01. What is CMS?

- A content management system, often abbreviated as CMS, is software that helps users create, manage, and modify content on a website without the need for specialized technical knowledge.
- In simpler language, a content management system is a tool that helps you build a website without needing to write all the code from scratch (or even know how to code at all).
- Instead of building your own system for creating web pages, storing images, and other functions, the content management system handles all that basic infrastructure stuff for you so that you can focus on more forward-facing parts of your website.

02. Can you list out Market leading CMS software's?

- Adobe AEM
- Sitecore
- HubSpot
- Drupal,
- WordPress
- Joomla
- OpenText
- Liferay
- Oracle WebCenter Content

03. What is AEM?

Adobe Experience Manager (AEM), is a java-based content management system that is offered from Adobe. It was previously called Day CQ5 but it was acquired from Adobe in 2010. AEM is based on a content repository and uses the JCR to access the content in the repository. AEM uses the Apache Sling framework to map request URL to the corresponding node in the content repository. It also uses the OSGI framework to internally allow modular application development.

04. 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.
- Implementation of workflows for creating, editing and publishing of content.
- Managing a repository of digital assets like images, documents and integrating them to the websites.
- Usage of search queries to find content no matter where it is stored in your organization.
- Setting up social collaboration blogs, groups.
- Tagging utility to organize the digital assets such as images.

05. Explain about architecture / Technology stack of AEM?

Adobe CQ5 uses the following technologies:

- **JCR –** Java specification for accessing a content repository JSR-283 specification jcr 2.0 , cq5 uses its own implementation of JCR called CRX. Apache Jackrabbit is an open-source implementation of JCR 2.0 specification.
- **Apache Sling** – RESTful framework to access a JCR over http protocol. It maps the request URL to the node in JCR.
- **OSGi(Apache Felix)** – Framework for modular application development using java. Each module called bundle can be independently started and stopped. OSGi container which provides implementation classes for OSGi framework.

06. What is the difference between the Author and Publish environment?

Every 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.

- Author instance is for development and content creation purposes.
- WCM module would be in enabled stage in author instance that’s why content author can edit/create pages in author instance, in publish environment WCM module in disabled state and can’t edit any content
- The author instance is where you will administer your site as well. The author's environment is usually kept behind a firewall.
- The publish instance is where you will make your content available for your targeted audience. Publish server is live server and it serves response to end user request.

07. What are the differences between parsys and iparsys?

parsys – It is a placeholder called “Paragraph System”, where we can drag and drop or add other components or scripts at page level.
iparsys – It is inherited paragraph system, it is similar to parsys except that it allows to inherits parent page “paragraph system”.

08. 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. For example, in 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.

09. 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), MVC, Adapter Service, Resource Adapter Service, and Whiteboard. Also, since AEM is modular, you should be able to use any design pattern in your application.

10. What are the features of AEM?

The features of AEM are:
a

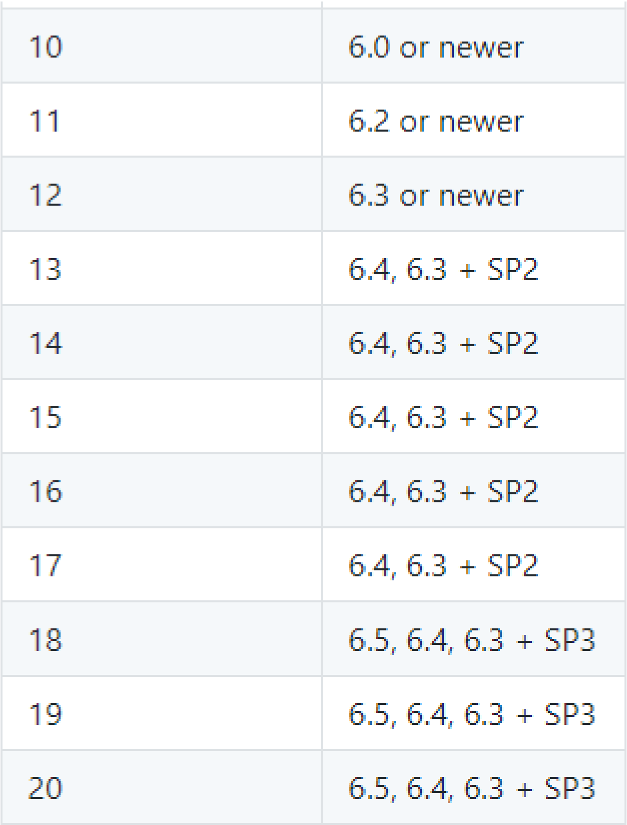
- It is easier to navigate through the different assets, pages, templates and elements in the web page.
- The new template editor has many features that allow the users to create and manage their personalized templates. Drag and drop, easy to make forms, along with HTML 5 functionality make the process easier.
- The Multi-Site Manager allows the seamless launching of online campaigns globally, that can be very useful for international brands. The local markets and customers can be targeted easily.
- The new version of AEM has associated with Livefyre, to provide better content development and social media marketing.
- It has a unified User Interface that allows better personalization for all.

11. How to create maven project structure for new project

Maven archetype creates a basic AEM maven project structure, while creating you will be asked few questions like project name, AEM version etc.

We can find compatible archetype for your AEM version from below link

https://github.com/adobe/aem-project-archetype/blob/master/VERSIONS.md



10	6.0 or newer
11	6.2 or newer
12	6.3 or newer
13	6.4, 6.3 + SP2
14	6.4, 6.3 + SP2
15	6.4, 6.3 + SP2
16	6.4, 6.3 + SP2
17	6.4, 6.3 + SP2
18	6.5, 6.4, 6.3 + SP3
19	6.5, 6.4, 6.3 + SP3
20	6.5, 6.4, 6.3 + SP3

12. Explain about filter.xml

Filter.xml defines what parts of the JCR repository are imported or exported during content package creation/installation.
The filter.xml consists of a set of filter elements, each with a mandatory root attribute and an optional list of include and exclude child elements.

```
        <workspaceFilter version="1.0">
          <filter root="/apps/project1" />
          <filter root="/etc/project1">
            <exclude pattern=".*\.gif" />
            <include pattern="/etc/project1/static(/.*)?" />
          </filter>
          <filter root="/etc/map" mode="merge" />
</workspaceFilter>
```

13. Explain about maven bundle plugin

Maven bundle plugin (BND) can be used to create OSGI bundle by passing metadata tags like import-package, export-package, private-package, bundle-symbolic-name etc.

Metadata would be useful to felix container to manage bundle lifecycle.

```
<plugins>

  <plugin>

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

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

    <extensions>true</extensions>

    <configuration>

      <instructions>

        <Export-Package>org.foo.myproject.api</Export-Package>

        <Private-Package>org.foo.myproject.*</Private-Package>

        <Bundle-Activator>org.foo.myproject.impl1.Activator</Bundle-Activator>

      </instructions>

    </configuration>


  </plugin>

</plugins>
```

14. How to sync (check in/out) code between eclipse (file system) and AEM

- Eclipse AEM developer plugin and VLT tool (fileVault) both can be used to sync code between file system and AEM.

- Eclipse AEM developer plugin directly provided two option to import and export code from server and it is very easy to compare with fileVault (VLT tool)

15. Explain about FileVault (VLT tool)

The Jackrabbit FileVault tool (VLT) is a tool developed by The Apache Foundation that maps the content of a Jackrabbit/AEM instance to your file system. The VLT tool has similar functions as source control system client (such as a Subversion (SVN) client), providing normal check-in, check-out and management operations, as well as configuration options for flexible representation of the project content. VLT (fileVault) provided different commands to sync code between file system and AEM.

16. What is new in AEM 6.1?

- Analytics & Target Integration
- Tags in Touch UI
- Workflows: The Workflows system is now more fully integrated within the Touch UI instead of only being in the Classic UI.

17. What is new in AEM 6.2?

- **AEM Content Fragment Enhancements:** Rich Text page editor in AEM Touch UI has been enhanced to support Content Fragments. Content Fragment a name suggests separates the content into fragments that can be reused and provides the ability to manage text-based content and its associated assets.
- **Template Editor and Policies AEM Site Enhancement:** Templates are used in AEM to provide consistent page structure throughout the project. Template console and Editor is one of the main features that is shipped with AEM 6.2.
- **User Interface Improvements:** Navigation has been moved from side rail to overlay, to provide more viewable area for actual work.
- Support for Mongo DB Enterprise 3.2 is added as a part of repository enhancement.
- Package replication can be done from package manager itself. No need to activate individual pages.

18. What is new in AEM 6.3?

- **Core Components:** Introduction to Production-ready components:- Faster time-to-market with Adobe-vetted, production-ready components based on best practices.
- **Experience fragments:** Composition and delivery of experiences to both owned web properties and affiliated properties, like Pinterest and Facebook.
- **Performance and scale enhancements:** Empower your teams to work more productively and efficiently with a broad range of enhancements to support large scale operations—including bulk ingestion, workflow processing, video processing and encoding, handling large files, repositories, and more.
- **Enhanced Assets Search** – New Search Boost to add keywords to the asset, multi-value search predicate, tag predicate.
- New side-by-side comparison of images in Assets' Review Task.
- **NEW Check-in/Check-out** – Let a user make changes to an asset while preventing other users from modifying the asset.

19. What is new in AEM 6.4?

- **Style System to template authoring:** Style System allows template authors to define styles for component by simply selecting different type of classes from drop down. This will eliminate the need for creating different components for different style and made components more flexible and adaptable to need of content authors.
- **Structured Content Fragments:** Structured content fragment is again a nice enhancement that has been introduced in aem 6.4. Now we we can define content fragment model using which we can create content fragments. This will help content authors in restricting what amount or data they want to share with other third party applications.
- **TouchUI Workflow Editor:** Workflow Editor is moved from Classic UI to Touch UI. The overall functionality of the Workflow Editor remains the same. One thing that i like most is once you edit an OOOB workflow , its configurations

automatically moves to /conf. This allow functional users to edit/customize a workflow without the need of an aem developer (for small changes).

- **Assets Enhancement:** There are many enhancements that has happened on aem assets side, few of the major one's are:
    - **Metadata Export and Import** –> Import and export metadata in simple csv format.
    - **Asset Reports** –> Generate a detailed report based on selected criteria.
    - **AEM Smart Tagging**
- **Experience Fragment Export to Adobe Target:** Once adobe target is configured under cloud configuration. Content authors who has create experience fragment now has an option to export them directly to adobe target by simply a button click.

20. What is new in AEM 6.5?

- **Extension towards "Headless CMS"**: AEM is moving towards a 'Headless CMS' approach to improve flexibility and help organizations reuse and distribute content across multi channels. The enhanced flexibility will also help IT teams to be more creative in using AEM with various systems and frameworks.
- **Experience Fragment export to Adobe Target:** Experience Fragments created in AEM can now be exported to Adobe Target in either HTM or JSON formats and used in driving Target activities. This lets you combine the ease-of-use and power of AEM, with the powerful Artificial Intelligence (AI) and Machine Learning (ML) capabilities in Target.
- **Content Fragments DIFF View & Annotations:** Content Fragments in AEM 6.5 haven been made more powerful by adding editorial content governance. Now, content can be reviewed across versions with a side-by-side view to check the differences. Another new enhancement is the ability to annotate text in the Content Fragment editor.
- **Core Components:** These are a set of standardized components to accelerate the creation of a new site or extend an existing site in AEM. These

components are production-ready that are configurable and very easy to customize if required.

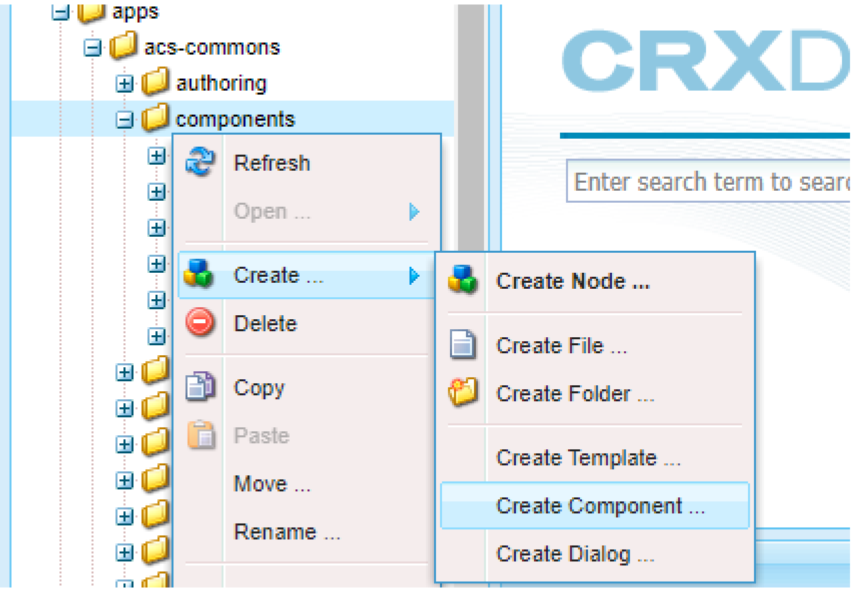
- **Translation Enhancements:** Efficiency has been increased for content translation with AEM through automation and improved third party integration.

21. What is Component?

- Components are modular units to render functionality specific data on your website.
- Components are re-usable and can contains other components
- Have edit behaviour that can be configured.
- Are developed using HTL or JSP.

22. How to create a component and what is the component node type?

"Create Component" option is available in crx to create new component structure and node type is of component is **"cq:Component"**



23. Explain properties of Component

- **jcr:title -** Component Title
- **jcr:description -** Component Description
- **componentGroup -** Component Group (ex: news-content)

- **sling:resourceSuperType -** to mention parent component path
- **cq:isContainer -** to enable/disable container behaviour
- **cq:noDecoration -** it removes/add default decoration element
- **dialogPath -** to reuse other component dialog.

24. What is static template?

- Template is blueprint of the page
- In order to create a page Content authors, need a template.
- Every template connects with one-page component to render common content.

25. How to create a template and what is the template node type?

“Create Template” option is available in crx to create new Template structure and node type is of Template is **“cq:Template"**

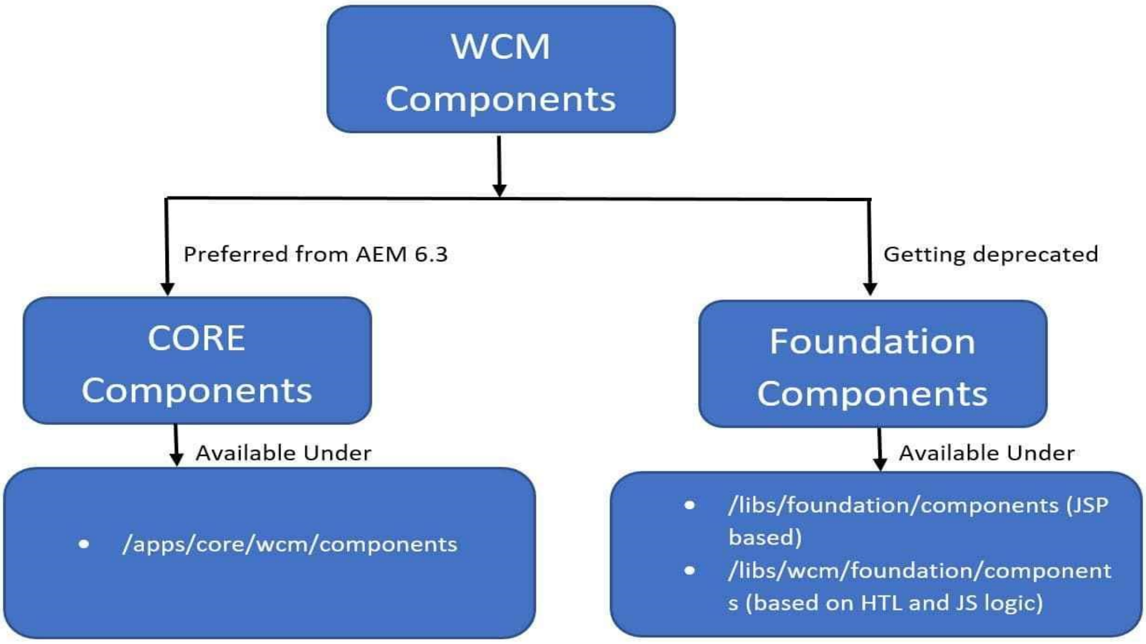
26. Explain properties of the template

- **allowedChildren -** Path of a template that is allowed to be a child of this template.
- **allowedParents -** Path of a template that is allowed to be a parent of this template.
- **allowedPaths -** Path of a page that is allowed to be based on this template.
- **jcr:created** - Date of creation of the template.
- **jcr:description -** Description of the template.
- **jcr:title -** Title of the template.
- **ranking** - Rank of the template. Used to display the template in the User Interface.

27. Explain about core components

- AEM Core WCM Components (commonly referred to as core components) are a set of production-ready components Adobe introduced with Adobe Experience Manager (AEM) 6.3. They are open source and delivered over Github, so the entire AEM developer community is welcome to provide feedback and contribute to the code base (cue applause). These components use the same implementations (e.g. HTML Template Language, or HTL, Touch UI dialogs and Sling Models) that Adobe advocates in AEM 6.3

- The main purpose of developing AEM core components in AEM is to make extensible components in AEM that are secure, robust, version-able and uses latest technologies.


WCM Components
Preferred from AEM 6.3
Getting deprecated
CORE Components
Foundation Components
Available Under
Available Under
• /apps/core/wcm/components
• /libs/foundation/components (JSP based)
• /libs/wcm/foundation/components (based on HTL and JS logic)

AEM CQ5 Tutorials

-

28. Difference between parsys and responsive layout

- Both can be used to hold components when you drag and drop in parsys/responsive grid.
- Responsive behaviour across different view ports is the main difference between a parsys and layout container.
- With parsys, you drag and drop components and the position of these components remains the same in all viewports (devices).Take for example a tile component which is positioned horizontally for desktop viewports and you want it stacked vertically as the viewport decreases, with just the paragraph system, you would require CSS changes or a responsive framework like bootstrap to achieve this.

29. What is overlay?

- When you overlay a component in AEM means that copy component from /libs/ folder to /apps/. folder. And you can impose your own definitions (like change title, group, business logic functionalities) on the newly copied components under /apps/..
- You can try overlaying these libs/foundation/components/ list, image, Text&Image, Carousel, etc. simple components to play around and change the dialogs, JSP level functionalities and see the behaviour. When you are overlaying a component remember that both components can show up in the authors sidekick, For your overlay /apps/. component, if the title, componentGroups are the same as /libs/. component, in design mode of the parsys to enable component can distinguish them with parenthesis around the component (foundation) vs (your project).

30. Difference between overlay and override?

- I basically prefer extending a component in AEM because, if you extend the component and setting the property sling:resourceSuperType to the particular path of the component and you can modify the component according to your project requirements.
- Also - Once you upgrade your AEM to higher version, you can inherit the features of existing component from OOTB and you no need to manually make the changes.
- And Overlays - Here you can copy the component from OOTB to your apps and make changes according to your project requirements. Here when you upgrade to higher version of AEM, you need to manually apply the changes (No inheritance from OOTB).

31. Explain about sling resource merger.

- Sling resource merger in AEM provide us the flexibility to have merged view of multiple other resources. The exact merging mechanism will depend on the resource picker implementation that we are using (i.e. Overlay or Override). It is a part of sling framework and available under Felix console by name (org.apache.sling.resourcemerger). To understand in detail how sling handles resource merging you can visit Sling Resource Merger.

- By using Sling Resource Merger we can:-
  1. remove existing resource/properties from the underlying resources.
  2. modify existing properties/child resources of the underlying resources.
  3. add new properties/child resources.

1. what is OSGI?

- OSGi (Open Service Gateway Initiative) is a Java framework for developing and deploying modular software programs and libraries.
- Every module is a bundle.
- These modules interact with each other via an API. The API is defined as a set of classes and methods which can be used from other components.

2. Advantages of OSGI

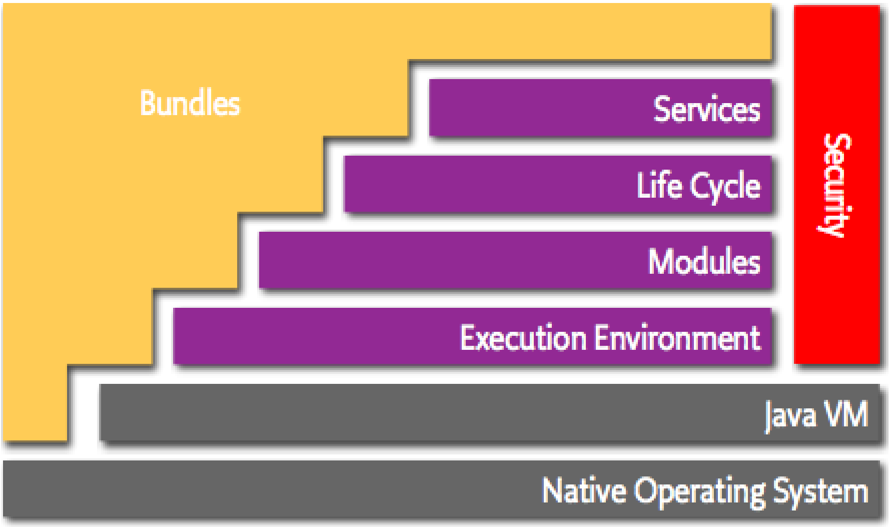
- OSGI is modular based framework, it can be used to develop application by splitting into multiple modules to develop parallelly.
- It reduces the complexity of the system.
- Bundles are loosely coupled and easy to manage
- It supports multiple versions of same bundle.
- down time not required to deploy new bundle

3. Architecture of OSGI

The OSGi has a layered model. The following list contains a short definition of the terms:

- **Bundles –** Bundles are normal jar components with extra manifest headers.
- **Services –** The service layer, which hold the service-side of the framework, keeps the service registry and manages it.
- **Life-Cycle –** The lifecycle layer manages and keeps track of the frameworks and bundles lifecycle state. It is used to install or uninstall framework objects and start or stop them.

- **Modules –** The module layer, which is the bundle space, holds the bundles that are installed on the framework and are managed through the lifecycle layer.
- **Security –** The security layer, which extends the Java 2 security architecture, is optional. When active, it validates the bundle signatures and controls the component access rights.
- **Execution Environment –** The execution environment layer, which is the bottom layer on which the bundles live, is selected to fit the underlying hardware or operating system.



4. What is Bundle?

OSGi Bundles are normal jars with extra manifest headers.

Manifest headers are nothing but some information about jar which includes groupId, artifactId, name, version, who can see its classes or use them (Export packages), on whom it is dependent (import packages) etc.

Metadata:

Manifest-Version: 1.0

Bundle-Description: Article bundle for gofortrainings

Bundle-Name: gofortrainings - Article

Bundle-SymbolicName: com.gofortrainings.article

Bundle-Version: 1.0.0.SNAPSHOT

Export-Package: model;version="1.0.0",service;version="1.0.0";uses:=model

Import-Package:
model;version="[1.0,2)",service;version="[1.0,2)",javax.inject;version="0.0.0"

5. What is difference between Jar & Bundle?

- There is basically no difference. A JAR is a bundle and a bundle is a JAR.
- But Bundle will have additional metadata in its manifest so that an OSGi framework can manage the visibility of classes between bundles.
- Metadata can be used to manage lifecycle of Bundle based on metadata configuration like import-package,export-packages etc.

6. What is difference between Package & Bundle?

Package	Bundle
Content packages can be used to deploy AEM implementation code into AEM Server, it contains AEM component, templates, clientlibs, runmode configuration etc.	Bundle can be used deploy OSGI code into Apache Felix container , It is basically a jar with extra manifest headers , it contains all backend (java) implementations
deploy into package manager	install into felix console
Extension of content packages is .zip	Extension of bundle is .jar

7. Life cycle of Bundle.

Installed – The bundle has been successfully installed.

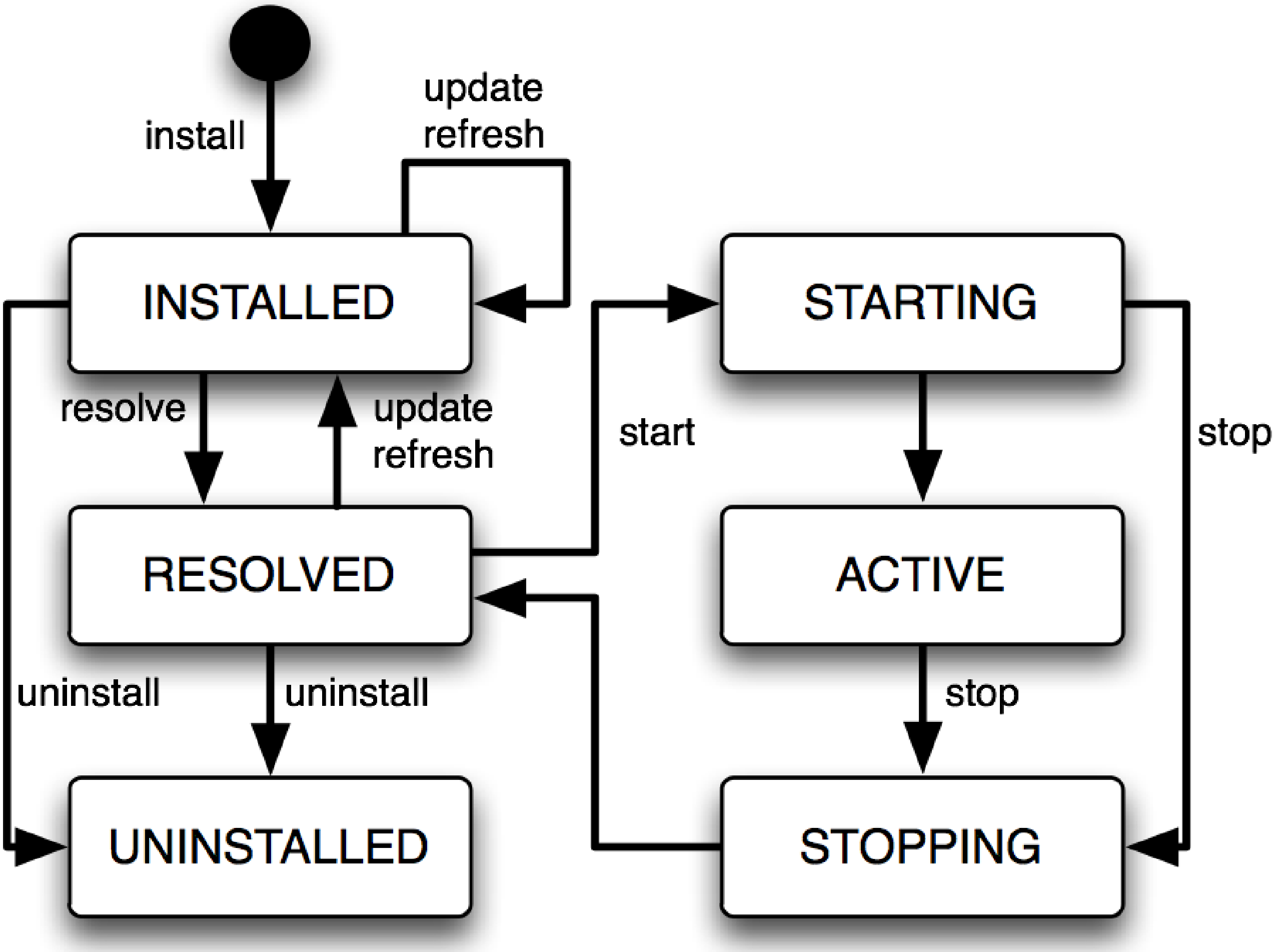
Resolved – All Java classes that the bundle needs are available. This state indicates that the bundle is either ready to be started or has stopped.

Starting – The bundle is being started, the BundleActivator.start method will be called, and this method has not yet returned. When the bundle has an activation policy, the bundle will remain in the STARTING state until the bundle is activated according to its activation policy

Active – The bundle has been successfully activated and is running; its Bundle Activator start method has been called and returned.

Stopping – The bundle is being stopped. The BundleActivator.stop method has been called but the stop method has not yet returned.

Uninstalled – The bundle has been uninstalled. It cannot move into another state.



8. What is OSGI Component?

- A bundle in OSGi has a life cycle in the container.
- If the user wants the object to participate in the life cycle, the object must be declared as a component.
- Life cycle of the component is managed by a component framework such as Declarative Services (DS)
- A component is an active participant in the OSGi system.
- @component is the annotation used to indicate the class as a component.

9. What is OSGI Service?

- A component is having a life cycle. But it can’t access other components itself. It needs to be registered as a service in order to be used in other components.
- A service object registers itself in the OSGi service registry and is looked up by other services using its interface name.
- @service is the annotation used to indicate the component as a service.

10. Difference between OSGI component & Services

- A component is having a life cycle. But it can’t access other components itself. It needs to be registered as a service in order to be used in other components.
- All the services are components. But vice-versa not true as all the components need not to be a service.

11. Explain about OSGi configuration service.

Configuring is the process of defining the configuration data for bundles and assuring that those bundles receive that data when they are active in the OSGi framework.

OSGi provides a way to configure services and modify those configurations on the run-time; changes of configuration values will be updated immediately in OSGI services / components.

New / Updated configuration values can be read from life cycle methods of OSGI services / component activate, deactivate and update.

12. How to read configuration in OSGI service/component.

We can read OSGI configuration from activate method, below is code snippet

SCR Annotations:

```
protected void activate(ComponentContext context) {
        Dictionary<String, Object> properties = context.getProperties();
        marketoSite = PropertiesUtil.toString(properties.get(MARKETO_SITE),
marketoSite);
}
```

OSGI R6/7 Annotations:

```
@Activate
public void activate(MyServiceConfiguration config) {
        this.config = config;
    }
```

13. Can you explain about OSGI Config nodes.

Create a node of type sling:OsgiConfig under the runtime folder you wish to configure. The name of the node should be the same name as the Persistent Identity (PID) of the configuration in the OSGi Console. HINT: If this is a factory class and can have multiple instances, use the Factory Persistent Identifier (Factory PID) and append the name with '-xxxxx', where xxxxx is a unique identifier.

14. Explain about configuration/Service factory?

Factory Configurations is a way to create a single service, and bind multiple configurations to it & then consume those configurations from different classes/Services. You all might have used Logger service provided by AEM out of the box, logger is an example of factory configuration service.

An **OSGi service factory** is a special class ServiceFactory, which can create individual instances of service objects for different bundles. Sometimes a service needs to be differently configured depending on which bundle uses the service

15. Can you explain real time examples about services/configuration factories?

- Logger factory is the best example for a configuration factory.

16. Does OSGI support multiple versions of Bundles?

OSGi supports multiple versions of packages deployed at the same time. You could deploy the 3 versions of 'core' and then request a specific version in your Manifest for the importing applications:

Bundle A

Import-Package: package.name.of.core;version="1.0"

Bundle B

Import-Package: package.name.of.core;version="1.5"

Bundle C

Import-Package: package.name.of.core;version="2.0"

17. Can you explain about bundle metadata tags.

Bundle-SymbolicName - A name that identifies the bundle uniquely.

Bundle-Version -This header describes the version of the bundle, and enables multiple versions of a bundle to be active concurrently in the same framework instance.

Bundle-Activator - This header notifies the bundle of lifecycle changes.

Import-Package - This header declares the external dependencies of the bundle that the OSGi Framework uses to resolve the bundle. Specific versions or version ranges for each package can be declared. In this example manifest file, the org.apache.commons.logging package is required at Version 1.0.4 or later.

Export-Package - This header declares the packages that are visible outside the bundle. If a package is not declared in this header, it is visible only within the bundle.

18. Can you explain any OSGI services which you written in your project

- < Explain any OSGI service from your project >

19. Explain about OSGI R6/7 annotations

@Component(service = MySimpleService.class) **- can be used to register OSGI components and/or services**

@Reference

private SlingSettingsService settings; - **To inject one OSGI component in another component**

@Designate(ocd = MyServiceConfiguration.class) - **To specify configuration class definition**

@Activate

public void activate(MyServiceConfiguration config) { } **- to specify activate method**

@Deactivate

public void deactivate(MyServiceConfiguration config) { } **- to specify deactivate method**

@Modified

public void update(MyServiceConfiguration config) { } - **to specify update method**

@ObjectClassDefinition **– to define configuration class**

@AttributeDefinition **– to define property in configuration class**

New annotations in R7:

@ServiceDescription("My simple service")

@ServiceRanking(1001)

@ServiceVendor("Feike")

20. Difference between SCR and OSGI annotations

These annotations are compile time annotations, which are handled by maven plugins. These plugins create metadata (same format) which are picked up during runtime. That means that the same runtime can handle both SCR and OSGI annotations if they follow the same spec (which they do).

OSGI R6/7 and SCR are pretty much at the same level, but the future development will happen only on OSGI annotations.

21. How to convert Jar into Bundle?

Approach 1:

- Download Jar file
- Create manifest.txt file and update all metadata headers.
- Execute the command "jar cvfm junit-4.4-bundle.jar manifest.txt junit-4.4.jar" , it converts jar into OSGI bundle.

Approach 2:

- Create a new Plugin project by selection File-> New -> Project...-> Plug-in Development -> "Plug-in from Existing JAR Archives"
- Select jars you want to have in this new plugin(bundle). Enter other plugin data(name, version, id etc.).
- Uncheck the flag Unzip the JAR archive into the project. Press then finish.
- Unchecking the checkbox Unzip the JAR archive into the project, prevents extracting class files from the Jar which is usually not necessary.

EDIT : To Export your bundle to install it into an OSGi runtime. Select your bundle and choose File -> Export -> Plug-in Development -> "Deployable plug-ins and fragments".

Uncheck the checkbox to Export source.

22. How to get ResourceResolver/Session object in OSGI services.

If a request/resource object is not available, then we can’t create a ResourceResolver object directly.

System user is another approach to create ResourceResolver object by using ResourceResolverFactory

1. Create system users and add all necessary permissions.
2. Update User Mapper configuration **(bundleid:subservice=systemuser)**

```
Map<String, Object> param = new HashMap<String, Object>();

param.put(ResourceResolverFactory.SUBSERVICE, "subservicename");

ResourceResolver resourceResolver =
resourceResolverFactory.getServiceResourceResolver(param);
```

23. Can you explain any OOTB OSGI Services?

Ex:

- **Externalizer -** To transform a resource path into an external and absolute URL
- **QueryBuilder -** To create and run an XPath server side query
- **MessageGatewayService -** To send email messages that are based on AEM email templates
- **SlingSettingsService –** It provides sling basic setting like runmodes, home path, Id and name

- **ResourceResolverFactory -** Defines the service API to get and create ResourceResolvers
- **Replicator –** It can be used to replicate content
- **AuditLog –** create audit logs

24. How to get runmodes in OSGI services

SlingSettingService OSGI service can be used to get a list of run modes of current AEM instances in your OSGI service.

```
@Reference
private SlingSettingsService slingSettingsService;


Set<String> runModes = slingSettingsService.getRunModes();
```

1. What is sling model?

Sling Models are simple POJO classes which are mapped automatically with Sling Objects (resource, request objects.) and allow us to access jcr node property values directly into java classes.

Features of Sling Models:

- Pure POJO classes.
- Entirely annotation driven (Need to write less code).
- Support both classes and interfaces.
- Using Sling Models, you can do more with less code so you can reduce your coding efforts.
- Your code is more maintainable using Sling Modes.

2. When do we use Resource and SlingHttpRequest adaptable types?

- Adaptable type Resource supports only limited slightly global objects, we can’t inject sling objects like request, response with adaptable type Resource.

- Adaptable type SlingHttpRequest supports all slightly global objects including sling request and response objects.
- If you are just reading node data then adaptable type Resource is enough otherwise we need to use adaptable type SlingHttpRequest to inject/use global objects (ex: request, response, resourceResolver,scriptHelper etc.)

3. Explain about post constructor

- Using PostConstruct annotation we can invoke any method which we want to execute when all injection has been completed.
- This is act as similar to the activate() method which we use for WCMUsePojo.

4. How do you inject OSGI services in sling models?

@OSGiService annotation can be used to inject OSGI services in sling models, @Inject can also be used to inject OSGI services in older versions of sling models.

5. Explain about different injectors

- **script-bindings -** To inject sling binding objects (log,request,response etc, most of the sightly global objects can be injected with script-bindings injector)
- **valuemap -** To read node properties
- **child-resources -** To get child resources by name
- **request-attributes -** To read request parameters and custom data
- **osgi-services -** To inject OSGI services
- **resource-path -** To inject resource based on path
- **self -** To convert node into custom adaptable class
- **sling-object -** To inject sling objects (request, response, resource resolver, current resource, SlingScriptHelper)

6. Explain about different annotations you used in sling models

@Model – To register one class as adaptable sling model class

@ValueMap – to inject property value from node.

@OSGIService - To Inject OSGI services

@ScriptVariable – To Inject all OOTB global objects

@SlingObject – To inject Sling objects

7. Explain about sling exporter

Sling Model Exporter was introduced in Sling Models v1.3.0. This new feature allows new annotations to be added to Sling Models that define how the Model can be exported as JSON.

If there is a sling model and you want to fetch the same properties as a JSON response,so there is no need to create a Sling Servlet. You just need to export your sling model using jackson exporter and that’s all. Sling Model Exporter can be used as a web service or as a rest API.

8. Difference between Sling Models and WCMUsePojo

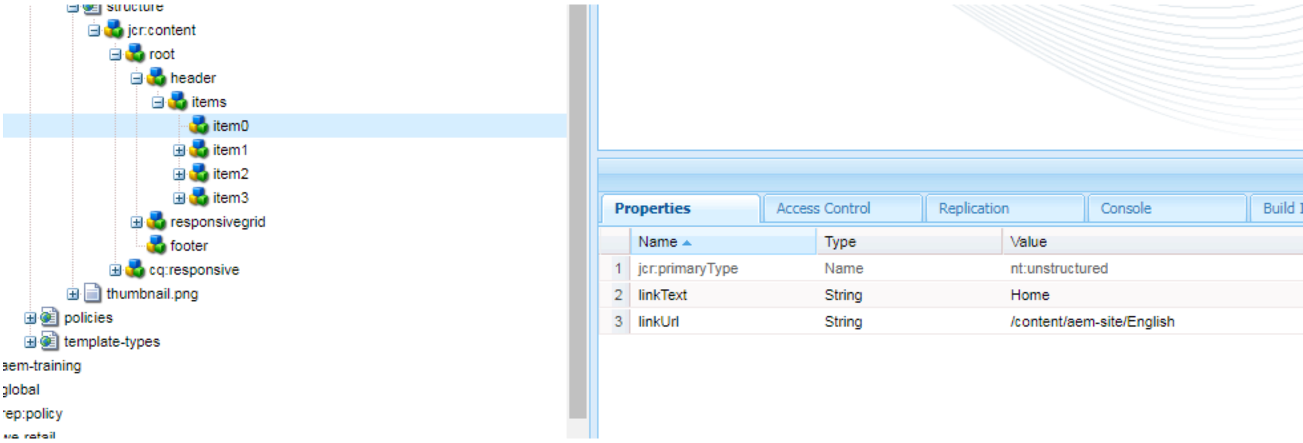
Both can be used to read node data to implement business logic, but sling models are more advanced and easier to write code.
In sling models , Node data automatically injected into java variables but in wcm use pojo we need to read and assign to java variable.

9. How to read multifield data in sling models.

Below is the syntax to read list of nodes.

@Inject
public List<Resource> items;

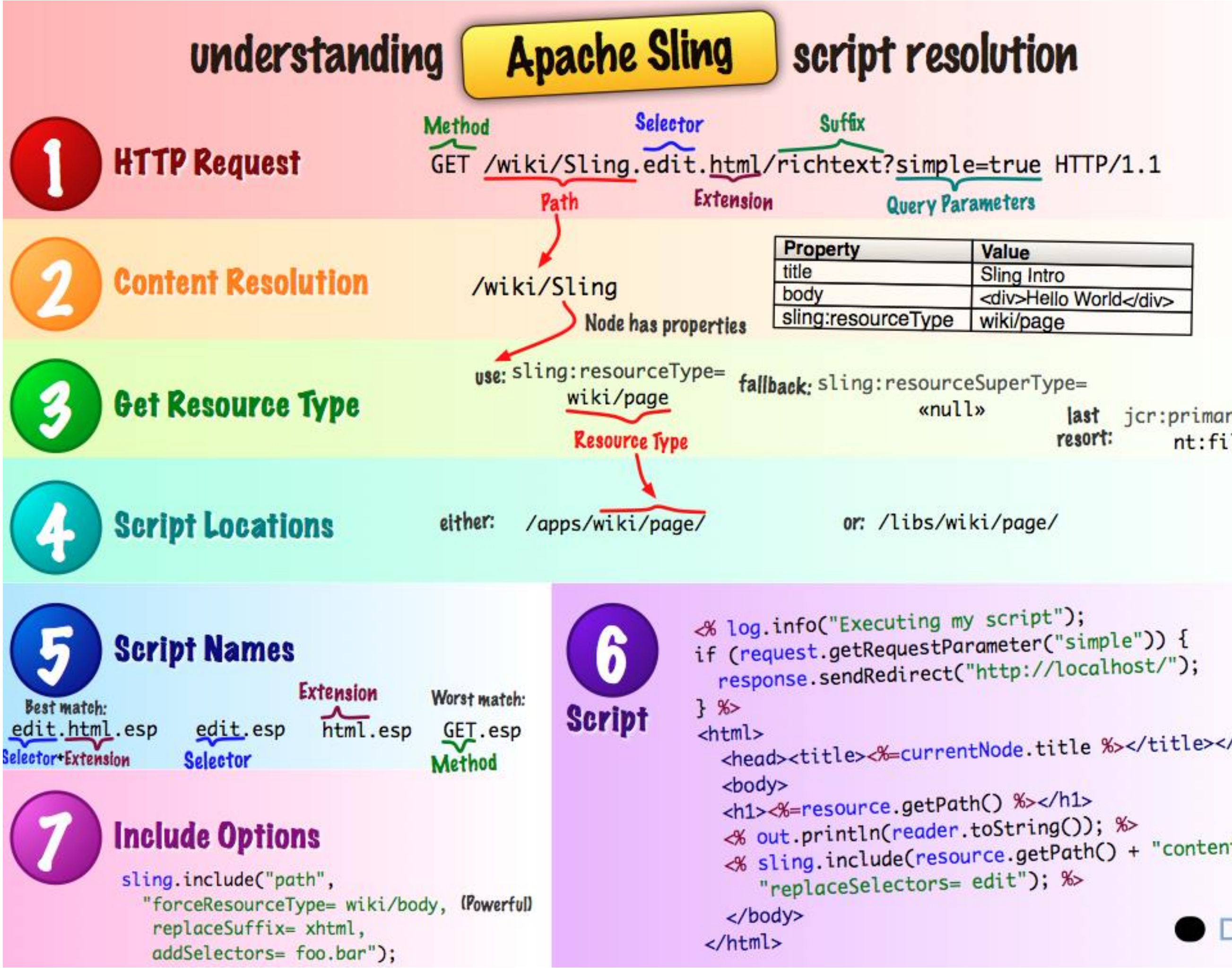
Here is items is node name , above syntax reads all child nodes of items node



	Name ▲	Type	Value
1	jcr:primaryType	Name	nt:unstructured
2	linkText	String	Home
3	linkUrl	String	/content/aem-site/English

01. Sling Request processing?

1. **URL Decomposition** - Sling splits the user request url to find out URI (ex: content page path) information.
2. **Content Resolution** - Sling checks whether URI (ex: page content path) is available in repository or not, if not available then it returns resource not found (404) error
3. **Get Resource Type** - If resource is available in repository then it extracts sling:resourceType property value from child node jcr:content
4. **Script Location** - If value of sling:resourceType is relative path then it will check first in /apps/ folder followed by /libs/ folder
5. **Script Names** - Once component located based sling:resourceType value , by default if selectors are not available then it will execute componentname.(html/jsp) script file ,otherwise it will search for selector specific script file
6. Sling executes the script files and return response in requested format (html,json,xml)



Property	Value
title	Sling Intro
body	<div>Hello World</div>
sling:resourceType	wiki/page

```
<% log.info("Executing my script");
if (request.getRequestParameter("simple")) {
  response.sendRedirect("http://localhost/");
} %>
<html>
  <head><title><%=currentNode.title %></title></
  <body>
  <h1><%=resource.getPath() %></h1>
  <% out.println(reader.toString()); %>
  <% sling.include(resource.getPath() + "conten
     "replaceSelectors= edit"); %>
  </body>
 </html>
```

```
sling.include("path",
  "forceResourceType= wiki/body,
   replaceSuffix= xhtml,
   addSelectors= foo.bar");
```

02. How to register Sling Servlet?

In AEM, servlet also an OSGI service of type javax.servlet.Servlet , to register OSGI service as servlet we need to configure few mandatory service properties either ‘sling.servlet.paths’ or ‘sling.servlet.resourceType’ ,If neither is set, the Servlet service is ignored.

03. How many ways we can register sling servlet?

You can register a Servlet using the two Standard approaches:

1. Registering the servlet by path (sling.servlet.paths)

2. Register servlet by ResourceType (sling.servlet.resourceTypes)

04. Difference between Path based servlet and resource-based servlet?

Binding servlets by paths has several disadvantages when compared to binding by resource types, namely:

- path-bound servlets cannot be access controlled using the default JCR repository ACLs
- path-bound servlets can only be registered to a path and not a resource type (i.e. no suffix handling)
- if a path-bound servlet is not active, e.g. if the bundle is missing or not started, a POST might result in unexpected results. usually creating a node at /bin/xyz which subsequently overlays the servlets path binding
- the mapping is not transparent to a developer looking just at the repository

Given these drawbacks it is strongly recommended to bind servlets to resource types rather than paths.

05. Explain about different OOTB servlet classes (SlingAllMethodServlet, SlingSafeMethodServlet)

SlingSafeMethodServlet: A base class for read-only Servlets used in Sling. This base class is just a better implementation of the Servlet API HTTP Servlet class which accounts for extensibility. So, extensions of this class have great control over what methods to overwrite. It supports GET, HEAD, OPTIONS etc methods.

SlingAllMethodServlet: A base class for data modifying Servlets used in Sling. This class extends the SlingAllMethodServlet by support for the POST, PUT and DELETE methods.

06. Can you explain different classes which you use in sling API?

- **ValueMap:** The ValueMap is an easy way to access properties of a resource.
- **ResourceResolver:** The ResourceResolver defines the service API which may be used to resolve Resource objects.
- **ResourceProvider:** API for providers of resources.
- **Resource:** Resources are pieces of content on which Sling acts The Resource is also an Adaptable to get adapters to other types.
- **ResourceWrapper:** The ResourceWrapper is a wrapper for any Resource delegating all method calls to the wrapped resource by default.
- **ResourceUtil:** The ResourceUtil class provides helper methods dealing with resources.

07. How many ways we can register Sling Scheduler?

The scheduler can be register in two ways,

- by registering the job through the scheduler API
- by leveraging the whiteboard pattern that is supported by the scheduler.

Scheduler using Schedular API :

```
@Reference

private Scheduler scheduler;
```

Defining the job : The following code sample defines a job object that writes a message in the logs:

```
final Runnable job = new Runnable() {

        public void run() {

                log.info("Executing the job");

        }

};
```

Scheduling with a cron expression:

To execute the job as defined above at 10:15am every Monday, Tuesday, Wednesday, Thursday and Friday, you can use the addJob() method with the following parameters:

```
String schedulingExpression = "0 15 10 ? * MON-FRI";

this.scheduler.addJob("myJob", job, null, schedulingExpression, true);
```

Scheduling at periodic times:

To execute the job as defined above every 3 minutes (180 seconds), you can use the addPeriodicJob() method with the following parameters:

```
long period = 3*60; //the period is expressed in seconds

this.scheduler.addPeriodicJob("myJob", job, null, period, true);
```

Scheduling at a given time:

To execute the job as defined above at a specific date (on January 10th 2020), you can use the fireJobAt() method with the following parameters:

```
SimpleDateFormat formatter = new SimpleDateFormat("yyyy/MM/dd");

String date = "2020/01/10";

java.util.Date fireDate = formatter.parse(date);
```

```
this.scheduler.fireJobAt("myJob", job, null, fireDate);
```

Scheduler using white board pattern:

Scheduling with a cron expression:

The following job is executed every minute by setting scheduler.expression to the cron expression "0 * * ?":

```
@Component

@Service(value = Runnable.class)

@Property( name = "scheduler.expression", value = "0 * * * * ?")

public class ScheduledCronJob implements Runnable {

    /** Default log. */

    protected final Logger log = LoggerFactory.getLogger(this.getClass());

    public void run() {

        log.info("Executing a cron job (job#1) through the whiteboard
pattern");

  }

}
```

Scheduling at periodic times:

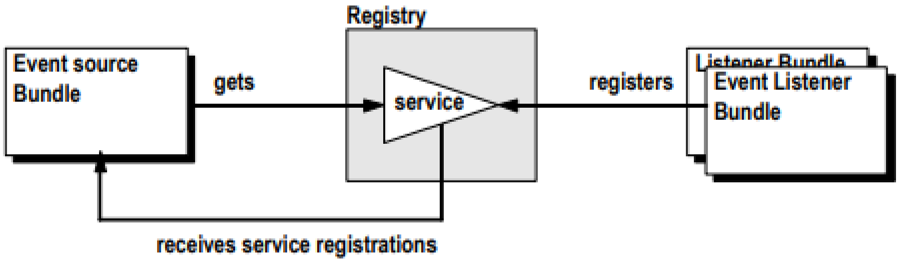
The following job is executed every ten seconds by setting scheduler.period to 10:

```
@Component

@Service(value = Runnable.class)

@Property( name = "scheduler.period", longValue = 10)

public class ScheduledPeriodicJob implements Runnable {

    /** Default log. */

    protected final Logger log = LoggerFactory.getLogger(this.getClass());

    public void run() {

        log.info("Executing a perodic job (job#2) through the whiteboard
pattern");

  }++
//
}
```

Reference : https://sling.apache.org/documentation/bundles/scheduler-service-commons-scheduler.html

08. What is white board pattern?

Whiteboard Design Pattern is an OSGi service model, which influences the OSGi framework's service registry. Instead of having event listeners track event sources and then register themselves with the event source, the whiteboard pattern has event listeners register themselves as a service with the OSGi framework. Now, an event listener can find the service in the service registry or react to it when the service is executed. In this pattern, control is given to an event listener. When the event source has an event object to deliver, the event source calls all event listeners in the service registry and the event source is not registered with the framework as a service.



09. Explain about cron expression?

- A cron expression is a string consisting of six or seven subexpressions (fields) that describe individual details of the schedule.
- A CRON expression takes the following format (years are optional):

  <seconds> <minutes> <hours> <days of month> <months> <days of week> <years>
- Cron expressions can be as simple as * * * * ? * or as complex as 0 0/5 14,18,3-39,52 ? JAN,MAR,SEP MON-FRI 2002-2010.

Example:

Expression	Means

0 0 12 * * ?	Fire at 12:00 PM (noon) every day
0 15 10 ? * *	Fire at 10:15 AM every day
0 15 10 * * ?	Fire at 10:15 AM every day
0 15 10 * * ? *	Fire at 10:15 AM every day

10. How to register Sling Event Handler?

- We can achieve event handling at sling level by implementing EventHandler interface in a class.
- By writing a service class that implements EventHandler interface which must be registered with a service property EventConstants.EVENT_TOPIC (e.g. ReplicationAction.EVENT_TOPIC , PageEvent.EVENT_TOPIC) whose value is the list of topics
- implementing handleEvent(Event) method to trigger the job.

```
@Service(value = EventHandler.class)

@Component(immediate = true)

@Property(name = "event.topics", value =
ReplicationAction.EVENT_TOPIC)

public class ReplicationLogger implements EventHandler, JobProcessor {

  private static final Logger LOGGER =
LoggerFactory.getLogger(ReplicationLogger.class);

  @Reference

  private JcrResourceResolverFactory jcrResourceResolverFactory;

  @Override

  public void handleEvent(Event event) {

      LOGGER.info("********handling event");

      process (event);

  }
```

11. Difference between Event Handler vs Event Listener?

Event handler	Event listener
It is at the Sling level with event handlers and jobs	It is at the JCR level with observation.
org.osgi.service.event.EventHandler interfa ce	javax.jcr.observation.EventListener interfa ce
override handleEvent(Event) method.	override onEvent(EventIterator) method

12. What is observation manager?

Observation is a JCR mechanism that allows code to listen for events generated during repository operations and react to them in various ways. Two types of observation may be supported by a JCR repository: Asynchronous Observation and Journaled Observation.

The seven type of events that can be detected are as follows:

- Node added
- Node moved
- Node removed
- Property added
- Property removed
- Property changed
- Persist

The ObservationManager object acquired via Workspace.getObservationManager(). Allows for the registration and deregistration of event listeners.Observation Manager Api is omni present in AEM. 200 thousand event it can handle in entire life time. It has methods like :-

- 1. addEventListener(EventListener listener, int eventTypes, java.lang.String absPath, boolean isDeep, java.lang.String[] uuid, java.lang.String[] nodeTypeName, boolean noLocal) - Adds an event listener that listens for the specified eventTypes (a combination of one or more event types encoded as a bit mask value).
    - listener - an EventListener object.
    - eventTypes - A combination of one or more event type constants encoded as a bitmask.
    - absPath - an absolute path.
    - uuid - Only events whose associated node has one of the UUIDs in this list will be received. If his parameter is null then no UUID-related restriction is placed on events received.
    - nodeTypeName - Only events whose associated node has one of the node types (or a subtype of one of the node types) in this list will be received. If his parameter is null then no node type-related restriction is placed on events received.
    - noLocal - if noLocal is true, then events generated by the session through which the listener was registered are ignored. Otherwise, they are not ignored.
- 2. removeEventListener(EventListener listener) - Deregisters an event listener.
- 3. getRegisteredEventListeners() - Returns all event listeners that have been registered through this session.

  Reference : http://blogs.adobe.com/experiencedelivers/experience-management/observation_in_jcr/

1. what is OSGI?

- OSGi (Open Service Gateway Initiative) is a Java framework for developing and deploying modular software programs and libraries.
- Every module is a bundle.
- These modules interact with each other via an API. The API is defined as a set of classes and methods which can be used from other components.

2. Advantages of OSGI

- OSGI is modular based framework, it can be used to develop application by splitting into multiple modules to develop parallelly.
- It reduces the complexity of the system.
- Bundles are loosely coupled and easy to manage
- It supports multiple versions of same bundle.
- down time not required to deploy new bundle

3. Architecture of OSGI

The OSGi has a layered model. The following list contains a short definition of the terms:

- **Bundles –** Bundles are normal jar components with extra manifest headers.
- **Services –** The service layer, which hold the service-side of the framework, keeps the service registry and manages it.
- **Life-Cycle –** The lifecycle layer manages and keeps track of the frameworks and bundles lifecycle state. It is used to install or uninstall framework objects and start or stop them.
- **Modules –** The module layer, which is the bundle space, holds the bundles that are installed on the framework and are managed through the lifecycle layer.
- **Security –** The security layer, which extends the Java 2 security architecture, is optional. When active, it validates the bundle signatures and controls the component access rights.
- **Execution Environment –** The execution environment layer, which is the bottom layer on which the bundles live, is selected to fit the underlying hardware or operating system.

4. What is Bundle?

5. What is difference between Jar & Bundle?

- There is basically no difference. A JAR is a bundle and a bundle is a JAR.
- But Bundle will have additional metadata in its manifest so that an OSGi framework can manage the visibility of classes between bundles
- Metadata can be used to manage lifecycle of Bundle based on metadata configuration like import-package,export-packages etc.

6. What is difference between Package & Bundle?

Package	Bundle
Content packages can be used to deploy AEM implementation code into AEM Server, it contains AEM component, templates, clientlibs, runmode configuration etc.	Bundle can be used deploy OSGI code into Apache Felix container , It is basically a jar with extra manifest headers , it contains all backend (java) implementations
deploy into package manager	install into felix console
Extension of content packages is .zip	Extension of bundle is .jar

7. Life cycle of Bundle.

Installed – The bundle has been successfully installed.

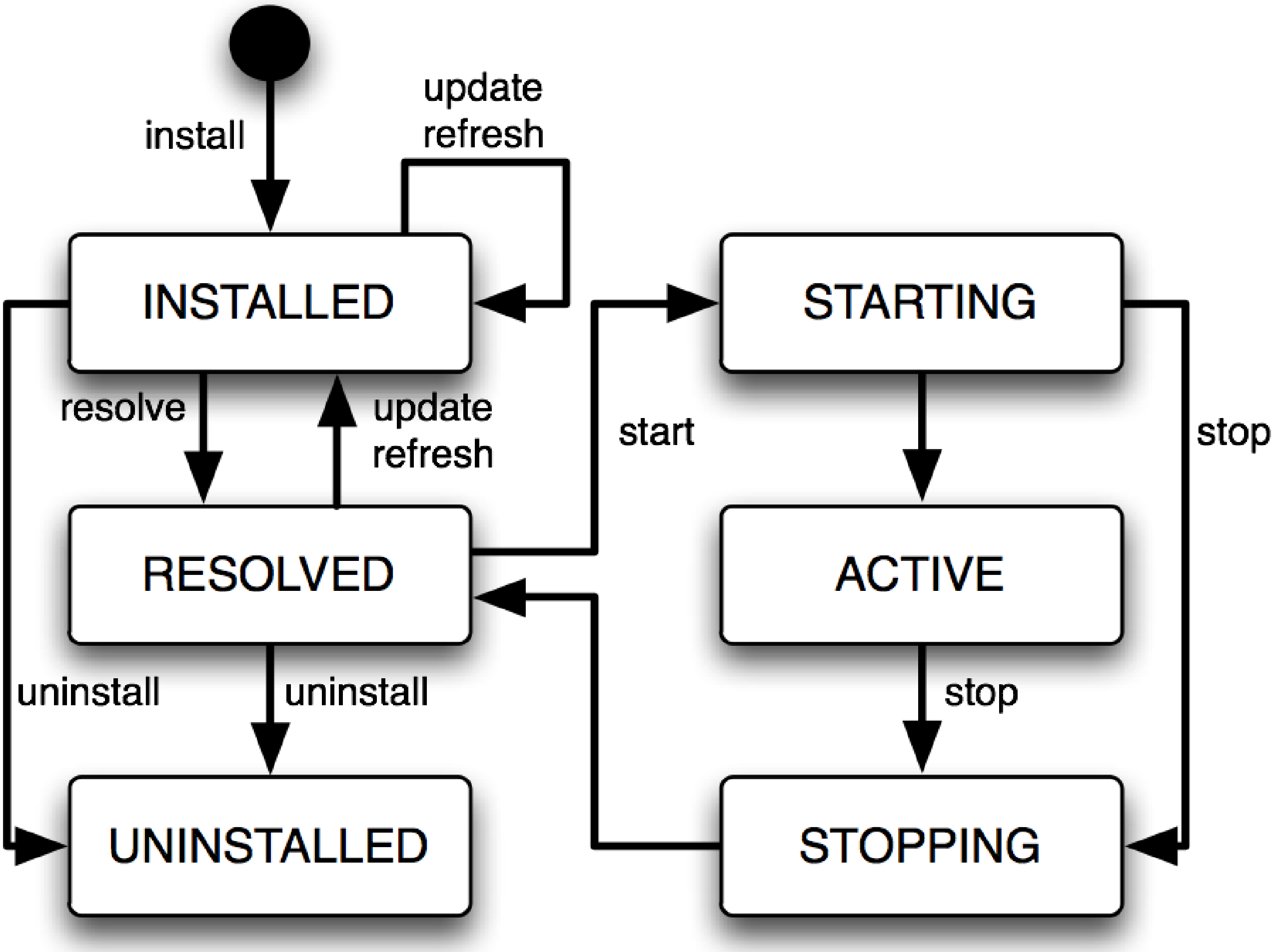
Resolved – All Java classes that the bundle needs are available. This state indicates that the bundle is either ready to be started or has stopped.

Starting – The bundle is being started, the BundleActivator.start method will be called, and this method has not yet returned. When the bundle has an activation policy, the bundle will remain in the STARTING state until the bundle is activated according to its activation policy

Active – The bundle has been successfully activated and is running; its Bundle Activator start method has been called and returned.

Stopping – The bundle is being stopped. The BundleActivator.stop method has been called but the stop method has not yet returned.

Uninstalled – The bundle has been uninstalled. It cannot move into another state.



8. **What is OSGI Component?**

9. **What is OSGI Service?**

10. **Difference between OSGI component & Services**

11. **Explain about OSGi configuration service**

12. **How to read configuration in OSGI service/component.**

13. **Can you explain about OSGI Config nodes.**

14. **What is configuration factory / Service factory?**

15. **Can you explain one real time example about services/configuration factory.**

16. Does OSGI support multiple versions of Bundles?

17. Can you explain about bundle metadata tags.

18. Can you explain any OSGI services which you written in your project

19. Explain about OSGI R6/7 annotations

20. Difference between SCR and OSGI annotations

21. How to convert Jar into Bundle?

22. How to get ResourceResolver/Session object in OSGI services.

23. Can you explain any OOTB OSGI Services?

24. How to get runmodes in OSGI services

01. What is clientlib in AEM?

- Node type of clientlib is cq:ClientLibraryFolder
- It allows us to manage our client side resources like (JavaScript, CSS, images, fonts etc), but also provide options to debug, minify, merge and gzip the client-side code.
- It merges all CSS/JS files of clientlib and generates one file with clientlib name (ex: desktop.js and desktop.js, Here desktop is clientlib name).

02. Explain dependencies property in clientlibs

- when your client library files are dependent on some other files, you can use dependencies.
- For example if Clientlib A depends on Clientlib B which depends on Clientlib C, then all clientlibs A,B and C will be included in the page.
- The order in which they will be included is First C the B then A.

03. Difference between embed and dependency property, how it loads on page load?

- AEM will merge all embedded clientlibs into the current clientlib. This is usually used for minimizing requests and for accessing clientlibs, only current clientlib includes on page. For example, if Clientlib A embed Clientlib B which embed Clientlib C, then only Clientlib A will be loaded by embedding Clientlib B code.
- If clientlib A depends on clientlib B then both clientlibs will be included in final html. The order in which they will be included is First B then A.

04. What is the use of channel property in clientlibs?

The channels property is useful when libraries of the same category are designed for different device capabilities.

To associate a client library folder with a device group, add a property to your cq:ClientLibraryFolder node with the following attributes:

Name: channels , Type : String[] , value : touch (Here ‘touch’ is channel name , this clientlib loads only on touch devices)

05. Explain about allow Proxy property

For the client libraries under /apps to be accessible, a proxy servlet is used. The ACLs are still enforced on the client library folder, but the servlet allows for the content to be read via /etc.clientlibs/ if the allowProxy property is set to true.

06. Explain about category property

This is the identifier into which categories a clientlib belongs. A clientlib can have one or more categories, value of this field is a String array.

07. Advantages of clientlibs

- Clientlib merges all JS/CSS files and generates single file, So it decreases no of CSS/JS requests.
- To organize your libraries more efficient/structured way
- Dependency of loading client libraries are managed efficiently.

- HTML client library manager service in AEM provides features like minify your JS files which provides better performance and security.

08. Explain about Compressor

- Minified JS/CSS means less bytes being downloaded from the client perspective, It can be used to improve your site performance.
- HTML Library Manager service configuration provides option to enable minification.
- By default, AEM uses the YUI Compressor.

09. How to debug clientlibs in AEM

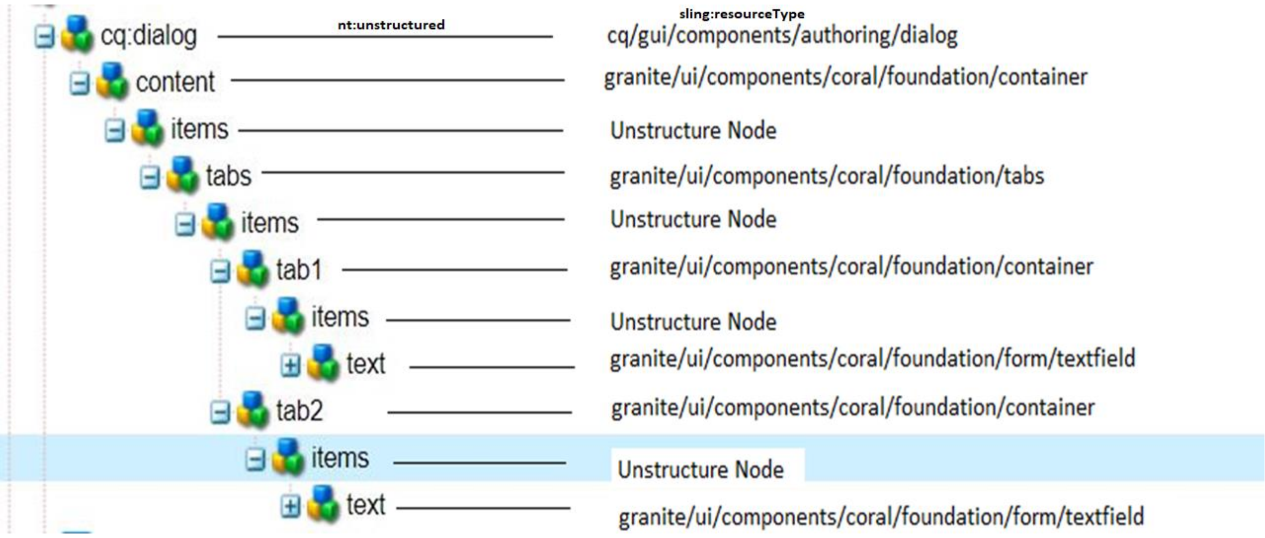
- ?debugClientLibs=true: Just add a query parameter in the page URL,If there is a need to debug how many clientlibs are getting embedded in the main clientlibs.
- Dublibs is adobe tool to test, rebuild and validate clientlibs, it would be useful to clear existing generated clientlibs from repository
- http://localhost:4502/libs/granite/ui/content/dumplibs.html

10. Explain about extra clientlibs in AEM

```
cq: dialog: {
                jcr: primaryType: "nt:unstructured",
                jcr: title: "Page",
                extraClientlibs: ["category1","category2"],
                mode: "edit",
              sling: resourceType: "cq/gui/components/authoring/dialog"
}
```

To have your client library loaded solely for your component dialog (i.e. it will not be loaded for another component) you need to set the property **extraClientLibs** of your dialog to the category name of the client library you have just created. This is advisable if your client library is quite big and/or your field is specific to that dialog and will not be needed in other dialogs.

11. Explain about dialog structure



cq:dialog	cq/gui/components/authoring/dialog
content	granite/ui/components/coral/foundation/container
items	Unstructure Node
tabs	granite/ui/components/coral/foundation/tabs
items	Unstructure Node
tab1	granite/ui/components/coral/foundation/container
items	Unstructure Node
text	granite/ui/components/coral/foundation/form/textfield
tab2	granite/ui/components/coral/foundation/container
items	Unstructure Node
text	granite/ui/components/coral/foundation/form/textfield

12. Difference between dialog and design dialog

Both dialog and design dialog are used by the user to configure the component.

Design Dialog: of a component can be seen/edited in design mode of the page. Design dialog is present at template level so all the page of the template will share same design dialog data.

Dialog: of a component is present at page level so each component instance will have its own dialog and information entered in dialog will be stored under /content folder.

13. What is the default storage location design dialog?

From 6.4 onwards default design dialog location has been changed from /etc/designs/default/jcr:content to /libs/settings/wcm/designs/default/jcr:content

14. How to change default storage location

cq:designPath property need to set root page of website to change default design dialog data path

15. Explain about Coral UI ?

It is a set of CSS and JS files designed and built for Adobe Cloud products.

Coral UI provides a wide range of html components predeveloped for your website like buttons, navigation bar, tool bar, tables, grid, dialogs etc. and save you time from developing these components.

16. Explain about Granite UI?

Granite UI: On top of Coral UI, provides to Touch UI foundational building blocks (Coral UI widgets wrapped into sling components) to build the dialogs and user interfaces consoles.

- Implements component libraries intended for building content-centric web applications
- Provides granular UI widgets
- Provides a default, standardized UI
- Is extensible

16. How to implement multi field dialog

OOTB granite multifield component (granite/ui/components/foundation/form/multifield) that allows developers to create a multifield of a fieldset (group of different fields).

As out of the box multifield component has many limitations, ACS commons extended multifield to store data in JSON format.

17. How to read dialog and design dialog data in HTL

HTL provides global objects to read dialog and design dialog data.

properties – to read dialog data

currentStyle – to read design dialog data

1. What is MSM?

Multi-Site Manager (MSM) allows AEM developers to create copy of existing site and automatically update the copy when changes are done to the source site. By using MSM we can create content in one site and reuse same content in different sites.

Advantages:

- Maintain common look and feel across all websites.
- Easily manage multiple websites that share common content.
- Reduces the time to manage your websites and increases the re-use of common content.
- MSM allows AEM developers to define relations between the sites so that content changes in one site are automatically replicated in all other sites.

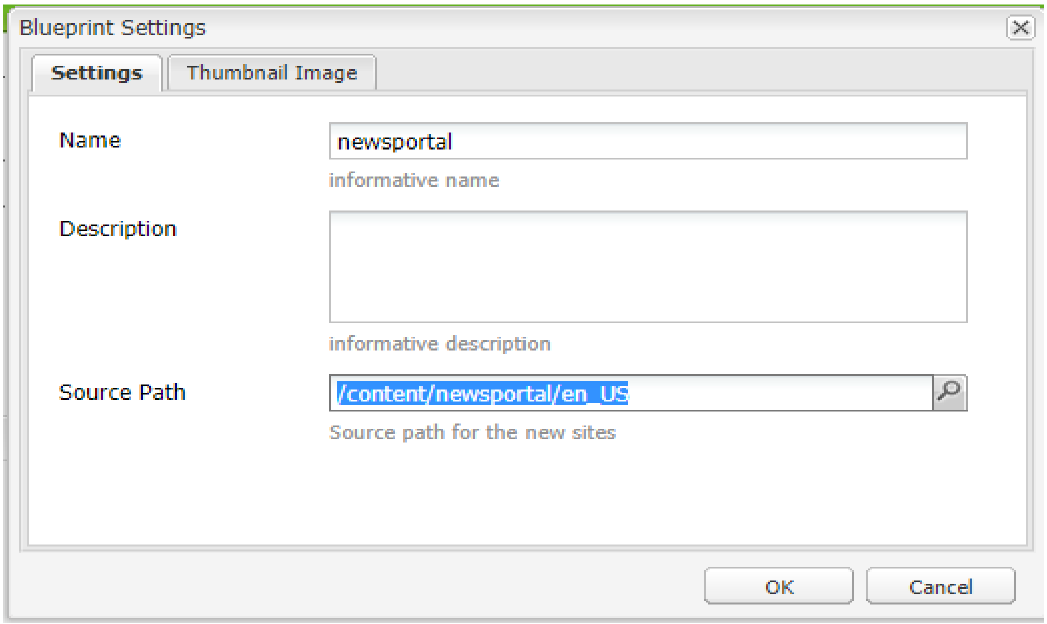
2. When we should use MSM

- Sites are in multiple countries.
- Sites are in multiple languages.
- Need to reuse/replicate same content in multiple websites.
- Maintain consistent branding and content across websites.

3. What is blueprint and how to create blueprint?

A blueprint configuration identifies an existing website that you want to use as the source for one or more live copy pages. Blueprint configurations enable you to push content changes to live copies.

- Create a new blueprint configuration page under http://localhost:4502/miscadmin#/etc/blueprints
- Open new blueprint configuration page and configure source site (parent site) path and name
- Below screenshot newsportal en_US site is blueprint (source or parent) site



Name: newsportal
informative name

Description:
informative description

Source Path: /content/newsportal/en_US
Source path for the new sites

4. What is live copy?

Live Copy: Is basically creating a new site based on the content and structure of an existing site or blueprint so that when changes are made in base/source site, it is automatically updated here.

How to create?

- Go to sites console http://localhost:4502/sites.html
- Select root page of your site and click on create and select live copy option.
- Select source (blueprint) and destination (new live copy) site
- And then create live copy.

5. Difference between language copy and live copy?

- **Language Copy:** It can be used to create new site (child) from existing website (parent), once new site is created it won't maintain any relationship with parent site. New changes in parent site will not reflect in child (new) site.
- **Live Copy:** A live copy is a copy of blueprint (parent) site content and live copy maintains live relationship with parent (blueprint) site. The live copy inherits content from its source (blueprint | parent); Prime use case is to reuse content at multiple locations. One very good example of the usage of Live Copy is for sharing the content on multiple sub-sites – example Car Dealership Sites.

6. What is rollout and how it works?

Rollout is one option in parent page properties to move content from blueprint (parent) to live copy (child). When you click on rollout, internally it will check for rollout configuration of page to move content to live copy. Standard rollout configuration is the default one if you don't configurate rollout configuration.

7. Explain about rollout configurations

The RolloutConfig holds a list of Synchronization Actions, to be used when triggered. The LiveCopy inherits the RolloutConfig and the result is present in the LiveRelationship.

Setting up a live copy for the very first time also uses a RolloutConfig (which triggers the Synchronization Actions).

8. How to create custom rollout configurations?

Create a rollout configuration when the installed rollout configurations do not meet your application requirements:

- Open the Tools console in the classic UI; for example, http://localhost:4502/miscadmin#/etc
- In the folder tree, select the Tools, MSM, Rollout Configurations folder.
- Click New, then New Page to define the Rollout Configuration properties: (Enter name,title etc.)
- Click Create and edit configuration page
- In the Rollout Config dialog, select the Sync Trigger to define the action that causes the rollout to occur.

9. Difference between sync and rollout?

Both can be used to pull data from parent (blueprint) to child (livecopy) pages but rollout option would be available in blueprint pages and sync option would be available in live copy pages.

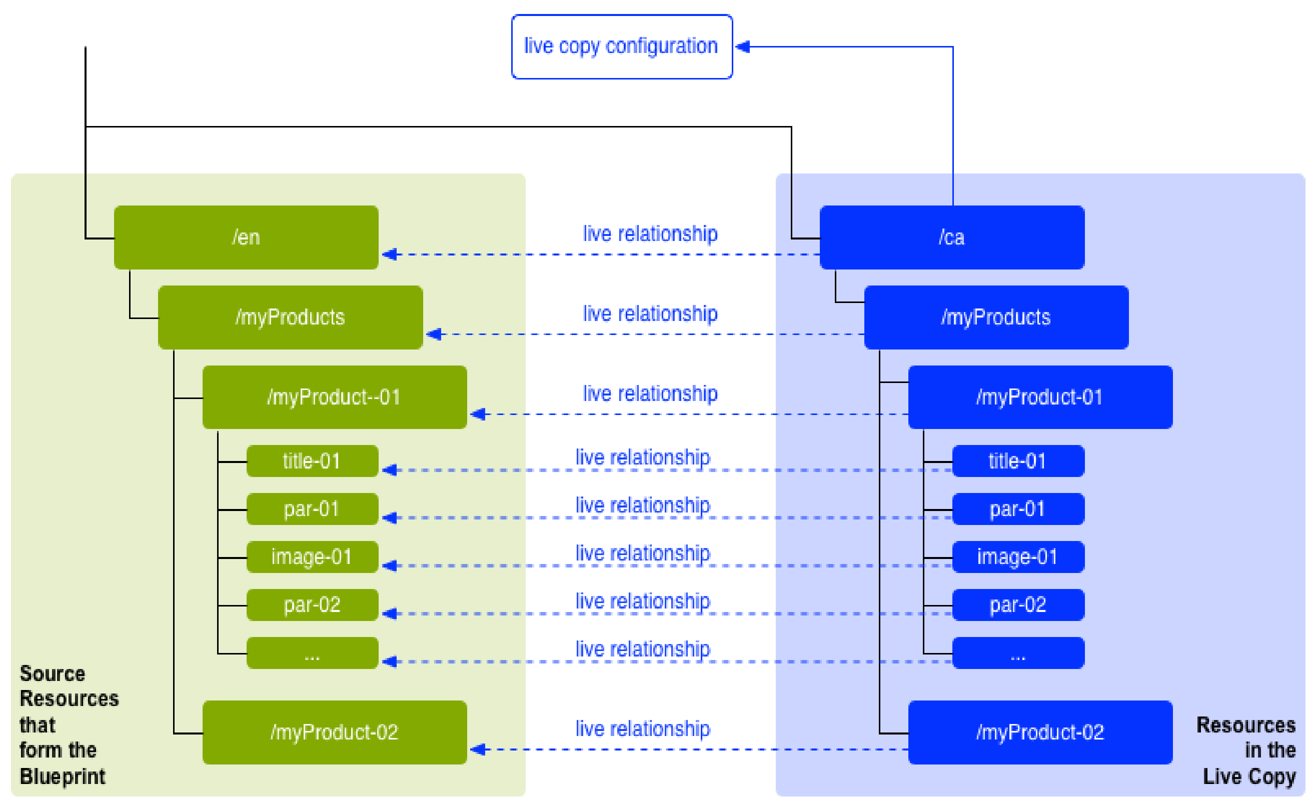
10. Explain about different synchronization actions?

- **contentCopy -** When nodes of the source do not exist on the live copy, copies the nodes to the live copy
- **contentDelete -** Deletes nodes of the live copy that do not exist on the source.
- **contentUpdate -** Updates the live copy content with the changes from the source.
- **editProperties -** Edits properties of the live copy. The editMap property determines which properties are edited and their value.
- **orderChildren -** On the live copy, it orders the children (nodes), based on the order on the blueprint
- **referencesUpdate -** On the live copy, this synchronization action updates references such as like links.
- **PageMoveAction -** The PageMoveAction applies when a page has been moved in the blueprint.

11. What is Inheritance and how to break it?

A live copy page/component inherits content from its source page/component when synchronization occurs.

We can break inheritance at page level and component level by clicking lock symbol.



12. Difference between MSM and I18N

i18n is used for translation of selected fields/properties based on sling:key and sling:message. whereas MSM allows you to view and manage your blueprint and its live copies, during rollout you can use translation services to translate the page based on its locale automatically. IN MSM your entire page is translated using third party translator services whereas we use i18n when we have selected field which we need to translate like First Name etc.

01. What is HTL (Sightly)?

It's html templating language (HTL) supported by AEM (from 6.0 to replace jsp's) to render presentation layer data. It uses data attributes for server side processing   ex : data-sly-use, data-sly-call, data-sly-test etc..

02. Explain about HTL data attributes

HTL uses below data attributes for server-side processing.

- data-sly-include
- data-sly-resource
- data-sly-use
- data-sly-test
- data-sly-list
- data-sly-repeat
- data-sly-template
- data-sly-call
- data-sly-unwrap
- data-sly-text
- data-sly-attribute
- data-sly-element

03. Advantages of HTL

- **Lightweight –** No dependencies, fast and lean.
- **Secure –** Automatic contextual XSS protection and URL externalization.
- **Code-less –** It provides clear separation between font end and back end logic.
- **Powerful –** Straight-forward API for logic, allowing to do virtually anything.

04. Difference between list and repeat

list just iterates the items written inside host element But repeat iterates the complete structure including   its host element.

05. Explain about template and call tags

data-sly-call: It can be used to call templates with name.

data-sly-template: It is html block which we can define like a function and it can be called later using data-sly-call

06. Explain about HTL context

The display context of a HTL expression refers to its location within the structure of the HTML page

HTL will automatically detect the context of expressions and escape them appropriately, to prevent XSS security problems and escapes that string appropriately.

07. Difference between JSP vs HTL

HTL Offers below advantages over JSP for better development in AEM

- Protection against cross-side scripting injection.
- Easily development of AEM Projects by front-end developers.
- Flexible and powerful templating and logic binding features
- Need to write less code in HTL thus productivity increases.

08. how to pass data to backend models

< info.html >

```
<div data-sly-use.info="${'Info' @ text='Some text'}">

<p>${info.reversed}</p>

</div>
```

< Info.java >

```
public class Info extends WCMUse {

 private String reversed;

   @Override

   public void activate() throws Exception {

      String text = get("text", String.class);

      reversed = new StringBuilder(text).reverse().toString();

   }

  public String getReversed() {

    return reversed;

   }

}
```

<InfoModel.java>

```
@Model(adaptables = SlingHttpServletRequest.class)

public class TestModel {

  @RequestAttribute(name = " text ")

   String param;



   public String getParam() {

     return param;

  }

}
```

09. HTL Global object

HTL provides access to all objects that were commonly available in JSP after including global.jsp

Ex:

properties - List of properties of the current Resource.

pageProperties - List of page properties of the current Page.

currentPage - com.day.cq.wcm.api.Page

pageManager - com.day.cq.wcm.api.PageManager

request - org.apache.sling.api.SlingHttpServletRequest

response - org.apache.sling.api.SlingHttpServletResponse

resolver - org.apache.sling.api.resource.ResourceResolver

resource - org.apache.sling.api.resource.Resource

currentStyle - com.day.cq.wcm.api.designer.Style

10. How to call clientlibs in HTL ?

```
<sly data-sly-use.clientlib="/libs/granite/sightly/templates/clientlib.html"
                                        data-sly-call="${clientlib.all @
categories='clientlib1,clientlib2'}"/>
```

data-sly-use.clientlib="/libs/granite/sightly/templates/clientlib.html - this line references the clientlib.html file that has templates marked with data-sly-template attributes. These templates are reusable piece of markup. Look at them as functions in HTL.

data-sly-call - used to call one of the templates from the above clientlib.html

clientL ib.all - "all" is the name of the template being called from clientlib.html which is referred using clientLib keyword (-use.clientLib).

11. How to call OSGI services in WCMUSEPojo

```
getSlingScriptHelper().getService(SampleOsgiService.class);
```

1. **What is Editable(dynamic) Template?**

   These editable templates (Dynamic) can be created dynamically at any time as per the requirement by author and it maintains connection after the page creation , all changes in dynamic template will reflect in all pages.

2. **Difference between static and dynamic templates?**

   Static templates do not maintain any connection between templates and pages. But editable templates maintain connection between template and existing pages.

3. **How to create Template type?**

   Ans: Creation of a template-type is a job of developer which helps template authors to create it's dynamic (editable) templates.

   Ex:

   - Go to /conf/my-project/settings/wcm/template-types. Create a node "empty-page" of type cq:Template
   - Create a node named "jcr:content" of type "cq:pageContent" under "empty-page" having "jcr:title" and "jcr:description".
   - Create "initial" of type "cq:page" inside "empty-page".
   - Create a Node "jcr:content" of type cq:pageContent under initial node and add a property "sling:resourceType" pointing to a page Component (myproject/components/page/page).
   - Create structure node hierarchy exactly like you have created initial hierarchy under "empty-page".
   - Now create policies hierarchy under "empty-page".

4. **What is the authors role in Editable (Dynamic) template?**

These editable templates will be created and managed by authors only.

5. **Explain about policies node?**

   There are two types of policies:

   Template Level Policy: This policy is used to define client side libraries of all the pages, created from a particular template.

   Component Level Policy: These policies are used to define components for a responsive grid (parsys) for all the pages created from the template.

6. **Explain about initial mode?**

   Initial Content mode is used to define content that will appear when a page is first created based on the template. The initial content can then be edited and removed by page authors.

7. **Explain about structure mode?**

   Ans: The structure allows you to define the structure of the template. The components defined in the template level can't be removed from the resulting page. If you want that template authors can add and remove components, add a paragraph system to the template. Components can be locked and unlocked to allow you to define initial content.

8. **Difference between structure and Initial content?**

   Ans: Intitial content could be edited in the resulted page but the content which was added in structure mode can't be edited in resulted page.

9. **How to configure design dialog data in Dynamic templates?**

   Ans: After adding design dialog in our component , need to go to edit template and need to click on policy of the corresponding component . There we need to create policy and set values under properties.

10. **Explain about layout mode?**

Ans: This mode will allow template authors to define layout for the page for different devices. By switching to Layout mode template authors can define layout for different components appearing on the page.

11. **Explain about locking mechanism in Dynamic templates?**

Ans: Once we lock the component in the structure mode , it can't be edited in the resulted page .ex: most of the cases header and footer get locked in structure mode .

Once we unlock the component in structure mode , it comes to initial mode and it can be edited in resulted page .

1) **What is workflow?**

Workflows enable you to automate Experience Manager activities. Workflows consist of a series of steps that are executed in a specific order. Each step performs a distinct activity such as activating a page or sending an email message.

2) **What is workflow console? What tabs all present in it?**

The Workflow console is the centralized location for workflow management in AEM. It can be accessed via Tools section of the touch-optimized UI.

http://localhost:4502/aem/worfklow

Within the Workflow console, there are several tabs:

- **Models -** Lists the workflow models currently available. Here you can create, edit or delete workflow models.
- **Instances -** Shows you details of workflow instances which are currently active. These instances are also version dependent.
- **Archive -** Enables you to access details of workflow instances which have terminated, for whatever reason.

- **Launcher -** Allows you to define a workflow to be launched if a specific node has been updated.
- **Failures -** Enables you to monitor and manage failed workflow instances.

3) **How to create custom process step in AEM? Give one example.**

The OSGI component needs to implement the WorkflowProcess interface with its execute() method.

execute method has three parameters :-

- **WorkItem** - A WorkItem is the unit that is passed through an Workflow instance of a WorkflowModel. It contains the WorkflowData the instances acts on and a reference to the WorkflowNode that describes the underlying workflow step. A WorkItem can be considered as a token in a petri net that flows through the net during execution.
- **WorkflowSession** - The WorkflowSession class provides all functionality (depending on the users rights) for managing WorkflowModels, Workflow instances and their execution.
- **MetaDataMap** - A value map for generic access to meta data values.

```
//This custom workflow step will use the AEM Replication API to replicate content from Author to Publish
@Component(service=WorkflowProcess.class, property = {"process.label=My Email Custom Step"})

public class CustomStep implements WorkflowProcess {

        /** Default log. */
        protected final Logger log = LoggerFactory.getLogger(this.getClass());

        @Reference
        private ResourceResolverFactory resolverFactory;
        //Inject a MessageGatewayService
        @Reference
        private MessageGatewayService messageGatewayService;

        private Session session;

        public void execute(WorkItem item, WorkflowSession
wfsession,MetaDataMap args) throws WorkflowException {
          try {
           // Busineess Logic
               } catch (Exception e)     {
                 e.printStackTrace() ;
               }
        }
}
```

4) What is dynamic Participant step? How to create dynamic Participant step?

Dynamic participant step assigns request to particular group/people dynamically at run time.

To define a dynamic participant step as an OSGI service component.

OSGI component needs to implement the ParticipantStepChooser interface and overrides getParticipant() method.

```
        @Component(service=ParticipantStepChooser.class, property =
        {"chooser.label=Sample Implementation of dynamic participant
        chooser"})
        public class ParticipantStep implements ParticipantStepChooser {

                public String getParticipant(WorkItem workItem,
        WorkflowSession wfSession, MetaDataMap metaDataMap)
```

```
                throws WorkflowException     {
                              String participant = "";
                // Business Logic to identify person or group dynamically
                return participant;
            }
}
```

5) **What is payload ?**

   A) Payload

6) *What are the different type of steps present in workflow?*

   Workflow models consist of a series of steps of various types which can be extended with scripts to provide the functionality and control you require.

   - *PROCESS STEP - It executes an ECMA script or calls an OSGi service to execute business logic.*
   - *PARTICIPANT STEP - It enables you to assign ownership for a person/group. 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; for example, a review step.*

   1) *CONTAINER STEP - A container step starts another workflow model that executes as a child workflow.*

   2) *OR SPLIT - It creates a split in the workflow, whereby only one branch is active. This allows you to introduce conditional processing paths into our workflow. more*

   3) *AND SPLIT - An AND Split creates a split in the workflow; both branches will be active. This component enables you to introduce multiple processing paths into the workflow. For example, you can allow certain review steps to occur in parallel, saving time. more*

   4) *GOTO STEP - The Goto step specifies the next step in the workflow model to execute, depending on the result of an ECMA script. More*

7) *Explain about Inbuilt workflows in AEM?*

AEM has some inbuilt workflows. Example

DAM MetaData Writeback - This workflow manages XMP write-back to the original binary and sets the last modified date in jcr.

If you want the feature to propagate metadata changes to select renditions of the asset, add the name of the renditions to the XMP Writeback Process workflow step of DAM Metadata WriteBack workflow. By default, this step is configured with the original rendition. more

Dam update asset - By default, upon image upload to CQ's Digital Asset Management, a "DAM Update Asset" workflow would be triggered and one of the many processes inside the workflow is to generate a web rendition of the uploaded image. And the default setting of CQ's Image API is to always render the web-enabled version of the uploaded image. This combination helps to limit the size and the quality of the image displayed thus reducing the page load time.

Approve for Adobe Campaign - Workflow handling the edition, review and approval of a newsletter to be sent using Adobe Campaign.
Other than this AEM has inbuilt workflow - Download Asset, Dam parse word document etc.

8) **What is difference between participant and dialog participant step?**

PARTICIPANT STEP - A participant step enables you to 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; for example, a review step.

DIALOG PARTICIPANT STEP - Use a Dialog Participant Step to collect information from the user who is assigned the work item. This step is useful for collecting small amounts of data that is used later in the workflow. Upon completing the step, the Complete Work Item dialog contains the fields that you define in your dialog. The data that is collected in the fields is stored in nodes of the workflow payload. Subsequent workflow steps can then read the value from the repository.

9) **Where I can see workflow model in crxde?**

Under /var/workflow/models

10) **What are the different built-in workflow users/groups present in AEM?**

We have two built-in workflow groups.

workflow-editors - Group that can create and modify workflow models.

workflow-users - A user participating in a workflow must be member of group workflow-users. This gives him or her full access to: /etc/workflow/instances so that he or she can update the workflow instance.

11) **How to trigger a workflow programmatically?**

Procedure to trigger a workflow programmatically.

Create a workflow session

```
WorkflowSession wfSession = workflowService.getWorkflowSession(session);
```

Get the workflow model

```
WorkflowModel wfModel = wfSession.getModel(model); # your workfllow model name
```

Get the workflow data.

```
WorkflowData wfData = wfSession.newWorkflowData("JCR_PATH", path);
```

Run the Workflow.

```
wfSession.startWorkflow(wfModel, wfData);
```

12) **How to pass values between two workflow steps?**

We can pass values between two workflow steps in **workitem**

- Set the value into first workflow step using code.

```
workItem.getWorkflowData().getMetaDataMap().put("firstStepMessage", "Hello second step");
```

- Get the value into Second workflow step using code.

```
workItem.getWorkflowData().getMetaDataMap().get("firstStepMessage", String.class);
```

13) **You want to define a Java based AEM workflow process step. Which interface must your class implement?**

com.day.cq.workflow.exec.**WorkflowProcess** interface

14) **You create a custom workflow process using the CQ API. Which method or methods needs to be overridden when your class implements the WorkflowProcess Interface?**

We need to override only execute method. It returns void.

15) **You want to define a Java based AEM workflow participant step. Which interface must your class implement?**

The com.day.cq.workflow.exec.ParticipantStepChooser interface

16) **You create a custom workflow participant using the AEM API. Which method or methods needs to be overridden when your class implements the ParticipantStepChooser Interface?**

We need to override only **getParticipant** method. It returns **String.**

17) **During a workflow step a comment is added requiring an update of a paragraph on a Web page. Where does the workflow author make the change?**

In the content page within the workflow step.

18) **You want to prompt a workflow user for information during the processing of a workflow step. Where is the content that the workflow user entered stored?**

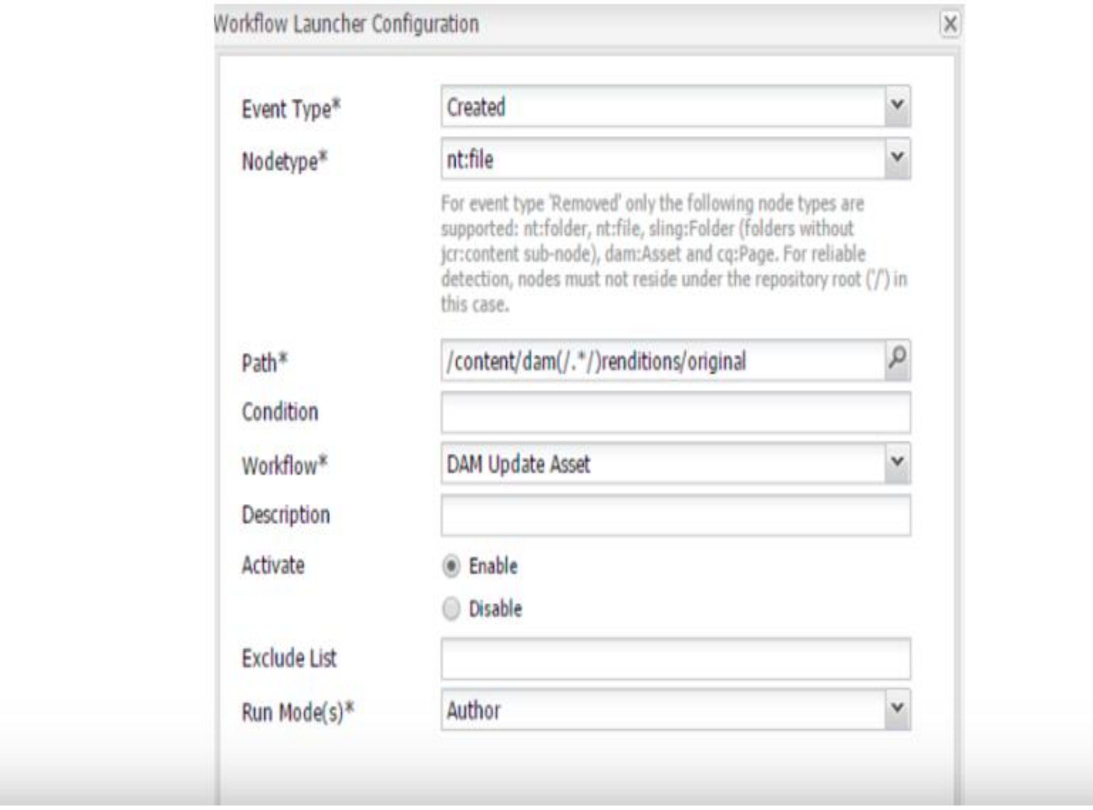
The content that the workflow user enters will be stored as part of the page.

19) **You want the process script to automatically advance the workflow to the next step after execution. Which dialog form field of a Process Step must be set to true?**

Handler Advance

20) **Workflow Launchers?**

Workflow launchers can be used to trigger the workflow based on some event that can be specified in the launcher. We can add a launcher under the Launcher tab.



Workflow Launcher Configuration

Event Type* Created

Nodetype* nt:file

For event type 'Removed' only the following node types are supported: nt:folder, nt:file, sling:Folder (folders without jcr:content sub-node), dam:Asset and cq:Page. For reliable detection, nodes must not reside under the repository root ('/') in this case.

Path* /content/dam(/.*/)renditions/original

Condition

Workflow* DAM Update Asset

Description

Activate ● Enable ○ Disable

Exclude List

Run Mode(s)* Author

- **Event type**:- can be either created, modified or deleted i.e. the action on which the launcher will trigger the workflow mentioned.
- **Nodetype**:- is for the type of the node on which the launcher will apply to start the workflow.
- **Path**:- specifies the path on which this action will be applied to during the workflow initiation i.e. on the which the launcher of the workflow will be applied to.
- Condition tab:- determines the condition to be applied when the workflow is launched. This is optional in case a user wants to start the launcher based on a specific condition only.
- **Workflow**:- Launcher needs to be associated with a workflow model so that it can be determined that what will be triggered with the help of this launcher.
- **Description:-** can be set and activate should be enabled to enable the launch of the workflow launcher.
- **Exclude:-** list is a comma separated list of JCR events or items to be ignored via workflow triggering.

- **Run modes:-** in this dialog is to specify the server on which this will be applied to (author, publish, author & publish).

1.

2. What is Content Fragment?

- AEM Content Fragments can be used to describe and manage structured content. Structured content is defined in models that a variety of content can contain types; including text, numerical data, Boolean, date and time, and more.
- They can also be delivered in JSON format which can be consumed by 3rd party applications.
- Content fragments are Stored as Assets
- Content fragments (and their variations) can be created and maintained from the Assets console.
- Authored and edited in the Content Fragment Editor.

3. Advantages of Content Fragments

The number of communication channels (ex: desktop, mobile, social media etc..) is increasing annually.

However, you (probably) do not want to use the same content for all channels - you need to optimize your content according to the specific channel.

Content fragments allow you to:

- Design content variations for specific channels.
- Can be export in JSON format without writing any code.
- Create and manage channel-neutral editorial content.
- Add images to your text by inserting assets (mixed-media fragments).

4. Different types of Content Fragments?

Content fragments can be either:

Simple fragments:

- No predefined structure
- Contains only text and images
- Based on Simple Fragment Template

Structured Fragments:

- Structured Content
- Based on Content Fragment Model which predefines a structure for the resulting fragment
- Contains structured data elements like text, dates, references, etc
- Used to realize Content Services using JSON Exporter

5. Features of Content Fragment?

- **Variations –** They are a significant feature of content fragments, as they allow you to create and edit copies of the master content for use on specific channels, and/or scenarios.
- **Versions –** Multiple versions of content fragments get created while working on them.
- **Synchronization of content across variations –** When editing a variation you have access to the action for synchronizing the current element of the variation with the Master. This allows you to automatically copy changes made to the Master to the required variation.
- **Visual difference of Content Fragment versions –** The Content Fragments can be compared with its versions, showing the differences between the versions and it can be reverted to a specific version.
- **Annotations of multi-line text elements –** Allows the author to annotate and communicate with other authors regarding the specific content fragment.
- **Intelligent summarization of multi-line text elements –** These features help users reduce the length of their text to a predefined number of words while keeping intact the key points and the overall meaning.
- **Import content of multi-line text elements –** Content of the element can be directly imported from a file from the computer.

- **Translation/localization –** Supports language copy workflows for content fragments

6. How to create simple content fragment?

Process of creating simple and structured content fragment is same, In 3rd we need to select template for simple content fragment and model for structed content fragment.

- Navigate to the Assets folder where you want to create the fragment.
- Select Create, then Content Fragment to open the wizard.
- Select template from list
- Enter CF (content fragment) title and description etc.
- Click on create

7. What is Content Fragment model?

Content Fragment Models define the structure of content for your content fragments.

The key concept with Content Fragments, is the authored content is presentation-agnostic, meaning its intended for multi-channel use where the consuming application, be that AEM, a single page application, or a Mobile app, controls how the content is displayed to the user.

8. How to enable Content Fragment model?

- Content Fragment Models must be enabled via AEM's Configuration Browser. It creates necessary folder structure under config folder.
- If Content Fragment Models are not enabled for a configuration, the Create > Content Fragment button will not appear for the relevant AEM configuration.

9. How to apply the Configuration to your Assets Folder?

Apply the WKND Mobile configuration (CF Configuration) to the WKND Mobile Assets Folder to allow Content Fragments from Content Fragment Models to be created within that Assets folder hierarchy:

- Navigate to AEM > Assets > Files
- Select the WKND Mobile folder
- Tap the Properties button in the top action bar to open Folder Properties

- In Folder Properties, tap the Cloud Services tab
- Verify the Cloud Configuration field is set to /conf/wknd-mobile
- Tap Save & Close in the upper-right to persist changes

10. How to create Content Fragment model?

- Navigate to Tools > Assets > Content Fragment Models.
- Tap the WKND Mobile folder to open.
- Tap Create to open the Content Fragment Model creation wizard.
- Enter Event as the Model Title (description is optional) and tap Create to save.

11. How can I add images in the Content Fragments?

Images can be added to and/or integrated with a content fragment by Inserting asset and Associated content options.

12. How to use Content Fragment on Page?

The Core Component Content Fragment Component allows for the inclusion of a content fragment on a page.

- The fragment and its properties can be selected in the configure dialog.
- Resource types to handle certain images and grids can be defined in the design dialog.
- The edit option will open the selected fragment within the content fragment editor.