## AEM Interview Questions

1. **What is a Content Management System?**

Content Management System (CMS) is software used to create, edit, manage, and publish the content in a consistently organized fashion.

1. **Why should we use a Content Management System (CMS)?**

A content management system is a piece of software that is used to create and manage the documents on Web sites. Many websites are dynamic, so their content needs to be updated frequently. In order manage these changes efficiently; it is recommended that a content management system be used.

1. **What is a Web Content Management (WCM) System?**

A web content management system (web CMS) is a bundled or stand-alone application to create, manage, store and publish the content on Web pages. Web content includes text and embedded graphics, photos, video, audio, and code (e.g., for applications) that displays content or interacts with the user.

1. **What is Enterprise Content Management (ECM)?**

The technologies used to capture, manage, store, preserve, and deliver content and documents related to

organizational processes.

1. **What is AEM (Adobe Enterprise Manager)?**

Adobe Experience Manager is a Java based web content management system for building, managing and deploying commercial websites and related services. It combines a number of infrastructure-level and application-level functionalities.

**Infrastructure level:**

* Web server(CQSE)
* Content repository(CRX)
* Web application framework (Sling, Felix, OSGi).

**Application level:**

* Build, author and publish websites.
* Replication and Reverse Replication.
* Wide variety of rich UI components.
* Dispatcher and dispatcher flush agents.
* Package Manager: Package everything and move from one instance to other easily.
* Workflow Management: Define and implement workflows for the creation, editing and publishing of content (Deliver the content and documents through organizational approval processes).
* DAM-Admin: Manage a repository of digital assets such as images, videos and documents, and integrate these assets into your website.
* Site-Admin: Manage all the content pages of the website.
* Search API: we can search the content no matter where it is stored in your organization.
* Set up social collaboration tools like ratings, comments, blogs, user groups, and calendars.
* Tagging: Organize your digital assets and web pages using tagging.
* Campaign Management (Plan, design, launch, and optimize marketing campaigns).
* MSM, Language copy, Live/Language copy, Blue print, I18N.
* Scaffolding.
* Personalization.

1. **What is CQ5 (Communiqué)? What is CQ5? Why should we go for CQ5 instead of any other ECM tools?**

CQ5 is a java based large-scale web content management system. It combines web content management (WCM), digital asset management (DAM), and social collaboration to deliver a solution that allows managing large amount of information, multiple internal and external websites.

1)    It is based on a content repository and use JCR (Java Content Repository) specification to access the content repository.  
2)    It uses RESTful Apache Sling framework to map request URL to the corresponding node in content repository  
3)    it uses powerful OSGi framework internally to allow modular application development. It means individual pieces of your application (called bundles in terms of OSGi) can be independently started and stopped.

**Specialty:**

CQ is an online content management system. This means that authors do not need to install special software in order to use CQ only a browser is needed. In CQ, authors create, modify and delete content on an authoring environment. Basic authoring activities in CQ such as adding text, images, and basic layouts, do not require the knowledge of HTML. Changes to content in CQ are not made visible to the public until authors activate them (i.e., publish the content). Authors can also deactivate content if necessary.

1. **What are the advantages of AEM over another CMS?**

One big advantage of AEM over another CMS is how it integrates with other products from Adobe and with the Adobe Marketing Cloud. AEM comes built in with features like workflows to control content in the CMS, the use of search queries to find anything you are looking for, setting up social collaboration, tagging content, and a way to manage your digital content.

AEM also includes a way to manage mobile applications, mobile websites, e-commerce, and marketing campaign management.

1. **What is Author instance CQ5 and how will you start it?**

An Author instance is the instance, where the content authors will log into and manage pages.

This includes: 1) creating, 2) editing, 3) deleting, 4) moving, 5) administering etc

**How to start:**

1. Place the cq5 quick start jar and license.properties in the directory structure something like “…/cq5/author”
2. Rename the cq5 quick start jar to cq-author-4502.jar.

* cq = the application
* author = the WCM mode it will run in (e.g. author or publish)
* 4502 = the port it will run in (e.g. any available port is acceptable)

1. Double-click the cq-author-4502.jar file.

**Note:** You can also install/start CQ5 from the command line while increasing the Java heap size, which will improve performance. Please see image below.



**Note2:** From next time onwards we can directly go to the folder …\author\crx-quickstart\server\ and directly run server.bat file to start cq instance.

1. **What is Publish instance CQ5 and how will you start it?**

A Publish instance is the instance where the content will be placed on the website, from which web site visitors will request the pages.

**How to start:**

1. Make sure that your Author instance is stopped. You cannot successfully clone a running CQ5 instance.
2. Copy your author folder (e.g. C:/day/cq5/author).
3. Paste your author folder into the same folder structure (e.g. C:/day/cq5).
4. After the copy rename your author-copy folder to publish (e.g. C:/day/cq5/publish).
5. Rename the CQ5 quickstart JAR to cq-publish-4503.jar.

* cq = the application
* publish = the WCM mode it will run in (e.g. author or publish)
* 4503 = the port it will run in (e.g. any available port is acceptable)

Double-click the cq-publish-4503.jar file.

1. **What is the difference between the Author and Publish environment?**

A production environment usually has two different instances of AEM running. One is the author instance, and the other is the publish instance. These two instances are usually kept in two different settings.   
The **author** instance is where you will enter and manage content for your website. This is where you will administer your site as well. The author environment is usually kept behind a firewall.

The **publish** instance is where you will make your content available for your targeted audience. The publish environment is usually kept in a Demilitarized Zone (DMZ).

In computer networks, a **DMZ** (demilitarized zone) is a physical or logical sub-network that separates an internal local area network (LAN) from other untrusted networks, usually the Internet.

1. **How to change the default port number? What is the port no hierarchy?**

We can set the port number by renaming the quickstart jar (aem-author-p6754.jar). The default port for AEM is 4502. If that port is not available or already in use, Quickstart automatically configures itself to use the first available port number as follows: 4502, 8080, 8081, 8082, 8083, 8084, 8085, 8888, 9362…etc.

1. **What is a run mode? What are all the run modes we have in AEM?**

Run modes allow you to tune your AEM instance for a specific purpose.

Run modes: author, publish, primary, test, development, intranet, samplecontent, nosamplecontent….etc.

A collection of values for configuration properties, used for a particular run mode, can be saved in the repository Ex: config.author, config.publish…etc.

1. **How can we start AEM with different run modes?**

* [sling.properties file](http://docs.adobe.com/docs/en/aem/6-0/deploy/configuring/configure-runmodes.html#Using%20the%20sling.properties%20file) (sling.run.modes=author)
* [-r option](http://docs.adobe.com/docs/en/aem/6-0/deploy/configuring/configure-runmodes.html#Using%20the%20-r%20option) (java -jar cq-56-p4545.jar -r dev)
* [system properties(In the start script](http://docs.adobe.com/docs/en/aem/6-0/deploy/configuring/configure-runmodes.html#Using%20a%20system%20property) , -Dsling.run.modes=publish, prod)
* [Filename detection](http://docs.adobe.com/docs/en/aem/6-0/deploy/configuring/configure-runmodes.html#Filename%20detection%20-%20renaming%20the%20jar%20file)(aem-author-p6754.jar)

1. **How do I install CRX packages automatically on CRX server startup?**

By default the folder “crx-quickstart/install” is watched for files. If a bundle, configuration or content package is put into this directory, it is automatically picked up and installed. If it's removed, it gets uninstalled.

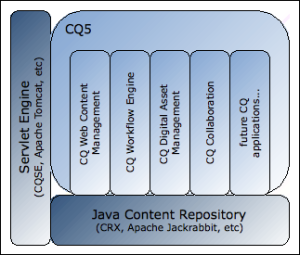
If something goes wrong, the web console and the repository are not reachable then we can use this option.

1. **What is the technology stack of cq5?**

JCR (JSR), RESTful Apache Sling Framework, OSGi framework.

**JCR**: Java specification for accessing a content repository, CQ5 uses its own implementation of JCR called CRX.  
**Apache Sling**: it’s web application framework based on REST principles used to access a JCR over HTTP protocol. It maps the incoming request URL to the respective node in the content repository.  
**OSGi**: It defines architecture for developing and deploying modular applications using java. Each module called bundle can be independently started and stopped.

1. **Architecture of CQ5 application?**

[](http://adobecq5interviewquestions.files.wordpress.com/2013/11/architecture_overview.png)

**Servlet Engine** (CQSE)

The Servlet Engine acts as the server within which each CQ instance runs as a web application. Any Servlet Engine supporting the Servlet API 2.4 (or higher) can be used. We can use day’s CQSE.

**JCR** (Java Content Repository)

JCR is a specification, provided by the java community (javax.jcr.\* package) to access the content repository bi-directionally in a uniform way). CRX is the Day's proprietary implementation of JCR.

**CQ5** (Communique)

The common foundation of the CQ5 platform provides a basis for the interoperability and seamless integration of all CQ applications. CQ WCM ([Web Content Management](http://en.wikipedia.org/wiki/Web_content_management_system)) and the CQ Workflow Engine, CQ DAM, CQ Collaboration and other CQ future applications.

1. **What is the server startup sequence?**

CQSE 🡪 CRX 🡪 SLING 🡪 CQ5

1. **What is a content repository? What is JCR?**

A Content repository is basically a place where the digital content is stored. Generally the structure of the content repository is hierarchical and represented as a tree structure where each node of the tree is used to store content.

**JCR:** Java Content Repository **JSR:** Java Specification Request

JCR is a specification, specified in the JSR 170 and 283, provided by the java community (javax.jcr.\* package) to access the content repository bi-directionally in a uniform way (platform independent and vendor independent way). CRX is the Day's proprietary implementation of JCR. Apache Jackrabbit is an open-source implementation of JCR 2.0 specification.

1. **What is REST? What are its principles?**

REST stands for Representational State Transfer (it’s Architecture), based on HTTP protocol. According to the REST architecture (useful in development of web services). Rest ful applications use HTTP requests to post, read and delete the data. It should follow some principles:

**Client-server:** There should be a clear separation between a client and server.

**Stateless:** The client context should not be stored on the server between requests. Each request from any client contains all the information necessary to service the request, and any session state is held in the client. The server can be stateful.

**Cacheable:** On the web, clients can cache responses. Well-managed caching partially or completely eliminates some client–server interactions, further improving scalability and performance.

1. **What is Sling? How is it different from other web-development frameworks?**

CQ5 is built using Sling, a Web application framework based on REST principles that provides easy means to develop content-oriented applications. Sling uses a JCR repository, such as Apache Jackrabbit, or Day's CRX, as its data store. It is a content driven(content centric) framework i.e. it maps the incoming user request based on URI with corresponding node in the content repository and depending on the type of the request (GET,POST, etc.) executes the corresponding dynamic script. The main advantages of sling are:

* It maps the URL directly to the content.
* It is REST full.

1. **How is resource resolution done in Sling?**

Using sling REST principles:

* the mapping uses the content path extracted from the request to locate the resource
* when the appropriate resource is located, the sling resource type is extracted, and used to locate the script to be used for rendering the content
* (Priority) Selector+extn 🡺 selector 🡺 extn 🡺 method (GET/POST)



1. **What is the selector and what is its significance?**

Selector is an extra string added to the request URL. This provides excellent support for Web content authors to build pages which are easily customized to their requirements for different renditions/views.

Ex: /content/trainingSite/en/company.mob.html

/content/trainingSite/en/company.html

Here “mob” is the selector, so if we want mobile page we can use mob as selector like company.mob.html (mob.html.jsp) otherwise we can use company.html (html.jsp) which gives normal web page.

1. [**Sling Scripts cannot be called directly in CQ5- Why?**](http://www.aptibook.com/discuss-technical?uid=tech-cqfive16&question=Sling-Scripts-cannot-be-called-directly-in-CQ5--Why?)

Scripts cannot be called directly within Sling since it will break the strict concept of a REST server; you would mix resources and representations. If you call the representation (the script) directly you hide the resource inside your script, so the framework (Sling) no longer knows about it.

1. [**Sling Tag Library - Explain**](http://www.aptibook.com/discuss-technical?uid=tech-cqfive18&question=Sling-Tag-Library---Explain)**?**

In order to utilize helpful Sling functions, Sling Tag library must be declared.

<%@taglib prefix=*"sling"* uri=*"http://sling.apache.org/taglibs/sling/1.0"* %>

1. **What is a Template?**

Template is the blueprint for a specific of a page. It is used to create a page and defines which components can be used to render within the scope. A template is a hierarchy of nodes that has the same structure as the page to be created, but without any actual content. So the pages which are created using the same template are similar in structure/look like.

A CQ template enables you to define a consistent style for the pages in your application. A template comprises of nodes that specify the page structure.

1. **What is the significance of template properties allowedPaths, resourceType? How the template will define component’s scope?**

“sling:resourceType” property is used to locate/define the script to be used for rendering the content­­.

So whatever the pages created using this template will have the similar structure as defined in the script. “sling:resourceType” will be created on the template's jcr:content node.

Ex: sling:resourceType = training/components/page/contentpage

“Allowed Paths” is used to define paths where this template may be used to create pages.

Ex: allowedPaths=/content(/.\*)? The pages created under the path /content can use this template.

1. **What is a Page? How will you create a page in CQ?**

A page is an 'instance' of a template. A Page is where the content authors will create and edit content that will most likely be published and viewed by site visitors.

1. **What is Page Component/Page-rendering component?**

It is a top level component, to be used to render the page. The Template you created will use this component's resources (scripts, dialogs, etc.) to render the pages.

1. **What are widgets? How are widgets used?**

**Widget:** In CQ all user input is managed by widgets. These are often used to control the editing of a piece of content. Dialogs are built by combining Widgets. Widgets, which are used to author/render the content.

1. **What is Dialog and Design Dialog Difference between them?**

A **Dialog** is a special kind of widget/key element of the component and provides an interface for author to configure and provide input to the component. The user input will be stored at the page level in jcr.

It can be **modified in edit mode**.

The root node of the Dialog has to be of node type “cq:Dialog” and named as “dialog”. The Default dialog will contain following node structure.

Dialog 🡺 items(cq:TabPanel) 🡺 items(cq:WidgetCollection) 🡺 tab1(cq:Panel)

Below this dialog root node, the nodes for the tabs of the Dialog have to be added. These “tab nodes” must be of node type “cq:WidgetCollection”. Below the “tab nodes” the “widget nodes” can then be added. The “widget nodes” must be of node type “cq:Widget”.

**Design dialog** is a dialog that will only display when you are in design mode. The input that is received here will be stored at the template level (globally) and will be accessed across the whole site.

The root node of the Design Dialog has to be of node type “cq:Dialog” and named “design\_dialog”. Design dialog will share the content at the template level. Also we can dynamically change the content in design mode.

**Ex:** Designers manage a Web site’s logo (binary content), through the use of a Design Dialog, it will ensure the displayed logo and related link are consistent for all pages that use the same Template/Design.

1. **What is a component? How to create a new component? How to configure a component?**

**Components** are re-usable modules that implement specific functionality/application logic to render the content on your website. A Component could be described as a collection of scripts (e.g. JSPs, Java servlets, etc.) that completely realize a specific function.

**Imp properties of a component:**

allowedChildren: Path of a component that is allowed to be a child of this component.

allowedParents: Path of a component that is allowed to be a parent of this component.

sling:resourceSuperType: When set, the component inherits from this component.

**Note:** A page has a hierarchy node of type “cq:Page” and a content node of type “cq:PageContent”.

The property sling:resourceType of the content node points to the Page Component used for

rendering the page.

1. **Why should we include init.jsp script? How to initialize CQ5 WCM?**

When we include init.jsp in our page it will perform all necessary steps to provide the whole WCM functionality on a page, including: Dialogs, Widgets, WCM CSS & JS, Sidekick, etc.

<cq:include script=*"/libs/wcm/core/components/init/init.jsp"*/>

1. **Why should we include the global.jsp script?**

CQ provided “global.jsp” declares Sling, CQ and JSTL tag libraries; this shortens and simplifies the JSP coding of our component. We can also get CQ define objects in our JSPs.

<%@include file=*"/libs/foundation/global.jsp"*%>

1. **What are all the different modes in CQ5? How can we identify a particular page (html) is in edit mode or design mode in my JSP page?**

EDIT, DESIGN, PREVIEW

By comparing “WCMMode.*EDIT”* or *“*WCMMode.*DESIGN”* or *“*WCMMode.*PREVIEW”* With the incoming request’s WCM mode using “WCMMode.*fromRequest*(actionVo.getSlingRequest())” or “actionVo.getWcmMode()”

1. **How can we identify whether a request is coming from Author or Publish instance in my JSP page?**

**Author Instance:**

If the incoming request’s WCM mode is EDIT or PREVIEW then it is author instance.

WCMMode.*fromRequest*(actionVo.getSlingRequest()) == WCMMode.*EDIT* || WCMMode.*fromRequest*(actionVo.getSlingRequest()) == WCMMode.*PREVIEW*

**Publish Instance:**

If the incoming request’s WCM mode is DISABLED or PREVIEW then it is publish instance.

WCMMode.*fromRequest*(actionVo.getSlingRequest()) == WCMMode.*DISABLED* || WCMMode.*fromRequest*(actionVo.getSlingRequest()) == WCMMode.*PREVIEW*

1. **How can we identify a particular request is coming from Author or Publish instance in my sling Service or Sling Servlet?**

@Reference

SlingSettingsService slingSettingService;

**final** Set<String> runmodes = slingSettingService.getRunModes();

**for** (String runmode : runmodes) {

**if** (runmode.equalsIgnoreCase("author"))

*isAuthoringInstance* = **true**;

}

1. **Want to restrict some script on publish instance how can I do that?**
2. **What are Define objects** (cq:defineObjects) **in CQ?**

These are all the objects available in our JSP page if we include global.jsp (/libs/foundation/global.jsp)

slingRequest - The wrapped Request Object (SlingHttpServletRequest).

slingResponse - The wrapped Response Object (SlingHttpServletResponse).

resource - The Sling Resource Object (slingRequest.getResource();).

resourceResolver - The Sling Resource Resolver Object (slingRequest.getResoucreResolver();).

currentNode - The resolved JCR node for the request.

log - The Default logger ().

sling - The Sling script helper.

properties - The properties of the addressed resource (resource.adaptTo(ValueMap.class);).

pageProperties - The properties of the page of the addressed resource.

pageManager - The page manager for accessing AEM content pages (resourceResolver.adaptTo(PageManager.class);).

component - The component object of the current AEM component..

designer - The designer object for retrieving design information (resourceResolver.adaptTo(Designer.class);).

currentDesign - The design of the addressed resource.

currentStyle - The style of the addressed resource.

1. **What is Parsys and IParsys? What is the difference between them?**

**Parsys:** Paragraph System, which is a placeholder/compound component where we can drag and drop the components on to the page. Allows authors to modify the content of the page dynamically i.e. add, delete, move, copy and paste “paragraphs” on a page. This component only contains a design dialog (i.e. design\_dialog), thus can only be modified in design mode.

Ex: <cq:include path="par" resourceType="foundation/components/parsys" />

**Iparsys:** Inherited Paragraph System is similar to parsys except it allows you to inherit the created paragraphs from the parent.

Ex: <cq:include path="par" resourceType="foundation/components/iparsys" />

1. **What is a Parbase? What its significance?**

Parbase is a key component which allows components to inherit attributes from other components, similar to subclasses in object oriented languages such as Java.

**Ex:** When you open the /libs/foundation/components/text node, you can see that it has a property named “sling:resourceSuperType”, which references the parbase (foundation/components/parbase) component. The parbase here defines tree scripts to render images, titles, and so on, so that all components sub classed from this parbase can use this script. Also for image component, crop,map etc inheritd using parbase.

1. **What is significance of allowedParents : “\*/\*parsys” ?**
2. **What is a CQ5 overlay/override component?**

The overlay/override component is used when you want to use an OOTB component in AEM and you

want to add extra features to it and you want to change it across all instances of that component.

Ex: If you want to add features to the OOTB “text” component, but you don’t want to create a new component, you would want to copy the “text” component from “libs/foundation/components” to “apps/foundation/components” and keep the same folder structure that is used.  
  
When this is done, any changes that done to this new component will be reflected in the out of the box “text” component, without changing the original code under “libs”. It is generally not a good practice to modify the original code, which is why CQ offers the ability to use overlays.

1. **What is extending a component?**

Extending a component is when you want to create a new component that will not override a base component, but will have the same features as the original so that you can change or build upon that component.  
In order to extend a component, you must set the “sling:resourceSuperType” of the component to the base component you would like to extend. By doing this, you will inherit everything from the base component.

1. **What is the difference between the Overlay components and Extended Components?**

|  |  |
| --- | --- |
| **Overlay/Override Components** | **Extended/Inherited Components** |
| 1. Universally available across the site. | 1. Not universally available across the site as they are inherited for specific project. |
| 2. They are copied from “/libs/foundation/components/ ” to “/apps/foundation/components/” | 2. It can be done by different ways.   * ResourceType Hierarchy * Container Hierarchy * Include Hierarchy |

1. **Explain about Extending/inheriting of components in CQ in detail?**

Components can be given a hierarchical structure to implement the inheritance of included script files, Dialogs, etc. Therefore it is possible for a specific “Page” Component (or any Component) to inherit from a “base” Component.

Components within CQ5 are subject to 3 different hierarchies:

**Resource Type Hierarchy:**

This is used to extend components using the property “sling:resourceSuperType”. This enables the component to inherit (scripts, dialogs, descriptions..) from a “base” Component.

**Ex:**

1. Add the property “sling:resourceSuperType” :: “foundation/components/page” on the component where you want to override for example our “page” component i.e. “contentpage”.
2. Now we can use foundation page component’s “head.jsp” script in our local “Page” component.
3. **How to extend an existing foundation component?**

By placing “sling:resourceSuperType” property on the component where we want to extend the existing foundation component.

Ex: sling:resourceSuperType = “foundation/components/page”

1. **What is “cq:EditConfig” what its significance?**

**Container Hierarchy:**

This is used to populate configuration settings to the child component and is most commonly used in a paragraph system scenario. Configuration settings for the edit bar buttons, control set layout (editbars, rollover, etc.), Dialog layout (inline, floating, etc.) can be defined on the parent component and propagated to the child components. Configuration settings (related to edit functionality) in cq:editConfig and cq:childEditConfig are propagated.

**Ex:**

1. First use the Sling:resourceSuperType :: foundation/components/parbase property on the component where you want inherit the parent component’s properties.
2. Add “cq:editConfig” node inside the component.
3. **How can you inherit properties of one dialog to another dialog?**

**Include Hierarchy:**

This is imposed at runtime by the sequence of includes. This hierarchy is typically used by the Designer, which in turn acts as the base for various design aspects of the rendering; including layout information, CSS information, and the available components in a paragraph system, etc.

**Ex:**

By adding the properties on the dialog where you want to include the parent dialogs/widgets/properties.

Xtype : cqinclude

Path : /libs/foundation/components/image/dialog/items.infinity.json

1. **How you can inherit properties of one dialog to another?**

You can inherit properties from one dialog to another if the new component has the “sling:resourceSuperType” and this property is set to the base component. This will automatically inherit all the properties from the original dialog.  
If you only want to inherit certain tabs from a dialog, you can use the “cq:include” xtype to inherit just that tab.

1. **Why should we include resourceType? Diff between resourceType and resourceSuperType?**

“sling:resourceType” is used to locate the script to be used for rendering the content­­.

“sling:resourceSuperType” is used to (achieve inheritance) form a hierarchy of resources. When we set it on the component it inherits the script/component from this.

1. **Have you developed any custom component which is not relevant to OTB component?**
2. **How to override users and groups instead of users and groups tabs?**
3. **How will you extend multicomposite field?**
4. **What are mixin node types, give some examples?**
5. **What are mixin types? How will you implement i18n without using mixin?**

There are two categories of node types, **primary** and **mixin**.

**Primary:** Every node has a primary node type assigned to it upon creation. It defines node structure; generally it will contain child nodes (allowed and required) and properties.

**Mixin:** It may be added to a node later in its lifecycle. It will specify additional properties or a capability being added to the node.

**Ex:**  mix:language, mix:versionable , mix:lockable, mix:tagable.

1. **What is i18n node? How will you create it?**

**Steps to create i18n node:**

* Create an i18n node of type sling:Folder.
* Create an en (language) node of type sling:Folder under the newly created i18n node.
* Assign the mixin to the newly created en node i.e. mix:language in the Content Explorer
* Assign the property to en node: Name = jcr:language Value = en\_US
* Create a “fieldLabel” folder (node) of type sling:MessageEntry under en node.
* Assign the following properties to the newly created fieldLabel node:
* Name = sling:key Value = some key value (i18n-title)
* Name = sling:message Value = locale specific(translated) message

**In JSP page:**

For rendering these i18n message values into jsp we use

<cq:setContentBundel>

<fmt:message key=”i18n-title” />

**OR**

java.util.Locale  pageLocale = currentPage.getLanguage(**true**);

//If above bool is set to true. CQ looks in to page path rather than jcr:language property.

java.util.ResourceBundle resourceBundle = slingRequest.getResourceBundle(pageLocale);

I18n i18n = **new** I18n(resourceBundle);

%>

To print <%=i18n.get("title Key") %>

1. **What is difference between a JSP, CQ and Sling include?**

The <%@ include file=”myScript.jsp” % > directive informs the JSP compiler to include a complete file into the current file at compilation time. It is as if the contents of the included file were pasted directly into the original file.

The <cq:include script=”myScript.jsp” /> directive is different, it includes the file at run time. When develop CQ components, Day recommends to use <cq:include /> tag. This allows you to directly include script files by their name when using the script attribute.

1. **What is a file-vault?**

FileVault is a tool that is offered by Adobe that will map the content of the CRX instance to your file system. The VLT tool is similar to a Subversion client since it will allow you to checkout content and to check-in content.

1. **How to connect Eclipse to CQ using file-vault?**

Need to use vault eclipse plugin and file vault software dump.

1. **What is DAM?**

CQ DAM (Digital Asset Management) is a digital asset management tool that is fully integrated with the CQ platform and enables your enterprise to share and distribute digital assets. Users across an organization can manage, store, and access images, videos, documents, audio clips, and rich media such as Flash files for use on the web, in print, and for digital distribution.

1. **How do you load digital assets in to DAM?**

You can add digital assets to your dam by using the GUI i.e. DAM-Admin or through WebDav access. If you are using the graphical user interface, you would just browse for the selected files you would like to add, and then cq will create the metadata for those assets in the dam folder. You would generally use the WebDav option when you want to upload a large number or assets at once.

1. **How will you give the privileges to user for uploading dam Assets?**

### Go to user admin console.

### Double click on the user/group name.

### Give read, write, and modify permissions on “/content/dam” node.

1. **How to connect CQ using WebDav?**

### Edit the regedit.exe by going to start

### System 🡪 CurrentControlSet 🡪 Services 🡪 WebClient 🡪 Parameters

### Change “BasicAuthLevel” parameter to 2

### Go to MyComputer

### Add Network Location 🡪 Choose a custom network location 🡪 next 🡪

### Give crx repository url (http:localhost:4502) and click on next.

1. **Can we restrict for certain users not to display some digital assets?**

Yes, we can follow different ways to achieve this.

1. You can restrict access to certain folders in the DAM by using closed user groups. You can do this from the digital assets menu, right click on the folder that you want to add this property to, and click “properties”. Then you go to the CUG tab and add the required information.
2. Go to the user admin console. And deny read permissions for that particular user at the required dam path. Ex: /content/dam/images/imageX.
3. **How is it possible to disable Site Admin actions for particular groups, e.g. Activate?**

In order to disable and completely hide a certain action in the Site Admin console for a particular group, a corresponding group ACL has to be defined that denies read access to this action.

Steps to disable site admin actions:

1. logged in as admin, open <http://localhost:4502/crx/explorer/browser/index.jsp>
2. Navigate to “/libs/wcm/core/content/siteadmin/actions/activate”
3. Next click on the Security button and select Access Control Editor
4. In the Applicable Access Control Policies section, mark the checkbox next to “org.apache.jackrabbit.core.security.authorization.acl.ACLTemplate”
5. Click on Set selected policies
6. Next click on New ACE
7. Browse the Principal for the group for which a privilege is to be set
8. DENY “jcr:read” and confirm
9. Click Apply and close the window
10. At this point, members of the specified group won't have access to the Activate action anymore.

**Actions on different consoles:**

|  |  |
| --- | --- |
| **Console** | **Path** |
| Site Admin | /libs/wcm/core/content/siteadmin/actions |
| DAM Admin | /libs/wcm/core/content/damadmin/actions |
| Tools | /libs/wcm/core/content/misc/actions |
| Security Admin | /libs/cq/security/content/admin/authlist/actions |
| Inbox | /libs/cq/workflow/content/inbox |
| Tagging | /libs/cq/tagging/content/tagadmin |
| Campaigns | /libs/mcm/content/admin |
| Community | /libs/collab/core/content/admin |

1. **I want to put 1000 images in DAM and want each image go through with workflow, how can I do that?**
2. **I have some image in DAM which was used in several pages, how can I know that in which pages that the image was used (search query)?**

**By using an xpath query on CRXDE we can achieve this.**

1. **What is a workflow, how will you start a workflow and how does it work?(Implementation of workflow)**
2. **What are the actions need to set while triggering the workflow?**
3. **What type of steps available in workflow and diff among them?**
4. **How do you write a workflow process step?**
5. **How to automate a workflow? How to trigger workflows automatically (launcher tab)?**
6. **What are workflows?**

Workflows in CQ are used to automate/implement managerial activities such as approval processes to receive sign-off by various participants. They consist of series of steps that are executed on the pages or assets when they are triggered. Ex of these steps are activating a page, sending an email, approving a page…etc.

Main workflow objects:

**Model**: Is made up of workflow nodes and workflow transitions. Transitions connect the nodes and define a “flow”. The Model has always a start node and an end node.

**Step**: A workflow is made up of work flow steps. Step represents a task/activity to perform by a model or workflow. When we edit a model we can see the available steps in the Sidekick.

* **Participant (User/Group) Step:** We can assign ownership for a particular action. The workflow will only proceed when the user has manually acknowledged the step. This is used when you want someone to take an action on the workflow.

**Ex:** A review step.

* **Process (Script, Java method call) Step:** Process steps are automatic actions that are executed by the system if certain specified conditions are met. We can define/implement these steps by using either an ECMA script or a service (a Java class in a bundle).

Services can be developed to listen to special workflow events and perform tasks according to the business logic. **Ex:** e-mail notification

* **Container:** A container step enables you to connect workflows, by referencing a child workflow. This can allow you to reuse a workflow (or sub-workflow)

**Ex:** translation process which is used in multiple editing workflows.

* **OR Split/Join:** Create a split in the workflow, whereby only one branch is active. This allows you to introduce conditional processing paths into your workflow.
* **AND Split/Join:** An AND Split creates a split in the workflow; both branches will be active. This enables you to introduce multiple processing paths into the workflow.

**Ex:** Allowing certain review steps to occur in parallel, thus saving time.

**Transition:** Defines the link between two consecutive steps. It is possible to apply rules to the Transition.

**WorkItem:** The workItem is nothing but a task, put into respective user inbox. A workflow instance can have one or many work Items at the same time (Depending on the workflow model). The workItem references the workflow instance. In the repository the workItem is stored below the workflow instance.

**Payload:** The payload implementation references a resource in the repository that has to be advanced through a workflow.

**Lifecycle:** Is created when starting a new workflow (by choosing the respective workflow model and defining the payload) and ends when the end node is processed.

**Inbox:** Each logged in user has its own workflow inbox in which the assigned work Items (tasks) are accessible. The work Items are assigned either to the user itself or to the group to which he belongs.

**Workflow Console:**

The Workflow console is the place from where we can manage the workflows in CQ.

Within the console there are 4 tabs:

**Models:** Lists the workflow models currently available. Using **Models** tab we can Create, Edit, and Delete workflow models.

**Instances:** Shows you details of workflow instances which are currently active. Using **Instances** tab we can Terminate, Suspend, Resume, and Restart a workflow.

**Archive:** Workflow status will be archived here. So Using **Archive** tab we can check the status of a workflow, whether it is completed or aborted (for whatever reason).

**Launcher:** Using **Launcher** tab we can launch a workflow automatically, if specific conditions are met (Automate a workflow).

Ex: A node has been created/updated.

There are several ways to start a workflow.

* Workflow Console
* Websites Console
* Workflow tab in the Sidekick
* Using Launcher

1. **What are all the content automate processes in AEM? Which one is the best one?**

You can create a JCR Observation Listener, or use Sling Eventing, to listen for a suitable event such as NODE\_ADDED, and trigger whatever action is required. Or you can use the Workflow Launcher.



1. **How to customize a workflow?**

Workflows can be customized by implementing“com.day.cq.workflow.exec.WorkflowProcess” interaface and override “execute” method.

**SampleCode:**

**import** com.day.cq.workflow.exec.WorkflowProcess;

@Component

@Service

@Properties({

@Property(name = Constants.*SERVICE\_DESCRIPTION*, value = "A sample workflow process implementation."),

@Property(name = "process.label", value = "My First Sample Workflow Process")})

**public** **class** MyProcess **implements** WorkflowProcess{

**public** **void** execute(WorkItem item, WorkflowSession workflowSession, MetaDataMap args ) **throws** WorkflowException{

// Business logic

1. **What is the difference b/w workflow session and crxsession?**

Workflow session will act on only workflow related objects such as workflowdata, workitem.

1. **Difference between Listeners (JCR Observation Listeners) and handlers (Sling event handlers)?**
2. **How to convert normal Servlet to Sling-Servlet?**
3. **What class should we extend to write a Sling-Servlet?**
4. **What is the need of Sling-Servlet?**

**SlingServlet:**

Sling servlet will follow REST principles (Sling built using REST principles). Servlet is a Java object which implements “javax.servlet.Servlet” interface.

HTTP Servlets are typically used to:

* Provide dynamic content like getting the results of a database query.
* Process and/or store the data submitted by the HTML client.
* Manage information about the state of a stateless HTTP request

**Convert normal servlet to sling servlet:**

@Component(name = "com.cisco.wem.training.MyReplicationServlet", specVersion = "1.1", metatype = **true**)

@Service

@Properties({ @Property(name = "service.description", value = "Training Replication Process"),

@Property(name = "service.vendor", value = "Cisco Systems Inc."),

@Property(name = "sling.servlet.paths", value = "/c/wcm/servlets/myreplication", propertyPrivate = **true**),

@Property(name = "sling.servlet.methods", value = "POST", propertyPrivate = **true**),

@Property(name = "sling.servlet.extensions", value = "json", propertyPrivate = **true**) })

**public** **class** MyReplicationServlet **extends** SlingAllMethodsServlet {

We can write a sling servlet by extending

“org.apache.sling.api.servlets.SlingAllMethodsServlet”

Or

“org.apache.sling.api.servlets.SlingSafeMethodsServlet” Classes and overriding the method

**public** **void** doPost(SlingHttpServletRequest request, SlingHttpServletResponse response) **throws** ServletException, IOException

1. **What is the difference between SlingSafeMethodServlet and SlingAllMethodServlet and which is preferable for what?**

**SlingSafeMethodServlet:**

public class **SlingSafeMethodsServlet** extends **javax.servlet.GenericServlet**

This base class is intended for applications where data is only read i.e. support only GET requests. As such, this servlet by itself does not support the POST, PUT and DELETE methods.

**SlingAllMethodServlet:**

public class **SlingAllMethodsServlet** extends [**SlingSafeMethodsServlet**](https://sling.apache.org/apidocs/sling5/org/apache/sling/api/servlets/SlingSafeMethodsServlet.html)

Applications wishing to support data modification should use or extend the [SlingAllMethodsServlet](https://sling.apache.org/apidocs/sling5/org/apache/sling/api/servlets/SlingAllMethodsServlet.html" \o "class in org.apache.sling.api.servlets) which also contains support for the POST, PUT and DELETE methods.

1. **How is content moved from author instance to publish instance? What is Replication agent?**

RA is used to move content from author to publish instance. For automatic activation/replication, set up a workflow launcher or Sling Event Handler to activate when specified content is created/modified on the author instance.

Replication, to a publish instance or dispatcher, takes place in several steps:

* The content once published, it is moved from author to publish instance using replication agent.
* The replication agent “packages” the content and places it in the replication queue.
* The content is lifted from the queue and transported to the publish environment using the configured protocol; usually this is HTTP.
* A Listener servlet in the publish environment receives the request and publishes the received content. The default listener servlet in publish instance is “<http://localhost:4503/bin/recieve>”.

****

**Steps to configure Replication Agent:**

Usually it is configured on author instance.

1. Go to tools tab <http://host:port/miscadmin#/etc/replication>
2. Open Agents on author
3. Click on Default Agent (publish) agent 🡺 settings Edit 🡺 Transport
4. Give the URI as <http://publishhost:port/bin/receive?sling:authRequestLogin=1>
5. **How is content moved from publish instance to author instance? What is reverse replication? How does it work?**

RRA is used to Return the user input (Ex: comments, form input) from publish to author instance (under control of Author). Features such as comments and forms, allow users to enter information on a publish instance. For this a type of replication is needed to return this information to the author environment, from where it is redistributed to other publish environments.

1. **Don’t you think is there any security risk? While doing reverse replication? How the author environment controls all the traffic?**

However, due to security considerations, any traffic from publish to author environment must be strictly controlled. This functions using an agent in the publish environment which references the author environment. This agent places the input into an outbox (http://publishost:port/etc/replication/agents.publish/outbox.html). This outbox is matched with replication listeners in the author environment. The listeners poll the outboxes to collect any input made and then distribute it as necessary. This ensures that the author environment controls all traffic.

**Steps to configure Reverse Replication Agent:**

Usually it is configured on author instance.

1. Go to tools tab <http://host:port/miscadmin#/etc/replication>
2. Open Agents on author
3. Click on [Reverse Replication Agent (publish\_reverse)](http://wbxapp-auth-dev3-01:4502/etc/replication/agents.author/publish_reverse.html) 🡺 settings Edit 🡺 Transport
4. Give the URI as <http://publishhost:port/bin/receive?sling:authRequestLogin=1>
5. Check whether the outbox (<http://publishhost:port/etc/replication/agents.publish/outbox.html>) is active or not on publish environment.

**Programmatically Triggering Reverse Replication:**

Content created on publish is not automatically reverse-replicated. Out of the box, reverse replication only works for CQ’s Blogs, Comments, Forums. Your application must write in one transaction (i.e. in one “save”): cq:distribute, cq:lastModified, cq:lastModifiedBy

1. **What is OSGi? What is the benefit of OSGI? What is Apache Felix?**

OSGi (Open Services Gateway Initiative/Dynamic Module System for Java) defines architecture for developing and deploying modular applications using java. OSGI containers allow you to break your application into individual modules called bundles. So that we can install, uninstall, start, stop and configure these bundles dynamically without restarting the container. OSGi bundles declare which other bundles they depend upon. This allows them to ensure that any dependencies are met before the bundle is resolved. Only resolved bundles can be activated.

In OSGi bundle, there will be an Activator.java class in OSGi which is an optional listener class to be notified of bundle start and stop events.

**The main advantages of using OSGI:**

1. Reduces the complexity of the system, by breaking your applications into individual modules called bundles.
2. Makes the components loosely coupled and easy to manage and we can achieve parallel development.
3. Dynamic deployment is possible without restarting the container.
4. Dependencies will be resolved at deployment time itself instead of runtime.
5. We can use multiple Versions of same jar file.

Ex: All OSGi bundles are given a version number, so it's possible for an application to simultaneously access different versions of the same bundle (eg: junit 3.8.1 and junit 4.0.). Since each bundle has its own class loader, both bundle classes can coexist in the same JVM.

1. Increases the performance of the system, since parts of application which are not in use, need not to be loaded in the memory. Ex: The logging framework can be deployed as an OSGi Bundle, which can be managed independently. Therefore, it can be started when required by our application and can be stopped when not in use.
2. One more level of control on the class file, even public classes are not available to outside environment unless they exported.

Ex: Apache Felix, Equinox

1. **Difference between OSGi bundle and Normal Jar file?**

Bundle is nothing but a jar file with metadata inside. Much of this metadata is in the jar's manifest, found at META-INF/MANIFEST.MF.

MetaData (META-INF/MANIFEST.MF): With OSGi, just because a class is public doesn't mean you can get it. Export-Package is a list of packages contained in the bundle that you would like to make available to other bundles running in the OSGi Container, and if a package isn't in the export list, it doesn't exist to the outside world.

# The list of services imported and exported by this bundle

**Export-Package**: \*(com.adobe.training)

**Import-Package**: \*

# If you don’t want the service to be publicly available.

**Private-Package**: \* (com.adobe.training.impl)

# Include-Resource:

**Bundle-Name**: My Training Bundle

**Bundle-Description**: Creating my training bundle

**Bundle-SymbolicName**: com.adobe.training.trainingbundle

**Bundle-Version**: 1.0.0-SNAPSHOT

# start and stop mtds with BundleContext implements BundleActivator

**Bundle-Activator**: com.adobe.training.Activator

1. **How do I convert an existing jar file into an OSGi bundle?**

By Adding MetaData/manifest file (META-INF/MANIFEST.MF) to the jar file we can convert a jar file to bundle. Create the bundle using the command: jar cvfm junit-4.4-bundle.jar manifest.txt junit-4.4.jar

Ex:

Manifest-Version: 1.0

Created-By: myself

Bundle-ManifestVersion: 2

Bundle-Name: JUnit 4.4 bundle

Bundle-Description: Package junit 4.4 in an OSGi bundle

Bundle-Version: 4.4.0

Bundle-ClassPath: .,junit-4.4.jar

Bundle-SymbolicName: org.junit.framework

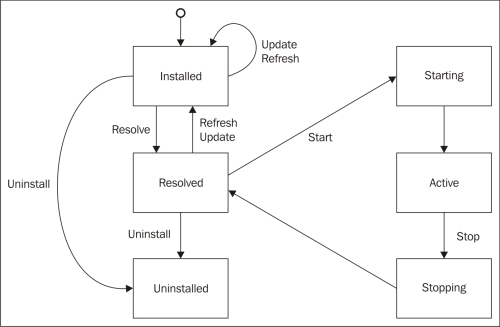
Export-Package: junit.framework,junit.extensions,org.junit.runner,org.junit,junit.textui

Note: Bundle-ClassPath header is set to allow embedding the original jar as is. Make sure its value matches the filename of the jar that you are going to embed in the bundle.

1. **What is the life cycle of OSGi?**

**OSGi Bundle lifecycle:** OSGI bundles are dynamically installed, resolved, started, updated, stopped, and uninstalled. The framework enforces the transitions between states; one cannot directly install a bundle and jump to an Active state without first passing through the resolved and starting states.

OSGi bundles declare which other bundles they depend upon. This allows them to ensure that any dependencies are met before the bundle is resolved. Only resolved bundles can be activated.



**Installed:** Bundles came into existence in an OSGi framework in the installed state. A bundle in this state cannot be immediately started or not active. There are three possible transitions: the bundle may become resolved, uninstalled, or refreshed.

**Resolved:** If all the dependencies of a bundle have been met then it will go into Resolved state. A resolved bundle may be refreshed, transitioning the bundle back to the installed state. A resolved bundle may also be transitioned to the uninstalled state. A resolved bundle is not active; however, it is ready to be activated.

**Starting:** A resolved bundle may be started. The starting state is transitory; the framework is initializing the resolved bundle into a running active state.

**Active:** The bundle is fully resolved, providing and consuming services in the OSGi environment. To perform any more transitions on an active bundle, it must first be stopped.

**Updating:** Bundle updates occur when the framework is instructed to re-evaluate a bundle's dependencies; this action is synonymous with refreshing a bundle.

**Stopping:** Stopping a bundle transitions it from the active to the resolved state. The bundle can be restarted while it remains in the resolved state.

**Uninstalled:** Uninstalling a bundle transitions an installed or resolved bundle out of the OSGi environment; however, the bundle is not removed from the environment.

1. **Can we deploy normal jar into OSGI?**
2. **How will you upload a jar file through Felix console?**
3. **How to create OSGI service?**
4. **How to register an OSGI service?**
5. **What is a bundle and how to create and implement a bundle in your application?**
6. **What is the purpose of Activator.java file?**

In OSGi bundle, there will be an Activator.java class, which is an optional listener class to be notified of bundle start and stop events.

1. **How the bundle will activate automatically?**
2. **How to create a bundle?**
3. **Steps to build bundle a through CRXDE?**
4. **Difference OSGI Bundle, Service and Component?**

**Bundle:**

Building on Java’s existing standard way of packaging together classes and resources: the JAR file, an OSGi bundle is just a JAR file. Metadata is added to promote a JAR file into a bundle. The additional metadata is added to the manifest. The OSGi metadata is provided as additional header information in the META- INF/ MANIFEST.MF file. The additional information consists of:

**Service:**

Services will provide functionality in OSGI. Services implement one or more interfaces, which define the type of service provided. It is the lifecycle of the bundle, which defines the lifecycle of the service: A service object may be instantiated when the bundle is started and will automatically be removed when the bundle is stopped.

A service is an extension of a component; via the service registry, it offers services to other bundles by publishing implemented interfaces and properties.

**Declarative Services:**

The Declarative Services specification enables the declaration of services in configuration files, which are read by the Declarative Services Runtime to observe dependencies and activate (register) and deactivate (unregister) services depending on whether requirements/dependencies are met.

**Component:**

With Declarative Services, the bundle that exposes a service is called a component and it's declared in a component.xml file that the framework understands. In OSGi terminology **a "component" is like a run-time service**. A component may or may not be a service registered with the service registry.

1. **How to call a service from other service?**

By using @reference annotation we can get the reference of another service.

@Reference

**private** ResourceResolverFactory resRelFactory;

1. **How will you get bundle context? How do you call a service from different files like JSPs and Java services?**

In a Service or Java class:

@SuppressWarnings("unchecked")

**public** **void** activate(ComponentContext componentContext) {

**this**.componentContext = componentContext;

properties = componentContext.getProperties();

}

ServiceReference serviceReference = componentContext.getBundleContext()

.getServiceReference(MessageGatewayService.class.getName());

MessageGatewayService messageGatewayService = (MessageGatewayService) serviceReference; .getBundleContext().getService(serviceReference); }

ResourceBundle resourceBundle =advancedSearchUtils. getResourceBundle(componentContext.getBundleContext().getBundle(), request.getLocale());

JSP:

<%@page session="false" import="com.adobe.training.RepositoryService"%>

<%@taglib prefix="sling" uri="http://sling.apache.org/taglibs/sling/1.0"%>

<sling:defineObjects/>

<%

RepositoryService repositoryService =

sling.getService(com.adobe.training.RepositoryService.class);

%>

<%= service.getName() %>

1. **What are all the different tabs available in Felix console and what are they used for?**

Bundles, Services, Components, Configuration, Dependency Finder, Events, JMX, Log Service, Product Information, Recent Requests, Sling Log Support, Sling Resource Resolver, System Information.

1. **How to set dialogues of editing of services in Felix console?**

By using @Property annotation we can define the configurable properties inside the java services which can be configured using felix console dynamically.

Ex:

@Properties({

@Property(name = "service.description", value = "Training Replication Process"),

@Property(name = "service.vendor", value = "Cisco Systems Inc."),

@Property(name = "sling.servlet.paths", value = "/c/wcm/servlets/myreplication", propertyPrivate = **true**),

@Property(name = "sling.servlet.methods", value = "POST", propertyPrivate = **true**),

@Property(name = "sling.servlet.extensions", value = "json", propertyPrivate = **true**) })

1. **How will you handle exceptions in Felix console?**
2. **What is URL rewriting CQ?**

The typical website structure for an Adobe CQ5/AEM project begins with /content in the URL structure. This is not an ideal URL path most people would like for their site.  To address this concern we can rewrite the URLs.

1. **Vanity URL in CQ?**

For some sites there might be a requirement to create a friendly/shorter/more expressive URL for navigating into your site.  Vanity URL allows us to do so.

**Steps:**

* Go to <http://host:port/system/console/configMgr>
* Click on “Apache Sling JCR Resource Resolver”
* Add the required entries in “Vanity URLs” field.

**Ex:** /content/en/us/products:/myProducts

1. **What is resource mapping?**

**Sling JCR Resource Resolver (**Configurable**)**

Inside AEM you can configure the Sling Resource Resolver to filter out the initial path of your site structure.

**Steps:**

* Go to <http://host:port/system/console/configMgr>
* Click on “Apache Sling JCR Resource Resolver”
* Add the required entries in “URL Mappings” field to remove the beginning portion of the URL you

want to remapped.

**Ex:** /content/:/c/

/bin/wemservices/ratings/>/c/services/rc/

**Sling Resource Mappings**

If you wish to keep URL mapping rules outside of the author’s control then you should utilize the Resource mapping features in Sling.  Under “/etc/map/http” you can create nodes of the type “sling:Mapping” that will allow you to do the same thing as vanity URLs.  These nodes require two properties to be set: “sling:match” and “sling:internalRedirect”.  The sling:match property uses regular expression to evaluate the url to match.  If the url is matched then the request is redirected to the path set in the sling:internalRedirect property.

|  |  |
| --- | --- |
| **Domain** | **Content branch** |
| www.geometrixx.fr | /content/geometrixx/fr |
| www.geometrixx.de | /content/geometrixx/de |

Ex:

/etc

   /map                       (sling:Folder)

      /http                   (sling:OrderedFolder)

         /www\_geometrixx\_fr   (sling:Mapping)

         /www.geometrixx.fr   (sling:Mapping)

{

  "jcr:primaryType": "sling:OrderedFolder",

  "www\_geometrixx\_fr": {

    "sling:internalRedirect": [

      "/content/geometrixx/fr.html"], "jcr:primaryType": "sling:Mapping",

    "sling:match": "www.geometrixx.fr/$"

  },

  "www.geometrixx.fr": {

    "sling:internalRedirect": [

      "/content/geometrixx/fr",

      "/"

    ],

    "jcr:primaryType": "sling:Mapping"

  },

  "www\_geometrixx\_de": {

    "sling:internalRedirect": [

      "/content/geometrixx/de.html"

    ],

    "jcr:primaryType": "sling:Mapping",

    "sling:match": "www.geometrixx.de/$"

  },

  "www.geometrixx.de": {

    "sling:internalRedirect": [

      "/content/geometrixx/de",

      "/"

    ],

    "jcr:primaryType": "sling:Mapping"

  }

}

1. **What is the role of Dispatcher in CQ5 and its advantages?**

The Dispatcher is Adobe’s caching and load balancing tool. Dispatcher in conjunction with a static web server (Apache httpd, Microsoft IIS, etc.) caches web pages produced by the publish instance to improve performance also protects your application server from attack. The goal of the dispatcher is to cache as much content as possible, so it does not need to access the layout engine.

### HOW DISPATCHER RETURNS DOCUMENTS

Dispatcher uses the 2 methods for caching:   
**Content Updates:** Remove the pages that have changed, as well as files that are directly associated with them, also updates the timestamp of the statfile to indicate the date of the last change.

**Ex:** If the file /en/index.html is updated, all the files that starts with /en/index. are deleted.

**Auto-Invalidation:** Auto-invalidation automatically invalidates parts of the cache. i.e. it effectively flags relevant pages as being out of date, without physically deleting them. At every content update, the statfile is updated, so its timestamp reflects the last content update.

1. **How Dispatcher performs Load-balancing?**

Load Balancing is nothing but distributing the (computational) load of the website across several instances of AEM. By this we can achieve increased processing power and increased fail-safe coverage. This is accomplished by, if the dispatcher does not receive responses from an instance, it will automatically relay the request to another instance.

Dispatcher will use 2 techniques to achieve load-balancing.

**Performance Statistics:** Dispatcher keeps statistics on how fast each instance of CQ is responding to a particular URL. Based on those metrics, dispatcher determines which instance of CQ will fetch the quickest response for any request and relays the request to that CQ instance.  
**Sticky Connections** (Personalized content): when a user session is established, then all incoming requests from that user should be served by the same CQ instance, because other CQ instances cannot recognize the user session and generate personalized pages for him. Dispatcher makes sure all requests for user session are served from the same CQ instance.

**Steps to configure Dispatcher using Apache webserver:**

* 1. Make sure the Apache is up and running.
  2. To test access http://localhost.
  3. Move “mod\_dispatcher.so” into the directory “<apache\_home>/modules” directory.
  4. Navigate <apache\_home>/conf/httpd.conf file and include the below code snippet to call “custom.conf” at the end.  
     Include conf/custom.conf
  5. Copy the attached “custom.conf” file into <apache\_home>/custom.d directory.

**custom.conf file:**



* 1. Copy “mod\_dispatcher.so” file into <apache\_home>/custom.d directory.
  2. Copy the “dispatcher.any ” file into <apache\_home>/custom.d directory.

**dispatcher.any file:**

****

1. **What is the purpose of farm file?**

A farm file is a file that is used in the configuration of a dispatcher. These are used when you want to break up the configuration file that is used for the dispatcher. They are then added under the “/farms” section in the “dispatcher.any” file.

1. **What is “/statfile” and “/statfileslevel”?**

“statfile”, “/statfileslevel” properties are used to define which parts of the website tree are invalidated when pages are activated.

**/statfile:**

The “/statfile” property identifies the file to use as the statfile. Dispatcher uses this file to register the time of the most recent content update. The statfile has no content. When content is updated, Dispatcher updates the timestamp. The default statfile is named “.stat” and is stored in the “docroot”. Dispatcher blocks access to the statfile.

**/statfileslevel:**

/statfileslevel property is used to selectively invalidate cached files according to their path.

Dispatcher creates “.stat” files in each folder from the “docroot” folder down to the level that you specify.

The “docroot” folder is level 0. When a file is updated, Dispatcher locates the folder on the file path that is at the statfileslevel, and invalidates all files below that folder only not all.

**Note:** If “/statfileslevel” is configured, Dispatcher ignores the “/statfile” property and uses “.stat” as the name.

1. **What is dispatcher flush agent? How to configure it?**

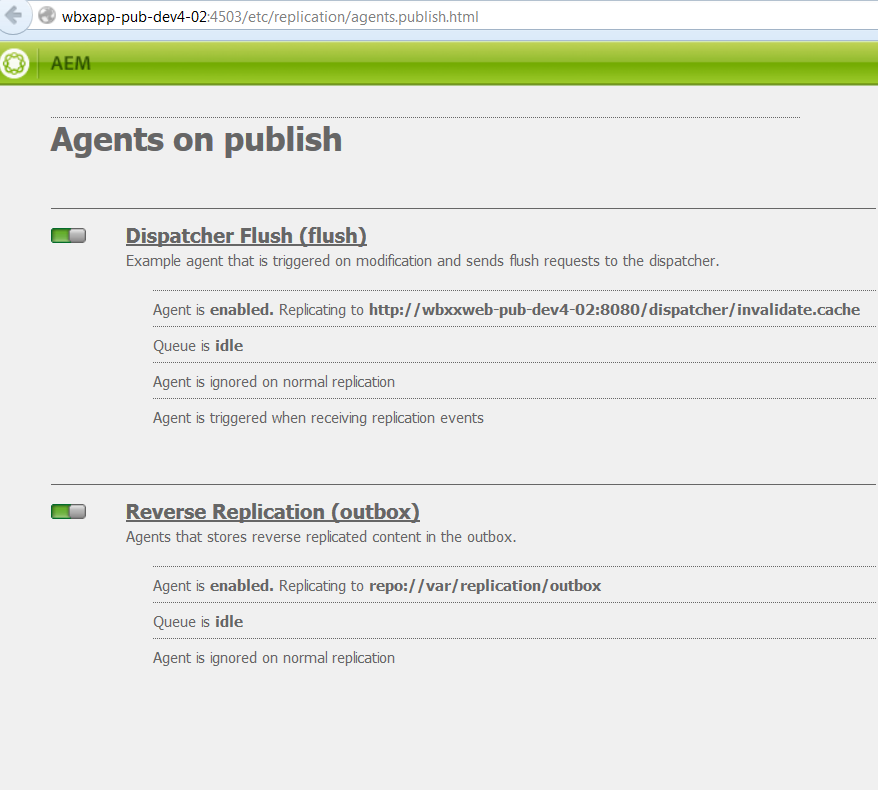
Dispatcher Flush Agent: DFA is used to flush the content from dispatcher cache If any content is updated.

Usually it is configured on publish instance.

1. Go to <http://host:port/miscadmin#/etc/replication>
2. Open Agents on publish
3. Click on Dispatcher Flush (Flush) agent 🡺 settings Edit 🡺 Transport
4. Give the URI as <http://host:port/dispatcher/invalidate.cache>

<http://publish:4503/etc/replication/agents.publish.html>

We need to make sure that agents on publish should be active on publish instance.



1. **How dispatcher cache invalidation works AEM/CQ?**

Dispatcher will use statfile and Dispatcher Flush Agent (DFA) to invalidate cache. The statfile file is a simple hidden text file. When a page is replicated the DFA sends a notification to the Dispatcher. The Dispatcher responds to the notification by updating the timestamp of the statfile. Dispatcher will serve the cached files

if and only if they are newer than the time stamp of the nearest statfile otherwise it will get them from the publish application server and update the cache.

1. **What is the use of /stickyConnectionsFor property?**

The “/stickyConnectionsFor” property defines one folder that contains sticky documents; this will be accessed using the URL. Dispatcher sends all requests, from a single user that are in this folder to the same web server. Sticky connections ensure that session data is present and consistent for all documents.

**Have any idea on Dispatcher configuration practical example? (Ex: banner component)**

**How will you set current date in dispatcher without calling cq?**

1. **Have you used any caching mechanism in your application?**

Yes we have used. JCS in ANG project, ehcache in Cisco.com

1. **What is scaffolding?**

Scaffolding used to create large set of pages that share the same structure but have differing in content.

With scaffolding you can create a form (a scaffold) with fields that reflect the structure you want for your pages and then use this form to easily create pages based on this structure.

1. **What is Tagging?**

Tagging allows authors to attach tags to a content item. We can tag a piece of content i.e. a node in the

repository. CQ enables to build tagging with various levels of complexity:

* Simple tagging: simple tags like joke.
* Hierarchical tagging: tags like cars/oldtimer or cars/suv.
* Namespaced tagging: tags containing a namespace like

channels: politics or channels: sports.

**Implementation:**

Cisco has lots of products like routers, switches, teleconferencing tools... etc., to market all the products CISCO will provide such product information on cisco.com. We have around 3 million of pages as of now.

to categorize them we used CQ tagging extensively. So for a page to exist on cisco.com it should map to at least one concept and doctype.

cq:tags is a property of type String[]

DocTypes:HOME/SUPPORT/TSD Products Support Home/TSD Products Support Category Home/TSD Products Support Series Home/TSD Products Support Install and Upgrade/TSD Products Support Install and Upgrade Guides List/TSD Products Installation and Configuration Guide

Products:Cisco Products/Routers/WAN Aggregation and Internet Edge Routers/Cisco 7600 Series Routers

Concept: what level/type of concept (product) it is

category: grouping similar products in to a category(classification)

series: similar feature models into a series

Doctype: To identify type of the document ex: marketing/sales/support/technology document.

**Tagging content in the repository:**

* Assign the “cq:Taggable” mixin to the node. (For pages (cq:Page) where the jcr:content node is of type cq:PageContent which includes the “cq:Taggable” mixin. For Assets (cq:Asset) where the jcr:content/metadata node always has the cq:Taggable mixin.)
* Set the multi-value String property cq:tags: each value is a tag ID referencing the tag.
* The tag ID needs to point to a cq:Tag node below /etc/tags.
* For each node there can only be one list of tags, that is one cq:tags property per node.

1. **What is the Query Builder?**

The Query Builder API is used will allow you to use the builder to accepts any query description, create and run an XPath query, and filter the result set. This is useful because it will allow you to search through the whole repository. It can also be used to find where certain nodes are used.

1. **What is Predicate?**

The query description is simply a set of predicates ([Predicate](https://docs.adobe.com/docs/en/cq/5-6-1/javadoc/com/day/cq/search/Predicate.html)).

Ex:

1. full-text predicate, which corresponds to the jcr:contains() function in XPath
2. image size predicate that looks for width and height properties in the DAM asset subtree.

**What is Localization and how to implement it?**

**What are the properties need to change for Internationalization?**

**What is Internationalization (I18N)?**

**I18N messages rendering? Without using tag library?**

**How will you get the locale in CQ?**

**How will you implement I18N without using mixin?**

1. **What is MSM? Rollout configurations?**

Multi Site Manager (MSM) enables you to easily manage multiple web sites that share common content. MSM lets you define relations between the sites so that content changes in one site are automatically replicated in other sites.

* MSM reduces the time it takes to manage your websites and increases the re-use of common content
* Efficiently manage different language versions of a website.
* Automatically update one or more sites based on a source site
* Enforce a common base structure and use common content across multiple sites.
* Maximize the use of available resources.
* Maintain a common look and feel.
* Focus efforts on managing the content that differs between the sites.

1. **What is Blue print and Live print difference b/w them?**
2. **What is multilingual site?**
3. **What is custom periodic importer?**

The feed importer is a framework to repeatedly import content from external sources into your repository. The feed importer will poll a remote resource at a specified interval and to create nodes in the content repository that represent the content of the remote resource. (Tools > Importers > Feed Importer). To implement your own feed importer, you can use the Interface: com.day.cq.polling.importer.Importer.

Ex: In the blog to support the auto-blogging feature, which automatically creates blog posts from an external RSS or Atom feed.

**How will you create a User and User Group?**

**Is it possible to have user without any group?**

**How will you give privileges to user for creating the user?**

**Authentication and Authorization**

1. **What is Authentication?** [**How it is achieved in CQ5?**](http://www.aptibook.com/discuss-technical?uid=tech-cqfive13&question=How-is-authentication-achieved-in-CQ5-?)

**Authentication:** It is the process of identifying, and verifying, a user.

1. Authentication information is extracted from the request. In CQ this is done by an authentication handler.
2. The authentication information is then checked to determine whether it is sufficient and/or correct. In CQ this is performed by the login modules.
3. The appropriate response is initiated.

To provide controlled access to various areas of your website we can use LDAP server. CQ can interact with a LDAP server that stores user information centrally, which will eliminate the duplicate logins. This central server is then used to verify login information which enables you to realize Single Sign On.

1. **What is Authorization? How will you achieve basic authorization CQ5?**

**Authorization:** Authorization determines whether a user is allowed to take a particular action on specific areas of the site within the system. For example, a user can be authorized to read or update a specific page.

Basic authorization in CQ can be achieved using a series of entities:

**User:** A user accesses a system using their user account. The user account holds the details needed for accessing CQ; a key purpose of an account is to provide the information for the authentication and login processes – allowing a user to log in.

**Groups:** A group is a collection of users and/or other groups. A change in the permissions/privileges assigned to a group is automatically applied to all users in that group. A user does not have to belong to any group, but often belongs to several.

**Action:** Actions are performed on a resource. For example, a user can read, edit or delete a page, amongst other actions.

**Permissions:** Permission allows a user to perform an action on a given resource within the repository. Permissions are stored, and can be seen, at resource level within the repository.

**Privileges:** Privileges allow access to functionality available within the application; for example, replication of a specific path, or the ability to update the page hierarchy (including creating new pages).

**Resources:** Resources define the functionality to be accessed.

1. **What is the process to give privilege to a user for creating more user(s)?**

Give read, write, create, and modify permissions at “/home/users” node. This node is the place where all the users are stored.

1. **Is it possible to have user without any group?**

Yes, this is possible but not advisable. If you do so you will have to manage permissions for each user at user level. But if you add multiple users to a same group then permissions can be managed at group level not user level.

1. **What is clustering in CQ5? How to configure that ?**

Clustering is basically used for synchronization of data. Clustering enables multiple repository instances running on separate machines to be joined together to act as a single repository. When a write operation updates the data in one cluster node, that change is immediately and automatically applied to all other nodes in the cluster. This guarantees that any subsequent read from any cluster node will return the same updated result. CQ5 CRX is pre-loaded to run within a cluster, even when running a single instance. Hence the configuration of multi-node clusters with little effort happens in CQ5.

1. **What is the purpose of farm file**

The farm file contains a series of single-valued or multi-valued properties that control the behavior of Dispatcher. In farm files you can define “allow” and “deny” rules specific to your application, set client headers etc, path for vhost file. Each entry in /farm describes a separate server that can provide content; requests are routed to the servers by the dispatcher.

1. **What are all the security checks we need perform after the first installation of AEM?**

* Access Control Entry on the root node (/) for the principal everyone; this would allow everyone to read/write/etc content. If there is such an entry, the security level is low, and all users, including anonymous access, has read/write access to content.
* Before you use CRX productively, it is recommend to change the default passwords

(Using security console, using curl command).

* Restricting the server access using “Apache Sling Referrer Filter” service (restricting protocols, hosts and port numbers).
* Read and write access to files in an AEM installation folder or its backups must only be allowed to the operating system user under which the AEM instance is running.
* should use the user session, not the administrative session. This means you should use:

userSession =slingRequest.getResourceResolver().adaptTo(Session.class);

AdminSession = repository.loginAdministrative(null);

* Customizing Error Pages: Error pages can be customized for CQ. This is advisable so that the instance does not reveal sling traces on internal server errors.

1. **How will you achieve performance in CQ?**

**Performance Tips:**

1. Caching, loadbalancing and clustering
2. Use ArrayLists, HashMap etc instead of Vector, Hashtable etc where possible. Even better is to

use just arrays where possible.

1. Set the initial capacity of a collection (e.g. ArrayList, HashMap etc), StringBuffer /StringBuilder appropriately. This is because these classes must grow periodically to accommodate new elements. So, if you have a very large ArrayList or a StringBuffer, and you know the size in advance then you can speed up the things by setting the initial size appropriately.
2. Minimize the use of casting
3. Exception creation can be expensive because it has to create the full stack trace. An efficient way to do this is to declare a public static final Exception in your exception class itself.
4. Avoid using System.out.println and use logging frameworks like Log4J etc, which uses I/O buffers
5. Minimize calls to Date, Calendar, etc related classes.
6. If there is a repeating code within a loop, avoid creating new objects for each iteration. Create objects before entering the loop (i.e. outside the loop) and reuse them if possible.
7. Use lazy initialization when you want to distribute the load of creating large amounts of objects.
8. In Java, typically memory leak occurs when an object of a longer lifecycle has a reference to objects of a short life cycle. This prevents the objects with short life cycle being garbage collected. The developer must remember to remove the references to the short-lived objects from the long-lived objects. Objects with the same life cycle do not cause any issues because the

garbage collector is smart enough to deal with the circular references.

Unreachable collection objects can magnify a memory leak problem. In Java it is easy to let go of an entire collection by setting the root of the collection to null. The garbage collector will reclaim all the objects (unless some objects are needed elsewhere).

1. **How to improve Performance?**
2. While installing/starting CQ5 we can improve performance by increasing the Java heap size.

**Command:** Java –Xmx512m –jar cq-author-4502.jar –gui

1. **Caching :** On the web, clients can cache responses. Well-managed caching partially or completely eliminates some client–server interactions, further improving scalability and performance.
2. **Load Balancing:** Well Managed load balancing will improve the performance a lot.

Load balancing will be done using Performance statistics

1. Whenever possible, a project should first be soft-launched to a limited audience in order to gather real-life experience and perform further optimizations.

A performance "tune-up" or "health-check" should be scheduled at 6-12 months intervals.

1. **How will you communicate (two way) with external third party from CQ?**

REST full web services, Connecting LDAP servers using JMX

**Have you done any sort of integration of CQ with external applications?**

**Have you implemented any web service?**

1. **How to connect external DB from CQ?**

Some database vendors provide JDBC drivers in an OSGi bundle, for example [MySQL](http://www.mysql.com/downloads/connector/j/). If the JDBC driver for your database is not available as an OSGi bundle, obtain the driver JAR and wrap it in an OSGi bundle. The bundle must export the packages that are required for interacting with the database server. The bundle must also import the packages that it references.

**Connection Pool:** JDBC Connections Pool (com.day.commons.datasource.jdbcpool.JdbcPoolService) is a factory service. Go to Felix console 🡺 services 🡺 JdbcPoolService and configure the required properties like DB url, Driver class name, DB username and password

Sample code to connect:

<%@include file="/libs/foundation/global.jsp"%><%

%><%@page session="false"%><%

%><%@ page import="com.day.commons.datasource.poolservice.DataSourcePool" %><%

%><%@ page import="javax.sql.DataSource" %><%

%><%@ page import="java.sql.Connection" %><%

%><%@ page import="java.sql.SQLException" %><%

%><%@ page import="java.sql.Statement" %><%

%><%@ page import="java.sql.ResultSet"%><%

%><html>

<cq:include script="head.jsp"/>

<body>

<%DataSourcePool dspService = sling.getService(DataSourcePool.class);

  try {

     DataSource ds = (DataSource) dspService.getDataSource("hsqldbds");

     if(ds != null) {

         %><p>Obtained the datasource!</p><%

         %><%final Connection connection = ds.getConnection();

          final Statement statement = connection.createStatement();

          final ResultSet resultSet = statement.executeQuery("SELECT \* from INFORMATION\_SCHEMA.SYSTEM\_USERS");

          int r=0;

          while(resultSet.next()){

             r=r+1;

          }

          resultSet.close();

          %><p>Number of results: <%=r%></p><%

      }

   }catch (Exception e) {

        %><p>error! <%=e.getMessage()%></p><%

    }

%></body>

</html>

1. **What is persistence manager in cq5 and how will you implement it?**

CQ uses persistence manager to save the content to a persistent storage like file system or a database. By default the crx content is stored using Tar Persistence manager. It stores the content to file-system in standard linux archive files called tar. If you want to store the repository content in a database, then you can configure cq5 to use a database persistence manager.

1. **What are Clientlibs?**

AEM allows us to create Client-side Library Folders, which allow you to store your client side code in the repository, organized it into categories, and define when and how each category of code is to be served to the client. These folders are a nice way of keeping all of your JavaScript and CSS files in one location, and a way to call all the files with one line of code. Adv: Client server calls will be reduced.

1. **How to include client libraries while creating a new project? Or how do you include “css” and “js” files into your project?**

* Create a node of type cq:ClientLibraryFlolder.
* Assign the properties like

“name” : ”clientlib”

“sling:resourceType” : ”widgets/clientlib”,

“categories” : ”name which can be used to refer in JSP”

* In our JSP we can use this client libraries using a tag <cq:includeClientLib> To reference the libraries, we have to use “categories” attribute (i.e. the property inside cq:ClientLibraryFlolder node)

Ex: /etc/clientlibs/foundation/jquery node is of type cq:ClientLibraryFolder with “categories” property of value cq.jquery.

Usage in JSP page:

<cq:includeClientLib categories="cq.jquery"/>

**Note:** Dependencies: When the code in your client library folder references other libraries, identify the other libraries as dependencies.

The dependencies property informs CQ to include the CSS files and JQuery libraries in the page. The categories property informs CQ which clientlibs must be included.

<http://blogs.adobe.com/experiencedelivers/experience-management/clientlibs-explained-example/>

1. **How does Sightly differ from other templating systems?**

Sightly differs from other templating systems in three ways.

* Sightly is a valid HTML5 file.
* Sightly is secure by default because it automatically filters and escapes all variables being output to the presentation layer to prevent cross-site-scripting vulnerabilities.
* Sightly offers separation of concern because it is purposely limited to ensure that a real programming language is used to express the corresponding logic. This is done through the Use-API pattern.

1. **What is sightly?**

Sightly is an HTML templating language that was introduced in AEM 6.0. It is meant to take the place of JSP files and is the preferred templating system for HTML. The name sightly means “pleasing to the eye”, and its focus is to keep your markup beautiful and maintainable.

**What is Cross Site Scription(CSS)?**

Cross-site scripting (XSS) allows attackers to inject their code into web pages viewed by other users. CQ applies the principle of filtering all user-supplied content upon output. Additionally, a web application firewall can be configured to add protection.

However Adobe CQ by design doesn’t directly translate the user inputs into queries like SQL, so SQL injection is not possible in CQ as it doesn’t use any relational DB.

In AEM 6.0 the situation is different because of the new HTML template rendering tool Sightly, where all the request will go through the filter rather than developers call them into code.

1. **What is the approach to implement application which make compatible for Desktop/Mobile/Tablets (website and mobilesite)?**

We can achieve this by responsive design pages.

**How to implement etc/mapping?**

**What CQ version you have used? How to migrate cq4 to cq5?**

**We have some pages in JSP, we need to migrate all to CQ How can I do that?**

**What is an event Listener? How to implement explain?**

**What is cardinality explain its significance?**

1. **Have you worked on PreProcessors?**

If we want to handle some of the tasks before the replication process such as validations automatically, we can use preprocessors.

We can write a preprocessor by implementing **“com.day.cq.replication.Preprocessor”** interface and

overriding the method

public void preprocess(ReplicationAction replicationAction, ReplicationOptions

replicationOptions) throws ReplicationException

1. **What is an Event Handler how can you implement it? Tell me some use cases?**

Whenever an event occurs (Ex: page or asset creation/modification, replication..etc.), if we want to perform any action automatically then we can use Event Handlers.

Event Topics:

“com/day/cq/dam", "com/day/cq/wcm/core/page", "com/day/cq/replication".

We can write an event handler by implementing **“org.osgi.service.event.EventHandler”** interface and overriding the method **public** **void** handleEvent(Event event)

**UseCases:** Sending emails to users after successful activation.

Ex in code:

AuditEventHandler, ActivationEmailHandler

**import** org.osgi.service.event.EventHandler;

@Component(name = "com.cisco.wem.training.MyActivationEmailHandler", description = "Handler Configuration for Activation", label = "Activation Email Handler", specVersion = "1.1", metatype = **true**, immediate = **true**)

@Service

@Properties({ @Property(name = "service.pid", value = "com.cisco.wem.training.MyActivationEmailHandler"),

@Property(name = "event.topics", label = "Event Topics", description = "Please add event topics to be audited", value = { "com/day/cq/dam", "com/day/cq/wcm/core/page", "com/day/cq/replication" }, cardinality = 2147483647)})

**public** **class** MyActivationEmailHandler **implements** MyConstants, EventHandler {

**public** **void** handleEvent(Event event) {

**What is LDAP? How will you configure?**

1. **How to enable log messages in CQ?**

JSP page:

The initialization of a “Logger” object, called “log”, has already been accomplished during the inclusion of global.jsp in whatever Component you may be working on. Log files will be placed inside

“<cq-installdir>/crx-quickstart/server/logs” directory.

**Ex:**

log.info("Child page [{}] found.", child.getTitle());

Java file:

* Go to configuration tab of Felix console.
* Apache Sling Logging Logger Configuration.
* Click on ‘+’ and enter the fields as you required.
* Log will be created inside “<cq-installdir>/crx-quickstart/server/logs” directory

1. **How to enable the debugger in CQ/ How to start CQ in debug mode?**
2. Navigate to the directory “<cq-install-dir>/crx-quickstart/server” (where the CQ servlet engine (CQSE) has been installed) using your command line.
3. Run the following command.
   1. If windows: “server.bat -debug socket”
   2. If Linux/Unix: “./start –d”

We can also configure inside start.bat file (cq 5.5).

set CQ\_JVM\_OPTS=-server -debug -XX:MaxPermSize=256M -Xnoagent -Xmx1024M -Djava.compiler=NONE -Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=8080

**Call back methods in AJAX?**

**How will you read cookie in AJAX/Jquery?**

**What is the difference between Dropdown and Combobox?**

1. **In context editing?**

Edit the page makes changes then and there itself we can preview the changes push the content to publish. Text components can be edited directly on the webpage without intervening dialog box of explicit saving. Services can be registered and consumed inside VM.

**How to implement localization?**

1. **What is segmentation/campaign management and how to implement?**

Segmentation is used when you want to target your content towards the sites visitors in order to offer a more personalized site. This can be done by analyzing and characterizing a visitor’s activity on the website and their profile.

**Segment:** A segment is a collection of visitors that share certain traits.

**Trait:** A trait is a property of a visitor that can be used to determine membership in a specific segment.

**Implementaion:**

In our project we used segmentations to implement entitlements. We have created segment rules using segment editor. From login user profile we will get access levels and channel qualification values and segmentrule/path will be set on the page or asset. While displaying the page we will check the segmentpath of the page compare with respective segment rule if the login user is entitled one then we will show the page.

1. **What is Personalization?**

Personalization provides your users with a customized environment that displays dynamic content selected according to their specific needs. There is an ever-increasing volume of content available today, be it on internet, extranet, or intranet websites. Personalization centers on providing the user with a tailor-made environment displaying dynamic content that is selected according to their specific needs; be this on the basis of predefined profiles, user selection, or interactive user behavior. Teaser Component used in Personalization

**What is personalization? Why the Personalization is used (practical example)?**

**How will you achieve personalization?**

1. **What design patterns are used in AEM?**

Since AEM is built using OSGI, many of the design patterns for OSGI are valid. Some of these design patterns are Singleton (Service), Adapter Service, Resource Adapter Service, and Whiteboard.  
Also, since AEM is modular, you should be able to use any design pattern in your application.

**What is persistence manager and how to implement?**

1. **Can you create a page without a template?**

Pages are usually created by selecting a template for the page when you create them in the Website console. However, a page can be created manually in the CRXDE by using the same properties that are created when you make a page in the Website console. We can also create programmatically.

Ex: CRXDE under the content folder, create a node with the following type “cq:Page” and save it. Then add a child node with the following type “cq:PageContent”. Then you can add the required properties to show any components you want displayed on the page. It is generally not good practice to create pages manually.

1. **What are xtypes and how are they useful in AEM?**

In the ExtJS language an xtype is a symbolic name given to a class. In AEM these xtypes are widgets that used in the creation of components. AEM comes with a bunch of widgets that are available out of the box. You can also create and define your own xtype to be used in AEM.  
  
Ex: When you are creating your dialog for your component, and you want the author to be able to enter text, you would add a “cq:Widget” that has an “xtype” property of “textfield”.

1. **How do you add properties to the Page Properties dialog?**

You can add properties to your page properties dialog by copying the dialog located under “libs/foundation/components/page/dialog” and adding it to your page template. This will allow you to add new tabs and add new properties that you would like the author to choose from for pages that use that template. You could modify the original page dialog as well to get the same result, however it is not recommended.

1. **How do you move content from one server to another?**
   1. Package Manager
   2. Vault Sync
   3. Replication

**Are there any components that you have used in your project that are not relevant to an out of the box component?**

This is mostly an experienced based question that is likely to come up in one form or another. Be prepared to discuss these components in full detail by talking about how they functioned, your thought process behind them, etc.

**Project Experience:**

**Project Description: Webex**

The project is to redesign the existing cisco’s webex site i.e. [www.webex.com](http://www.webex.com) to AEM 6.0, for the ease of content authoring and managing digital assets.

##### *Contribution*

Built the entire new AEM platform for webex development.

* Involved in AEM 6.0 installations to split metada (SegmentNodeStore) and binaries (FileDataStore).
* Setup Cold stand by instance, Adobe AEM 6.0 new feature.
* Involved in configurations such as LDAP setup, Dispatcher, Replication, Reverse Replication and dispatcher flush agents.

1. **How to split metadata (SegmentNodeStore) and binaries (FileDataStore) in AEM while installing? And how to setup cold standby instance in AEM?**

**crx-quickstart 🡺 install 🡺 install.primary**

org.apache.jackrabbit.oak.plugins.blob.datastore.FileDataStore.config

path="./crx-quickstart/repository/filestore"

minRecordLength=I"4096"

org.apache.jackrabbit.oak.plugins.segment.SegmentNodeStoreService.config

customBlobStore=B"true"

standby=B"false"

org.apache.jackrabbit.oak.plugins.segment.standby.store.StandbyStoreService.config

org.apache.sling.installer.configuration.persist=B"false"

mode="primary"

port=I"8023"

primary.allowed-client-ip-ranges=["0.0.0.0-255.255.255.255"]

secure=B"false"

**crx-quickstart 🡺 install 🡺 install.standby**

org.apache.jackrabbit.oak.plugins.blob.datastore.FileDataStore.config

path="./crx-quickstart/repository/filestore"

minRecordLength=I"4096"

org.apache.jackrabbit.oak.plugins.segment.SegmentNodeStoreService.config

customBlobStore=B"true"

name="Oak-Tar"

service.ranking=I"100"

standby=B"true"

org.apache.jackrabbit.oak.plugins.segment.standby.store.StandbyStoreService.config

org.apache.sling.installer.configuration.persist=B"false"

mode="standby"

primary.host="173.37.124.104"

port=I"8023"

secure=B"false"

interval=I"5"

standby.readtimeout=I"1800000"

1. **What is LDAP? What are all its advantages? How it will be used in CQ?**

LDAP (the Lightweight Directory Access Protocol) is used for accessing centralized directory services and used to manage user accounts of multiple applications. One such LDAP server is Active Directory. LDAP is often used to achieve Single Sign On which allows a user to access multiple applications after logging in once.

User accounts can be synchronized between the LDAP server and CRX, with LDAP account details being saved in the CRX repository. This allows the accounts to be assigned to CRX groups for allocating the required permissions and privileges.

1. **Explain how to configure LDAP in AEM 6.0?**

To configure AEM with LDAP, we need to create three OSGi configurations:

1. An LDAP Identity Provider (IDP).
2. A Sync Handler.
3. An External Login Module.

**Configuring LDAP Identity Provider (IDP):**

The LDAP Identity Provider is used to define how users are retrieved from the LDAP server.

Go to <http://host:port/system/console/configMgr> and find for “Apache Jackrabbit Oak LDAP Identity Provider” name. click on “+”.



1. Click on + symbol to add a new LDAP identity provider.
2. Add the configuration values as below ( For Admin and For Author )
3. Click on Save

|  |  |
| --- | --- |
| LDAP Provider Name | Name of this LDAP provider configuration. |
| LDAP Server Hostname | Hostname of the LDAP server |
| LDAP Server Port | Port of the LDAP server |
| Use SSL | Indicates if an SSL (LDAPs) connection should be used. |
| Use TLS | Indicates if TLS should be started on connections. |
| Disable certificate checking | Indicates if server certificate validation should be disabled. |
| Bind DN | DN of the user for authentication. If this is left empty, an anonymous bind will be performed. |
| Bind Password | Password of the user for authentication |
| Search timeout | Time until a search times out |
| User base DN | The DN for user searches |
| User object classes | The list of object classes an user entry must contain. |
| User id attribute | Name of the attribute that contains the user id. |
| User extra filter | Extra LDAP filter to use when searching for users. The final filter is formatted like: '(&(<idAttr>=<userId>)(objectclass=<objectclass>)<extraFilter>)' (user.extraFilter) |
| User DN paths | Controls if the DN should be used for calculating a portion of the intermediate path. |
| Group base DN | The base DN for group searches. |
| Group object classes | The list of object classes a group entry must contain. |
| Group name attribute | Name of the attribute that contains the group name. |
| Group extra filter | Extra LDAP filter to use when searching for groups. The final filter isformatted like: '(&(<nameAttr>=<groupName>)(objectclass=<objectclass>)<extraFilter>)' |
| Group DN paths | Controls if the DN should be used for calculating a portion of the intermediate path. |
| Group member attribute | Group attribute that contains the member(s) of a group. |

Sample LDAP Identity Provider Configuration:



**Configuring Synchronization Handler:**

The synchronization handler will define how the Identity Provider users and groups will be synchronized with the repository. Go to <http://host:port/system/console/configMgr> and find for “Apache Jackrabbit Oak Default Sync Handler” name. Click on “+”.



The following configurations options are available for the Synchronization Handler:

|  |  |
| --- | --- |
| Sync Handler name | Name of the sync configuration. |
| User Expiration Time | Duration until a synced user gets expired. |
| User auto membership | List of groups that a synced user is added to automatically. |
| User property mapping | List mapping definition of local properties from external ones. |
| User Path Prefix | The path prefix used when creating new users. |
| User Membership Expiration | Time after which membership expires. |
| User membership nesting depth | Returns the maximum depth of group nesting when membership relations are synced. A value of 0 effectively disables group membership lookup. A value of 1 only adds the direct groups of a user. This value has no effect when syncing individual groups only when syncing a user’s membership ancestry. |
| Group Expiration Time | Duration until a synced group expires. |
| Group auto membership | List of groups that a synced group is added to automatically. |
| Group property mapping | List mapping definition of local properties from external ones. |
| Group path prefix | The path prefix used when creating new groups. |

Sample Sync Handler Configuration:



**Configuring External Login Module:**

The External Login Module is used to define which Identity Provider and Sync Handler to use, effectively binding the two modules. Go to <http://host:port/system/console/configMgr> and find for “Apache Jackrabbit Oak External Login Module” name. Click on “+”.



The following configuration options are available:

|  |  |
| --- | --- |
| JAAS Ranking | Specifying the ranking (i.e. sort order) of this login module entry. The entries are sorted in a descending order (i.e. higher value ranked configurations come first). |
| JAAS Control Flag | Property specifying whether or not a LoginModule is REQUIRED, REQUISITE, SUFFICIENT or OPTIONAL. Refer to the JAAS configuration documentation for more details around the meaning of these flags. |
| JAAS Realm | The realm name (or application name) against which the LoginModule is being registered. If no realm name is provided then LoginModule is registered with a default realm as configured in the Felix JAAS configuration. |
| Identity Provider Name | Name of the identity provider. |
| Sync Handler Name | Name of the sync handler. |

Sample External Module Configuration:



**Note:**

If you plan on having more than one LDAP configuration with your AEM instance, separate Identity Providers and Synchronization Handlers need to be created for each configuration.

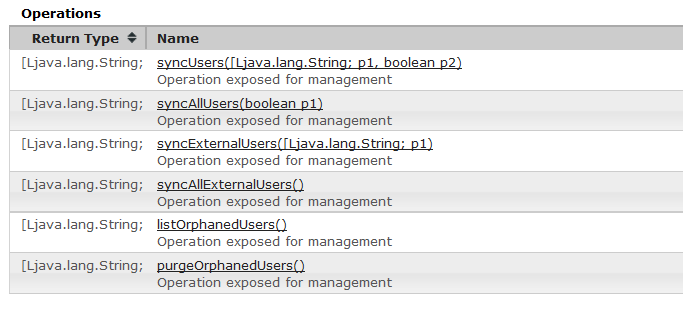
1. [**How does CQ interface with LDAP systems?**](https://helpx.adobe.com/experience-manager/using/faq-integrating-cq-ldap.html)

CQ interfaces with LDAP systems, such as Apache Directory or Windows Active Directory, using the Java Authentication and Authorization Service (JAAS).

1. [**How do I synchronize LDAP and CQ users?**](https://helpx.adobe.com/experience-manager/using/faq-integrating-cq-ldap.html)

**Steps:**

* Go to <http://host:port4502/system/console/jmx> console.
* Click on the sync handler that you have configured.
* There you can find several operations to sync the users and groups from LDAP.



1. **How do I configure CQ with LDAP for SSO?**

Configuring CQ with LDAP for Single Sign On (SSO) requires that you set trust\_credentials\_attribute="TrustedInfo" in ldap\_login.conf.

1. **What is “cq:EditConfig” what its significance?**

The edit behavior of a component is configured by adding a “cq:editConfig” node of type  “cq:EditConfig” to the component and by adding specific properties and child nodes.

It has some properties and child nodes to configure.

**Properties:**

cq:actions (String array): defines the actions that can be performed on the component.

**Ex:** Edit, insert, delete

cq:layout (String): : defines how the component is edited(floating).

**ChildNodes:**

cq:dropTargets (node type nt:unstructured): defines a list of drop targets that can accept a drop from an

asset of the content finder.

cq:listeners (node type cq:EditListenersConfig): defines what happens before or after an action occurs

on the component.

1. **What is the listener property in AEM?**

The listener property in AEM for a component is used to define what happens before or after an action on the component. This is added by using the “cq:listeners” node with a node type of “cq:EditListenersConfig”.

The listener property can also be added to any widget in AEM. In order to add a listener to that widget you just need to add node that is called “listeners” with a type of “nt:unstructured”. Then you just need to add child nodes to the “listeners” node that are events of that widget. You can find a list of events for each widget by searching CQ Widget API documentation.

1. **How to find out in which components a given xtype e.g. ‘checkbox’ is used in AEM?**

By using an Xpath query in CRXDE

/jcr:root/apps//element(\*, cq:Widget) [@xtype = 'checkbox']

1. **What Is Ecommerce?**

Transacting or facilitating business on the Internet is called ecommerce. Ecommerce is short for "[electronic commerce](http://ecommerce.about.com/od/Ecommerce-Book-Reviews/fr/Electronic-Commerce-By-Gary-Schneider.htm)." Buying or selling online. Any form of business transaction conducted electronically is ecommerce.

Basic demonstrated purpose AEM provided around 12 to 15 components like products display, promotions, vouchers, shopping cart, checkout…etc. They can be stored in the CRX repository and retrieved using internal APIs to show the ecommerce capabilities. But it is not recommended approach from AEM itself, said that we need to integrate with enterprise level ecommerce engine. AEM provided a bunch of interfaces/APIs through which we can integrate with enterprise scale ecommerce engine such as Hybris, elastic path…

The eCommerce integration framework has been built to allow you to easily integrate eCommerce systems with AEM. This allows you to connect with a purpose built eCommerce system to control product data, shopping carts, checkout and order fulfillment, while AEM controls the data display and marketing campaigns.

**REST Services Implementation**

Webservice URL

http://{host:port}/c/wcm/servlets/services/wem.activatecontent.json

/c/wcm/servlets/services/wem.activatecontent.json

WEMServicesServlet

1. **What is externalizer in CQ?**

In AEM, the Externalizer is an OSGI service that allows you to programmatically transform a resource path (e.g./path/to/my/page) into an external and absolute URL (for example,  http://www.mycompany.com/path/to/my/page) by prefixing the path with a pre-configured DNS.

String assetPath = “/content/en/us/support/testPage”;

Externalizer externalizer = resourceResolver.adaptTo(Externalizer.class);

String authorLink = externalizer.authorLink(resourceResolver, assetPath);

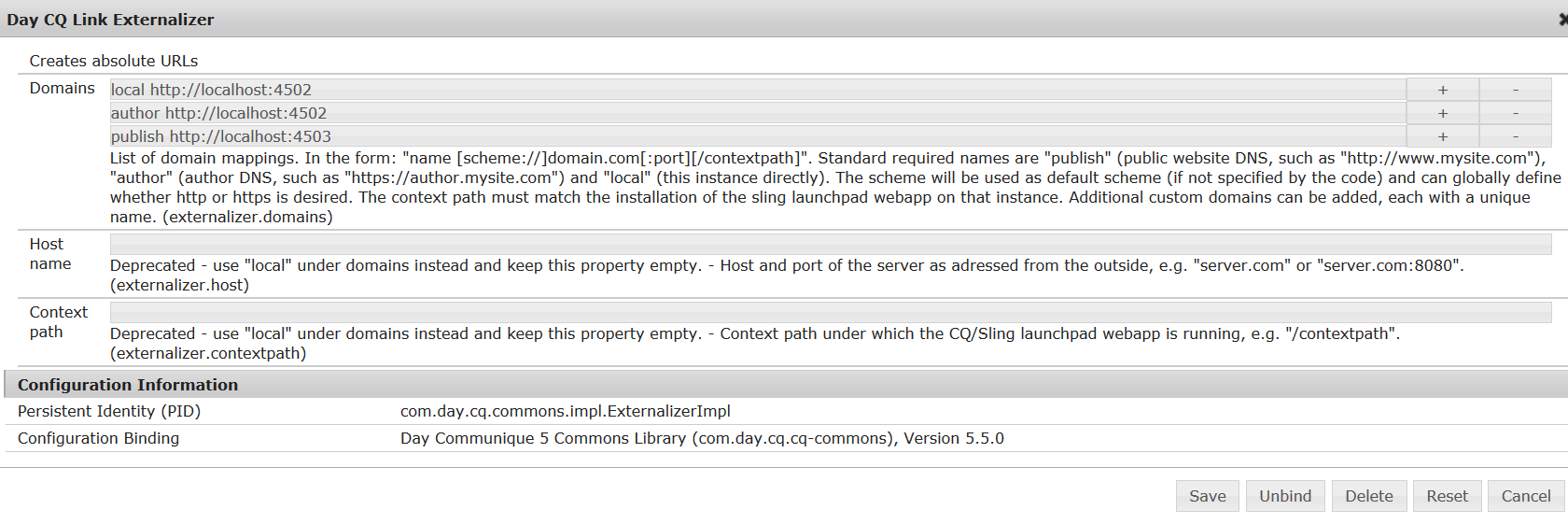
String publishLink = externalizer.publishLink(resourceResolver, assetPath);

It will return “authorLink” as <http://localhost:4502/c/en/us/support/testPage>;

“publishLink” as <http://localhost:4503/c/en/us/support/testPage>;

To define a domain mapping for the Externalizer service:

* Go to the Adobe CQ5 Web Console, select the Configurations tab.
* Click Day CQ Link Externalizer to open the configuration dialog.
* Define a domain mapping: a mapping consists of a unique name that can be used in the code to reference the domain, a space and the domain: <unique-name> [scheme://]server[:port][/contextpath]



1. **What is JSON?**

JSON, or JavaScript Object Notation, is a minimal, readable format for structuring data. It is used primarily to transmit data between a server and web application, as an alternative to XML.

Keys and Values

The two primary parts that make up JSON are keys and values. Together they make a key/value pair.

**Key:** A key is always a string enclosed in quotation marks.

**Value:** A value can be a string, number, boolean expression, array, or object.

**Key/Value Pair:** A key value pair follows a specific syntax, with the key followed by a colon followed by the value. Key/value pairs are comma separated.

Ex: "name" : "Srikanth"

**Arrays:** An Array is indicated by square brackets.

**Objects:** An object is indicated by curly brackets. Everything inside of the curly brackets is part of the object.

Ex**:**

"collection" : {"title" : "Blog", "categories" : [ "Category-1", "Category-2" ]}

Here Collection is an Object and Categories is an array



.infinity.json

Debug=layout

1. **What is Maven explain about it?**

<https://www.youtube.com/watch?v=al7bRZzz4oU>

Maven is the one of the popular built tool similar to ant. It makes the developers life very easy.

It is used as Project management tool, Dependency management tool.

It will take care of:

* Defines the project structure.
* Including multiple jars.
* Maintain the dependencies and versions among the jars.
* Building, publishing and deploying

Commands:

mvn archetype:generate

mvn compile

mvn package

mvn install

<plugins>

<plugin>

<groupId>org.apache.felix</groupId>

<artifactId>maven-scr-plugin</artifactId>

<version>1.7.4</version>

</plugin>

<plugin>

<groupId>org.apache.felix</groupId>

<artifactId>maven-bundle-plugin</artifactId>

<version>2.3.6</version>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<!-- use version 2.3.2 to have java 1.5 as the default -->

<version>2.3.2</version>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-resources-plugin</artifactId>

<version>2.5</version>

</plugin>

<plugin>

<groupId>org.apache.sling</groupId>

<artifactId>maven-sling-plugin</artifactId>

<version>2.0.6</version>

</plugin>

<plugin>

<groupId>com.day.jcr.vault</groupId>

<artifactId>content-package-maven-plugin</artifactId>

<version>0.0.23</version>

<configuration>

<failOnError>true</failOnError>

<outputDirectory>${crx.target}</outputDirectory>

</configuration>

</plugin>

<plugin>

<groupId>org.mortbay.jetty</groupId>

<artifactId>jetty-maven-plugin</artifactId>

</plugin>

<plugin>

<artifactId>maven-war-plugin</artifactId>

<version>2.3</version>

<configuration>

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

</plugins>

1. **When would you use GET and POST requests in JS?**

Different in length limitation, security, caching and a few others. We should only use GET for idempotent requests, it means requests that don't make significant changes in the backend system or database but if you do need to make inserts, updates or deletes in a database, trigger emails or any other major action, POST is recommended.

**Site-Admin Console Customizations:**

/apps/wcm/core/content/siteadmin/actions/myactivate

Handler 🡺 CQ.wcm.SiteAdmin.myActivatePage

SiteAdmin.Actions.js

CQ.wcm.SiteAdmin.activatePage

internalActivatePage

/etc/workflow/models/scheduled\_activation/jcr:content/model

Call CQ.wcm.SiteAdmin.myActivatePage function

**ActivateNow:**

/apps/wcm/core/content/damadmin/actions/activate/menu/activateNow

Handler: CQ.wcm.SiteAdmin.activatePage

/apps/cq/ui/widgets/source/widgets/wcm/SiteAdmin.Actions.js

CQ.wcm.SiteAdmin.activatePage = **function**();

CQ.wcm.SiteAdmin.internalActivatePage = **function**(paths, callback);

CQ.HTTP.post(CQ.shared.HTTP.externalize

("/c/wcm/servlets/wemreplication.activateDeactivateLater.json"),

callback,

{

"\_charset\_" : "utf-8",

"assetpath" : paths,

"stopworkflow" : "yes",

"absoluteTime" : **new** Date().getTime(),

"cmd" : "Activate",

"Scheduled" : "yes"

}, **this**, **true**);

/c/wcm/servlets/wemreplication

WemBookAssetReplicationServlet

**ActivateLater:**

CQ.wcm.SiteAdmin.scheduleForActivation

CQ.wcm.SiteAdmin.internalScheduleForActivation = **function**(date, paths);

CQ.wcm.SiteAdmin.internalScheduleForActivation = **function**(date, paths) {

**var** admin = **this**;

**var** scheduleRequired = **false**;

// call post servlet activate the selected item

**var** callback = **undefined**;

**var** params = {

"\_charset\_" : "UTF-8",

"model" : "/etc/workflow/models/scheduled\_activation/jcr:content/model",

"absoluteTime" : date ? date.getTime() : **new** Date().getTime(),

"payload" : paths,

"payloadType" : "JCR\_PATH"

};

CQ.wcm.SiteAdmin.myActivatePage = **function**() {

console.log("Inside CQ.wcm.SiteAdmin.myActivatePage function");

**var** admin = **this**;

**var** paths = [];

**var** selections = **this**.getSelectedPages();

**for** ( **var** i = 0; i < selections.length; i++) {

paths.push(selections[i].id);

admin.mask();

// Added to activate all the selected assets even if there is an exception for any of the selected assets

CQ.wcm.SiteAdmin.myInternalActivatePage.call(admin, selections[i].id);

}

};

CQ.wcm.SiteAdmin.myInternalActivatePage = **function**(paths, callback) {

Q.HTTP.post(CQ.shared.HTTP.externalize("/c/wcm/servlets/myreplication.activateAsset.json"),

callback, {

"\_charset\_" : "utf-8",

"isMultiple" : isMultiple,

"assetpath" : paths

}, **this**, **true**);

com.cisco.wem.training.MyReplicationServlet

1. **What is Jenkins what are all its advantages?**

**Jenkins**

Jenkins provides an easy-to-use continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. The automated, continuous build increases the productivity.

**Features:**

1. **Easy installation**: Just java -jar jenkins.war, or deploy it in a servlet container. No additional install, no database.
2. **Easy configuration**: Jenkins can be configured entirely from its friendly web GUI with extensive on-the-fly error checks and inline help. There's no need to tweak XML manually anymore, although if you'd like to do so, you can do that, too.
3. **Change set support**: Jenkins can generate a list of changes made into the build from Subversion/CVS. This is also done in a fairly efficient fashion, to reduce the load on the repository.
4. **Permanent links**: Jenkins gives you clean readable URLs for most of its pages, including some permalinks like "latest build"/"latest successful build", so that they can be easily linked from elsewhere.
5. **RSS/E-mail/IM Integration**: Monitor build results by RSS or e-mail to get real-time notifications on failures.
6. **After-the-fact tagging**: Builds can be tagged long after builds are completed.
7. **JUnit/TestNG test reporting**: JUnit test reports can be tabulated, summarized, and displayed with history information, such as when it started breaking, etc. History trend is plotted into a graph.
8. **Distributed builds**: Jenkins can distribute build/test loads to multiple computers. This lets you get the most out of those idle workstations sitting beneath developers' desks.
9. **File fingerprinting**: Jenkins can keep track of which build produced which jars, and which build is using which version of jars, and so on. This works even for jars that are produced outside Jenkins, and is ideal for projects to track dependency.
10. **Plugin Support**: Jenkins can be extended via 3rd party plugins. You can write plugins to make Jenkins support tools/processes that your team uses.
11. **How will you upgrade to AEM 6.0?**

There are two distinct steps to doing a full upgrade:

1. [**In-Place Upgrade**](http://docs.adobe.com/docs/en/aem/6-0/deploy/upgrade.html#par_title_4)**:** Drop in the new aem-6.0 jar file right next to your existing <crx-quickstart> directory and launch it. This upgrades the system, but leaves the existing CRX2 repository as it is.
2. [**Migrating the Repository to Oak**](http://docs.adobe.com/docs/en/aem/6-0/deploy/upgrade.html#par_title_7)**:** The AEM 6.0, backend has moved to OAK to enhance scalability and performance of the content repository. Once the in place upgrade is complete.

There are two options for the storage backend used by Oak:

TarMK: A microkernel that uses the Tar persistence system for storage.

MongoMK: A microkernel that uses MongoDB for storage.

Adobe provides a tool that can be used to migrate the repository to the new OAK implementation.

CRX2OAK repository migration tool (<crx-quickstart>\opt\helpers\crx2oak.sh).

1. **Explain about GIT?**

GIT is a very popular and efficient open source Version Control System. GIT allows a team of people to work together, all using the same files. It makes sure that everyone on the same page.

**Features:**

* Tracks the content such as files and directories.
* Branch switch, pull, push.
* Merge the changes easily.
* We can see the commit history of the file what the changes were done in particular commit.
* Compare the files with earlier versions.

The Eclipse IDE has excellent support for the GIT version control system. This support is developed via a set of plug-ins from the *EGit* project.

<http://www.vogella.com/tutorials/EclipseGit/article.html#egitscope>

1. **Difference between SVN and GIT?**

|  |  |
| --- | --- |
| **SVN/CVS (ControlVersionSystem)** | **GIT** |
| 1. Complex servers to be installed. 2. Uses two tree architecture.  * Repository * Working Directory  1. As it maintains only Repository and working Directory If you don't have internet at the moment, you cannot commit. If you want to make a copy of your code, you have to literally copy/paste it. | 1. Complex servers are not required to install. 2. Uses three tree architecture  * Repository * Staging Index * Working Directory  1. Your local copy is a repository (staging index), and you can commit to it and get all benefits of source control. When you regain connectivity to the main repository, you can commit against it. |

Cisco.com

We have used maven as built tool, GIT for version controlling and Jenkins for continuous integration.

Customized Site-Admin and Dam-Admin consoles of AEM

CISCO has various/lots networking products.

Routers, Switches, Webex, modems, teleconferencing...etc

an end user will check for features of the products, compare the models with others check for the support/warranty they are going to get.

all such kind of information will be provided on cisco.com.

Cisco has lots of products, to market all the products it will provide such product information on cisco.com we have around 3 million of pages as of now.

to categorize them we used CQ tagging extensively. so for a page to exist on cisco.com it should mapped to atleast one concept and doctype.

Cocept: what level/type of concept(product) it is

category: grouping similar products in to a category(classification)

series: similar feature models into a series

Doctype: to identify type the document

marketing documentation

Promotions & and marketing : Presale content

Support to the customer : postsale/support content

for each product on cisco.com we will have both marketing and support content

Autogen: will generate pages based on doctype hierarchy for any concept till LLL

HMP: Hierarchy Management Program

will create all the concepts and doctypes and uploaded into WEM

concept hierarchy.

Category

SubCategory

Series

Model

Doctype hierarchy.

Series

HLL

LLL

EOT

**Project Description: ANG**

As part of this engagement, J.P. Morgan Treasury Services wanted to redesign Documentum based private portal named 'ACCESS Community' to ANG i.e. Access Community for the Next Generation of J.P. Morgan using Adobe Day CQ5. The next generation of J.P. Morgan ACCESS (an online, electronic banking portal for corporate clients) allows clients to utilize a new community feature that helps to educate users about the new platform via help-articles, products, user guides, FAQ’s, videos and podcasts. Most content stored in CQ for ANG in its English branch will need to be translated into different languages. A third party vendor, Transperfect is used to provide the translation service. The ANG user will have the ability to send feedback to Community administrators on any of the products. After the user completes the entries in the feedback form and the all validations are completed, user input will be posted to a pre-defined email address. The ANG user's first name and last initial needs to be displayed on every page, this user information should be getting by calling an external web service.

HighLevel APIs:

ComponentContext

Component

CurrentDesign

CurrentPage

CurrentNode

CurrentStyle

Designer

EditContext

PageManager

PageProperties

Properties

Resource

ResourceDesign

ResourcePage

1. Sling Servlets
2. OSGI Services
3. EventHandlers
4. OSGI Listeners
5. Schedulers
6. Workflow Processes
7. Pre Processes
8. Webservices
9. APIs(PageManager, TagManager, Sling, ResourceResolver, Session)
10. Integration with external services
11. Configuring OSGI dynamic properties
12. Creating a sample project with dialogs, design dialogs, widget, component, client libraries, i18n
13. My restfull Webservices to support upload/edit/activate content into CQ
14. Custom workflows
15. Design Patterns
16. Agents(Rep, Rev and Dispatcher flush)
17. LDAP config
18. Disp config
19. Custom complex component Ex: community carousel component
20. Users and Groups
21. Curl command
22. [How to show/edit custom properties on the properties page in AEM](http://stackoverflow.com/questions/31383230/how-to-show-edit-custom-properties-on-the-properties-page-in-aem)
23. segmentations
24. Components &Dialogues
25. Client Libraries
26. JavaScript
27. ExtJS
28. Override the default activate button
29. /apps/wcm/core/content/damadmin/actions/activate/menu
30. CQ.wcm.SiteAdmin.activatePage
31. Damadmin.actions.js, Siteadmin.actions.js
32. Build a training aem project
33. MSM/localization
34. Creating a custom component ratings&comments component
35. Xpath queries

[Adobe CQ Help](https://helpx.adobe.com/experience-manager/topics.html) /

Integrating the JQuery Framework into Adobe CQ

<https://helpx.adobe.com/experience-manager/using/integrating-jquery-framework-cq.html>

1. **Any custom components used which not relevant to out of box component?   
   16. How you can inherit properties of one dialog to another dialog?   
   17. What is the approach to implement application which make compatible for Desktop/Mobile/Tablets?   
   18. How to implement etc/mapping?**
2. **Where design dialogue data will store?**
3. **Have you used search component?**
4. **Social Integration components   
   Tags, featuers.   
   Pagination / how do you cache?   
   What is Resource resolver ?   
   Event Listeners / Schedulers ?    
   How to create a custom widget client-library?  
   Can you tell me that how can we registered the newly created servlet in Activator.java?  
   Clustering in CQ5**
5. **mobile site application -**
6. **blue print and live copy**
7. **multilingual site**
8. **what are the devices support mobile**
9. **how will u upload jar file through felix console**
10. **how will u create a user**
11. **is it possible to have user without any group?**
12. **how will u give privileges to user for creating the user?**
13. **what are the steps for creating custom components**
14. **Maven Integration with Eclipse?**
15. **What is the difference between Ant and Maven tools ?**
16. **Dependency Injection ?**
17. **Why design\_dialog content stores under etc/designs/project namespace? How it recognizes to store there itself ? Can we store in some other place ?**
18. **How comfortable are you with jQuery ?**
19. **Do you interest in jQuery or EXT JS ? Why ?**
20. **Query Builder and its API ?**
21. **Have you ever worked on DAM API ? Explain one Scenario ?**
22. **What features of CQ do you like most ? Why ?**
23. **What Did you do in Admin Level ?**
24. **Have you created any out of the box buttons ?**
25. **What are the limitations of CQ ?**
26. **What is a personalization ?**
27. **What are the servlets that you will extend in CQ ?**
28. **How do you covert a class to a service ?**
29. **What servelts do you implement in CQ ?**
30. **How to register a servlet in CQ ?**
31. **Do you have any  idea how comments component works ?**
32. **How to run a workflow ?**
33. **What are the differences between CQ 5.4 and 5.6.1 ?**
34. **What components did you feel complex in your project ?**
35. **Have you ever worked with Social collaboration in CQ ?**
36. **How to add the tab in the Page Properties dialog.**
37. **Under page properties, basic tab, what is the use of On time and Off time.**
38. **Can you explain me how the Logo component works.**
39. **How to create a page from the template.**
40. **Have you worked on xtypes.**
41. **Can you tell me how dispatcher works and if I have only 1 publish server**
42. **How would you architect an AEM application?  
    http://www.terrabeata.com/2015/03/aem-interview-question-how-would-you-architect-an-aem-application/**
43. How is localization implemented in AEM? **http://www.terrabeata.com/2015/03/how-is-localization-implemented-in-aem/**
44. How is access restricted or allowed to assets within AEM’s repository?
45. What is the best way to perform redirects or URL rewrites?
46. How is data stored? What is the Tar Persistence Manager? What is the TarMK? What is the MongoMK? How are they different? When should each one be used?
47. Have you created any custom components?
48. Have you created any servlets?
49. How can one component dialog inherit the properties from another component dialog?
50. How is a web site implemented for a desktop browser vs. a smart phone or a tablet?
51. From the original site: “How to implement the etc/display?” I’m not sure what this is about.
52. How can resources be ordered in the repository?
53. How to implement localization?
54. How to implement etc/mapping ?

**Users and Groups**

Content personalization and content targeting

**Tasks:**

1. **Creating a custom component (E2E flow)**
2. **Ratings & Comments component functionality (E2E flow)**
3. **Implementing a business functionality E2E flow (ActivateLater button functionality)**
4. **Implementing a REST Web service E2E flow (Upload/Activate content service)**
5. **Implementing a Workflow E2E flow**
6. **Minimum basics of JS, JQuery and CSS**

There are some basic initial configurations we can set to get a better performance out of our CQ instances.  An average performance target value can be defined. Once this value is proven to be both achievable and maintainable, it can be used to monitor the performance of the website and indicate the development of potential problems.

1. Optimize Tar Files: As data is never overwritten in a tar file, the disk usage increases even when only updating existing data. When optimizing, the TarPM copies data that is still used from old tar files into new tar files and deletes the old tar files that contain only old or redundant data. If there is only one file, optimization will have no effect. By default optimization is automatically run each night between 2 am and 5 am. and alternatively you can start optimizing the tar file manually.
2. Merge Tar Index Files: If many entries are stored in the tar files, the number of index files may grow. The index files are automatically merged before and after the scheduled Tar PM optimization. To reduce the number of index files at other times, you can merge these index files manually through the JMX console.
3. Consistency Checks: The Tar PM can check repository consistency and fix consistency problems at startup.
4. Run Garbage Collector for Data Store: When a conventional WCM asset is removed, the reference to the underlying datastore record may be removed from the node hierarchy, but the datastore record itself remains.
5. Check DataStore Consistency: CRX can check the consistency of the datastore to confirm that objects expected to be available in the datastore are actually there. Any datastore inconsistencies found will be logged.
6. CRX search index’s resultFetchSize param: The SearchIndex element specifies the configuration of the search engine. By default the search index is managed by the text search engine Apache Lucene.
7. TarJournal File Maximum size: Maximum size of a single journal file before it will get rotated. Default: 104857600 bytes (100 MB).
8. Set logging level: By default logging level is set to INFORMATION. For performance purposes it can be switch to ERROR and set back to INFORMATION or DEBUG as required (logging levels: trace -> debug -> info -> warn -> error).
9. Apache Sling JSP Script Handler: To improve performance you should disable as much as possible. In particular for production instances: disable Generate Debug Info, disable Keep Generated Java, disable Mapped Content, disable Display Source Fragments.
10. Apache Sling Java Script Handler: Configure settings for the compilation of .java files as scripts (servlets). for production instances: Source VM and Target VM, define the JDK version as that used as the runtime JVM, disable Generate Debug Info.

This is the last part of our series of blogs discussion “Gold-Copy”. Remember to keep track to all configurations so you know what has been set, when it was set and who is accountable for it.

Every new developer should be given a copy of “Gold-Copy” and a set of instructions on how to install other tools such as JDK, Eclipse, Maven and Vault. Whenever you set new configurations you should share across all team members via CRX Packages. Should you restore a running production instance you can deploy your gold-copy and install CRX Packages for code and content.