level.
  - Now with bootstrap 3.0 all Liferay AUI components already used it and it won't be problem to create themes with bootstrap 3.0.

## 1. Liferay DXP Forms

- Splitting a form into multiple pages so that you can create wizard- like forms
- Ability to control the layout of the form including organizing form fields into multiple columns on a given form page. This can also support complex layouts wherein each row can have a different number of columns.
- Ability to populate a form dropdown (i.e. a Select field) with a REST Data

### **What are the types of Pages in Liferay?**

- Public Pages
- Private Pages
- Widget Pages
- Content Pages

### **What are the Implicit Objects available in Liferay's Portlet JSP?**

Implicit Objects From Portlet:defineObjects Taglib

**To Get all these implicit objects, <portlet:defineObjects /> must be added In the JSP, these objects holds the information about the portlet parameters and related data.**

- actionRequest
- actionResponse
- eventRequest
- eventResponse
- liferayPortletRequest
- liferayPortletResponse
- portletConfig
- portletName
- portletPreferences
- portletPreferencesValues
- portletSession
- portletSessionScope
- renderRequest
- renderResponse
- resourceRequest
- resourceResponse
- searchContainerReference

#### **Implicit Objects From Liferay-Theme:defineObjects Taglib**

**To Get all these implicit objects, <liferay-theme:defineObjects /> must be added In the JSP, these objects holds various information related user, portlet, layout, theme, portal, permissions etc.**

- account
- colorScheme
- company
- contact
- layout
- layouts
- layoutTypePortlet
- locale
- permissionChecker
- plid
- portletDisplay
- portletGroupId
- realUser

- scopeGroupId
- theme
- themeDisplay
- timeZone
- user

### Implicit Objects From JSP

These are the default implicit objects available from the JSP, you don't have to include any taglib for them

- application
- config
- out
- page
- pageContext
- request
- response
- session

### How to Configure EH Cache In Liferay 7?

Plugin cache reconfiguration allows:

- Reconfigure any cache (Single VM, Multi VM, and Hibernate)
- Creating new caches, including caches for ServiceBuilder services introduced in a plugin
- This capability can be leveraged in any plugin.

To use in a plugin you simply need to add a portlet.properties file to your plugin that defines the same properties that are used in the core portal to define cache configuration file location:

- net.sf.ehcache.configurationResourceName
- ehcache.single.vm.config.location
- ehcache.multi.vm.config.location

### How to Configure CDN In Liferay 7, What are its uses?

**CDN:-** A Content Delivery Network (CDN) is an interconnected system of servers deployed in multiple data centers that use geographical proximity as a criteria to deliver content across the Internet

### Using a CDN

A Content Delivery Network (CDN) is a distributed infrastructure for caching static, file-based content. When visitors hit your site, your static content is loaded from the server closest to them, speeding up requests.

The screenshot shows the 'Instance Settings' page in Liferay, specifically the 'Configuration' tab. Under the 'General' section, the 'Main Configuration' fields are visible: 'Name' (Liferay), 'Mail Domain' (liferay.com), and 'Virtual Host' (localhost). To the right of these fields, a red box highlights the 'CDN Host HTTP' and 'CDN Host HTTPS' input fields, along with the 'Enable CDN Dynamic Resources' checkbox, which is checked. Below the 'General' section, the 'Navigation' section is partially visible, showing 'Home URI' and 'Default Logout Page'.

### How to Configure Domain Name (Virtual Host) In Liferay 7, What are its uses?

#### Running multiple site

- Company1.example.com
- Company2.example.com
- One possible use case is to have two different access points to access the same database. The Liferay platform implements name-based virtual hosts at two different levels: (1) through the Virtual Instance settings (formerly *Portal Instances* for those upgrading from legacy Portal) and (2) at the site level.

4. There are four fields and a toggle box:

- **Web ID:** A general convention is to use the domain name for this; it is a user-generated ID for the instance.
- **Virtual Host:** Put the domain name you configured in the network here. When users are directed to your Liferay server via this domain name, the Liferay platform will then be able to send them to the proper virtual instance.
- **Mail Domain:** Enter the domain name for the mail host for this instance. The Liferay platform will use this to send email notifications from the instance.
- **Max Users:** Enter the maximum numbers of user accounts for the virtual instance to support. The default '0' means an infinite number, not zero.

**What are the Differences between Liferay 7 CE(Community Edition) and Liferay 7 DXP(Enterprise Edition)?**

Liferay DXP	Liferay 7
Enterprise Edition	Community Edition
100 + Portlets (workflow , web forms and system integrations)	70+ Out Of the Box Portlets
Liferay Developer Studio	Eclipse IDE
Professional Support 24/7 Phone / WEB	No Professional Support
Patches and Updates Alerts	No Updates
Liferay Sync – One Site	Liferay Sync- Multi Sites
Audience Targeting Feature – Specific content / Campaigns to Specific Users	– Not Available-
Support for Enterprise Databases (Oracle Database,Microsoft SQL Server,IBM DB2,Sybase DBs)	No Support for Enterprise Databases
Elastic Search Engine ( Lucene Replaced)	Elastic Search Engine ( Lucene Replaced)
Liferay Subscription Apps ( Market Place )	Liferay Free Apps ( Market Place)
Auditing Features	– Not Available-
Performance Monitoring Feature	– Not Available-
Enhanced Cache Replication	-Not Available-
Workflow Designer	-Not Available-

### What is IndexPostProcessor In Liferay 7?

- **Create liferay plugin project of type Hook.**
- **Step 2: Open liferay-hook.xml and add the below content.**

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

<!DOCTYPE hook PUBLIC "-//Liferay//DTD Hook 6.2.0//EN" "http://www.liferay.com/dtd/liferay-hook_6_2_0.dtd">

<hook>

<indexer-post-processor>

<indexer-class-name>com.liferay.portal.model.User</indexer-class-name>

<indexer-post-processor-impl>com.proliferay.demo.MyCustomPostIndexer</indexer-post-processor-impl>

</indexer-post-processor>

</hook>
```

**Step 3 public class MyCustomPostIndexer extends BaseIndexerPostProcessor {**

### What is ActionCommand, ResourceCommand and RenderCommand In Liferay 7?

```
@Component(
    immediate = true,
    property = {
        "javax.portlet.name=" + HelloWorldPortletKeys.HELLO_WORLD,
        "mvc.command.name=/hello/edit_entry"
    },
    service = MVCRenderCommand.class
)

public class EditEntryMVCRenderCommand implements MVCRenderCommand {
```

### What is Portlet Filter in Liferay 7, How to Implement it?



Portlet Filter is used to intercept and manipulate the request and response before it is delivered to the portlet at given life cycle such as action, render and resource

#### What Is Portlet Filter?

Portlet Filter is used to intercept and manipulate the request and response before it is delivered to the portlet at given life cycle such as action, render and resource

#### Types Of Filters For A Liferay Portlet

Action Filter

Render Filter

Resource Filter

Base Filter

#### Which Filter To Use?

**Action Filter** : To Intercept only The Action Requests

Render Filter : To Intercept only The Render Requests

Resource Filter : To Intercept only The Resource Requests

Base Filter : To Intercept all the http request comes to your portlet.

#### How to implement it?

- Create A Portlet Filter Component Class

```
@Component(immediate = true,  
property = {  
"javax.portlet.name="+MyPortletKeys.My,  
},  
service = PortletFilter.class  
)
```

```
public class MyRenderFilter implements RenderFilter{
```

#### What is Class Loader ? What are the types of ClassLoaders available in Liferay 7?

Here are the classloader descriptions:

- **Bootstrap:** The JRE's classes (from packages java.\*) and Java extension classes (from \$JAVA\_HOME/lib/ext). No matter the context, loading all java.\* classes is delegated to the Bootstrap classloader.
- **System:** Classes configured on the CLASSPATH and or passed in via the application server's Java classpath (-cp or -classpath) parameter.
- **Common:** Classes accessible globally to web applications on the application server.
- **Web Application:** Classes in the application's WEB-INF/classes folder and WEB-INF/lib/\*.jar.
- **Module Framework:** Liferay DXP's OSGi module framework classloader which is used to provide controlled isolation for the module framework bundles.
- **bundle:** Classes from a bundle's packages or from packages other bundles export.
- **JSP:** A classloader that aggregates the following bundle and classloaders:
  - Bundle that contains the JSPs' classloader
  - JSP servlet bundle's classloader
  - Javax Expression Language (EL) implementation bundle's classloader
  - Javax JSTL implementation bundle's classloader
- **Service Builder:** Service Builder classes

The classloader used depends on context. Classloading rules vary between application servers. Classloading in web applications and OSGi bundles differs too. In all contexts, however, the Bootstrap classloader loads classes from java.\* packages.

Classloading from a web application perspective is up next.

### Web Application Classloading Perspective

Application servers dictate where and in what order web applications, such as Liferay DXP, search for classes and resources. Application servers such as [Apache Tomcat](#) enforce the following default search order:

10. Bootstrap classes
11. Web app's WEB-INF/classes
12. web app's WEB-INF/lib/\*.jar
13. System classloader
14. Common classloader

First, the web application searches Bootstrap. If the class/resource isn't there, the web application searches its own classes and JARs. If the class/resource still isn't found, it checks the System classloader and then Common classloader. Except for the web application checking its own classes and JARs, it searches the hierarchy in parent-first order.

Application servers such as [Oracle WebLogic](#) and IBM WebSphere have additional classloaders. They may also have a different classloader hierarchy and search order. Consult your application server's documentation for classloading details.

### Other Classloading Perspectives

The [Bundle Classloading Flow](#) tutorial explains classloading from an OSGi bundle perspective.

Classloading for JSPs and Service Builder classes is similar to that of web applications and OSGi bundle classes.

You now know Liferay DXP's classloading hierarchy, understand it in context of web applications, and have references to information on other classloading perspectives.

### What are the Portlet Modes Available in Liferay 7 and Explain them?

```
E:\workspaces\my-liferay-workspace\modules>blade create -l
activator
api
content-targeting-report
content-targeting-rule
content-targeting-tracking-action
control-menu-entry
fragment
mvc-portlet
panel-app
portlet
portlet-configuration-icon
portlet-provider
portlet-toolbar-contributor
service
service-builder
service-wrapper
simulation-panel-entry
template-context-contributor
theme
```

### What is the Directory (Folder) Structure of a Portlet Module In liferay 7?

Here's the module project anatomy:

- src/main/java/ → Java package root
- src/main/resources/content/ (optional) → Language resource bundle root
- src/main/resources/META-INF/resources/ (optional) → Root for UI templates, such as JSPs
- bnd.bnd → Specifies essential OSGi module manifest headers
- build.gradle → [Configures dependencies](#) and more using Gradle

The figure below shows an MVC portlet module project.

### **What are the Databases Supported by Liferay 7?**

- All major database support by liferay including
  - Db2
  - Mysql
  - Oracle
  - SQL

### **What are the steps to migrate plug-in from Liferay 6.2 to Liferay 7 DXP?**

- DXP you need to update your existing code to make it compatible with DXP. Liferay provides code upgrade tool for it.
- For JSP hook migration it's better to create a hook module from scratch and overwrite the required JSP page. The reason behind that is because Liferay JSPs have changed significantly from Liferay 6.x to 7.x, so using old 6.2 JSP pages for new 7.1 Liferay may break some functionality.
- You can follow LR's documentation to upgrade your existing code to LR DXP

### **How to Use JUnit In Liferay Portlet Classes?**

- Test source folder
- Test source properties
- Dependencies
- Mockito or Powermockito
- Test classes
- Mock Class

### Explain the Parameters available in the Portlet URL and their uses?

- **p\_p\_id:** current portlet id
- **p\_p\_state:** window state
- **p\_p\_mode:** portlet mode either view/edit
- **p\_p\_lifecycle:** this is life cycle of portlet 0/1/2
  - 0: render phase or render URL
  - 1: action phase or action URL
  - 2: server resource URL
- **mvcPath:** this is another parameter which we passed from <portlet:param/> tag

<https://www.surekhatech.com/blog/liferay-migration-process-from-6-2-to-7-1-dxp>

### What is Multi tenancy In Liferay ?

- Liferay supports multi-tenant configurations via multiple portal instances in a single web container.
- What is the Portal instance? It is a Liferay's feature which allows you to have two or more separated user spaces under one physical installation. It simplifies the deployment and administration.
- Liferay also provides a [Properties Override](#) capability to override default properties file so each Portal Instance can have its own properties file. As you might know, settings for Documents and Media are put in properties file.

### How to Embed a Web Content (Journal Article) In Portlet?

<liferay-ui:journal-article