**Questions & Answers**

**Basic Question**

1. I want to get the component name on jsp. How? 🡪${component.name}
2. Difference between parsys and iparsys?
3. Difference between cq:include and include page directive?  cq –runtime ; jsp –compile time
4. How to include jsp and how to include component in jsp using cq include? What is “path” attribute.
5. Why we use client library? How will you create client library? Embed and dependencies property?

**Dependencies** 🡪 If we add the dependencies property, For every dependency property value there is an individual hit to the server for loading these categories i.e. if you have four values in this property then 4 hits will be shown at debugging console network tab.

**Embed** 🡪 When we add embed property. For all embed property value there is only one hit to the server for loading these categories i.e. if you have four values in this property then only one combined hits will be shown at debugging console network tab.

1. While creating a template in the wizard what is the significance of allowed child and allowed parent?
2. In which scenario you feel, you need to create design dialog? Where the data of design dialog is saved in CRX? How can we access that property value?
3. I want to add a custom property to a resource. How will I do it programmatically?

Session session = request.getResourceResolver().adaptTo(Session.class);

resource = request.getResource();

Node node = resource.adaptTo(Node.class);

javax.jcr.Property nodeProperty = node.getProperty("cq:designPath");

Node newNode = node.addNode("newcomponent", "nt:unstructured");

newNode.setProperty("newCustomProperty", "New Custom Property Added");

session.save();

1. How can you delete a node programmatically?
2. I want to bulk changes, let’s say adding a property on all the pages in my project. What will I do? - groovy
3. Where implicit object in CQ is defined? Difference between properties and pageProperties implicit object? [cq:defineObjects in global.jsp]
4. Difference between a resource and resource Resolver?
5. How will you extend a component? Using resourceSupertype
6. What is a parbase component?
7. How can we create ValueMap ? How can we create a custom valuemap
8. How can we create a TOUCH UI dialog?

**Step Up Question**

1. How will you create a custom tags in bundles?
2. Advantage of using bundles in AEM ? How AEM manages of using different version of the same bundles?
3. What are run modes in CQ? Why we use run modes? How will you start CQ **with a specific run mode?**

Ans : Design dialog values are stored under design page located under **/etc/design/default** directory. If you set the a **template level** property named as **cq:designPath** pointing to design page ex. **/etc/design/geometrixx**, then all the design dialog property will be stored at

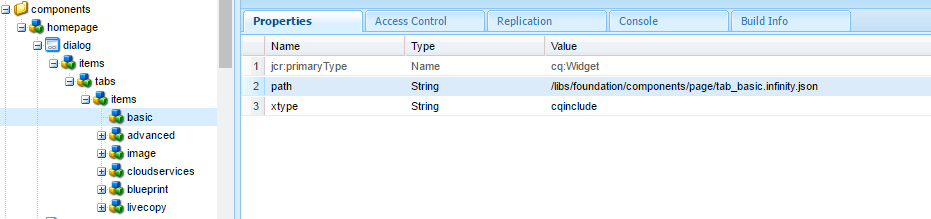
1. How to access the values of design dialog?
2. Define Apache Sling?
3. We want to track the operation whenever a new node is added in crx? How can we do it?- EventListener Interface 🡪 onEvent ()
4. **. How Cq manages to display images on multiple channels[like mobile /desktop/ipad]?With respect to size?**

**DAM Renditions** : The DAM allows you to create renditions of an image that include different sizes and versions of the same asset. You can use those renditions in the content of your website, and that can be helpful when you need to create thumbnails or smaller views of large, high-resolution images. When you are creating multi-device websites where Internet bandwidth is a varying concern, this feature becomes even more important. After all, you want to minimize the size of any files a site visitor has to download to view your site. Making your site load faster improves your search engine optimization and the user experience.

1. **What is overlaying in CQ?**
2. **Where exactly the OSGI Configuration gets saved ?**

**Questions of page Properties**

1. What is the significance if On Time and Off time? – Want to deactivate a page for certain period.
2. When you decide we have to make a java class a Component and When a service?
3. How will you extend a dialog? 🡪Let us consider we have a dialog of our custom component and we want to extend our dialog from other component’s dialog. For example to extend a Tab from the foundation page component dialog. Create a dialog of the custom component 🡪 Create a widget and extend it as above .example: Path : /libs/foundation/components/page/tab\_basic.**infinity.json 🡺**Xtype : **cqinclude**



1. II want to call a sling servlet when the home page loaded. – By registering the servlet with resourceType

**WorkFlow Question**

1. What are workflow model? And where I can able to find the workflow model in CRX? **🡪/etc/workflow/<custom\_workflow> folder in CRX**
2. What is a process step in work flow?
3. How will you create a custom workflow?

**public** **class** TrainingWorkFlowProcess **implements** **WorkflowProcess** {

@Override

**public** **void** **execute**(WorkItem item, WorkflowSession workflowSession,

MetaDataMap metaDataMap) **throws** WorkflowException {

item.getWorkflowData().getPayload().toString();

**/\*\* /content/geometrixx/de(Returns Page Path) \*\*/**

System.***out***.println("My Work Flow Process Executed");

}}

1. What is a payload in workflow? How will get the payload in the workflow(code)
2. Handler Advance in Workflow process step configuration? 🡪 proceed to next step, even if the exception occurs in the previous step

MSM

1. What is MSM?
2. What is difference between blue print and live copy?
3. What is a language copy?
4. What do understand by Rollout ?
5. In crx how will you identify the pages are Live copies ?
6. How will you create your own Rollout Configuration?
7. How will you maintain a multiligual site?

Replication

1. What is Replication? How replicaion happens in AEM?
2. What is reverse replication ? Use case reverse replication?

OSGI

1. **What is the difference between an OSGi bundle and a JAR file?**

* There is basically no difference. A JAR is a bundle and a bundle is a JAR, the formats are identical. However, a useful bundle requires **OSGi metadata in its manifest** **so that an OSGi framework can manage the visibility of classes between bundles**. A JAR without this metadata would only contain invisible classes, could not see any classes from other bundles, **nor could it get started in any way**. The Import-Package manifest header tells what packages should be made visible to the bundle, and the Export-Package defines the packages in the bundle that should be made visible to others. The key difference with OSGi is that a JAR is now all private, adding metadata in the manifest makes it a bundle that can safely share with other bundles.
* A bundle can be started and stopped but not the Jar file

1. **When you decide we have to make a java class a Component and When a service?**
2. **How will you access the method of a service in JSP?**
3. **sling.getService(TrainingRepoService.class).getDataFromMyMethod()**
4. **What we write in MANIFEST.MF file?**
5. **How will you create a OSGI Configuration? And how will you retrieve the value of the OSGI configuration?**

* **Using webcosole**

@Activate

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

myPropertyMap.put("myMessage",PropertiesUtil.*toString*(componentContext.getProperties().get("myMessage"), "myMessage"));

}

* **Using CRX[sling:OsgiConfig]**

@Reference

ConfigurationAdmin configurationAdmin;

**public** String sayHello() {

String configuredMessage = configurationAdmin.getConfiguration("com.adobe.sightly.training.impl.HelperServiceImpl").getProperties().

get("myMessage").toString();

}

**return** configuredMessage;

}

**AEM Query**

1. What do understand by Hits?
2. Query : Want to search all the pages which are created by some specific template? Top 10 result which are recently modified

type=cq:Page

1\_property=jcr:content/cq:template

1\_property.value=/apps/geometrixx/templates/homepage

1. How to prepropulate the drop down of a AEM dialog from external source?
2. How can you write a scheduler in AEM?

Sightly

|  |  |  |
| --- | --- | --- |
| **Block Statements[https://docs.adobe.com/docs/en/htl/docs/block-statements.html]** | | |
|  | **Syntax** | **Description** |
| use | data-sly-use | Initializes a helper object (defined in JavaScript or Java) and exposes it through a variable |
| unwrap | data-sly-unwrap | Removes the host element from the generated markup while retaining its content. This allows the exclusion of elements that are required as part of HTL presentation logic but are not desired in the actual output. |
| text | data-sly-text | Replaces the content of its host element with the specified text. |
| attribute | data-sly-attribute | Adds attributes to the host element. |
| element | data-sly-element | Replaces the element name of the host element. |
| test | data-sly-test | Conditionally removes the host element and it's content. A value of **false** removes the element; a value of **true** retains the element |
| list | data-sly-list | Repeats the content of the host element for each enumerable property in the provided object. |
| resource | data-sly-resource | Includes the result of rendering the indicated resource through the sling resolution and rendering process. |
| include | data-sly-include |  |
| template & call | data-sly-template data-sly-call |  |

Personalization

# Interview Questions

## Core Java

* What is Lambda Expression? Scenario?
* What is Functional Interface?
* Scenario for Java Stream API?
* Default method in Interface? What will happen in Multiple Inheritance?

## Maven

* Life Cycle of a Maven Build
* Maven Scope
* Compile
* Provided

## Spring Boot

* Spring BOM?
* Spring boot profiles and usage? And how to use specific profile?
* What is AutoConfiguration ?
* Enabling Cache?
* Cache Eviction?
* Lambok?
* Expain the concept of mono and flux?
* What is the meaning of Aspect-Oriented Programming (AOP)?

### Use case of using a static class

### What is static import

If we have to use any static variable or method from other class, usually we import the class and then use the method/variable with class name.

import java.lang.Math;

//inside class

double test = Math.PI \* 5;

Copy

We can do the same thing by importing the static method or variable only and then use it in the class as if it belongs to it.

import static java.lang.Math.PI;

//no need to refer class now

double test = PI \* 5;

Copy

Use of static import can cause confusion, so it’s better to avoid it. Overuse of static import can make your program unreadable and unmaintainable.

### What is a static block?

Java static block is the group of statements that gets executed when the class is loaded into memory by Java [ClassLoader](https://www.digitalocean.com/community/tutorials/java-classloader). It is used to initialize static variables of the class. Mostly it’s used to create static resources when class is loaded.

### What is a Marker interface?

A marker interface is an empty interface without any method but used to force some functionality in implementing classes by Java. Some of the well known marker interfaces are Serializable and Cloneable.

### 42. How to sort a collection of custom Objects in Java?

We need to implement Comparable interface to support sorting of custom objects in a collection. The Comparable interface has compareTo(T obj) method which is used by sorting methods and by providing this method implementation, we can provide default way to sort custom objects collection.

However, if you want to sort based on different criteria, such as sorting an Employees collection based on salary or age, then we can create Comparator instances and pass it as sorting methodology. For more details read [Java Comparable and Comparator](https://www.digitalocean.com/community/tutorials/comparable-and-comparator-in-java-example).

### 6. What are Wrapper classes?

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final. Java 5 autoboxing and unboxing allows easy conversion between primitive types and their corresponding wrapper classes.

Search in custom Object

* Why String is class is final

## Why is String immutable in Java?

String is immutable in Java because this offers several benefits:

* String pool is possible because String is immutable in Java.
* It increases security because any hacker can’t change its value and it’s used for storing sensitive information such as a database username or password.
* Since String is immutable, it’s safe to use in multi-threading and you don’t need any synchronization.
* Strings are used in Java class loaders and immutability provides assurance that the correct class is getting loaded by the ClassLoader class.

## What are the differences between String, StringBuffer, and StringBuilder in Java?

A String object is immutable and final in Java, so whenever you manipulate a String object, it creates a new String object. String manipulations are resource consuming, so Java provides two utility classes for string manipulations, StringBuffer and StringBuilder.

StringBuffer and StringBuilder are mutable classes. StringBuffer operations are thread-safe and synchronized, while StringBuilder operations are not thread-safe. You should use StringBuffer in a multi-threaded environment and use StringBuilderin a single-threaded environment. StringBuilder performance is faster than StringBuffer because of no overhead of synchronization.

Learn more about [the differences between String, StringBuffer and StringBuilder](https://www.digitalocean.com/community/tutorials/string-vs-stringbuffer-vs-stringbuilder) and benchmarking of [StringBuffer and StringBuilder](https://www.digitalocean.com/community/tutorials/string-vs-stringbuffer-vs-stringbuilder).

### 6. What is the difference between Checked and Unchecked Exceptions in Java?

1. Checked Exceptions should be handled in the code using try-catch block or else the method should use the throws keyword to let the caller know about the checked exceptions that might be thrown from the method. Unchecked Exceptions are not required to be handled in the program or to mention them in the throws clause of the method.
2. Exception is the superclass of all checked exceptions whereas RuntimeException is the superclass of all unchecked exceptions. Note that RuntimeException is the child class of Exception.
3. Checked exceptions are error scenarios that require to be handled in the code, or else you will get compile time error. For example, if you use FileReader to read a file, it throws FileNotFoundException and we must catch it in the try-catch block or throw it again to the caller method. Unchecked exceptions are mostly caused by poor programming, for example, NullPointerException when invoking a method on an object reference without making sure that it’s not null. For example, I can write a method to remove all the vowels from the string. It’s the caller’s responsibility to make sure not to pass a null string. I might change the method to handle these scenarios but ideally, the caller should take care of this.

### 7. What is the difference between the throw and throws keyword in Java?

throws keyword is used with method signature to declare the exceptions that the method might throw whereas throw keyword is used to disrupt the flow of the program and handing over the exception object to runtime to handle it.

### 8. How to write custom exceptions in Java?

We can extend Exception class or any of its subclasses to create our custom exception class. The custom exception class can have its own variables and methods that we can use to pass error codes or other exception-related information to the exception handler. A simple example of a custom exception is shown below.

package com.journaldev.exceptions;

import java.io.IOException;

public class MyException extends IOException {

private static final long serialVersionUID = 4664456874499611218L;

private String errorCode="Unknown\_Exception";

public MyException(String message, String errorCode){

super(message);

this.errorCode=errorCode;

}

public String getErrorCode(){

return this.errorCode;

}

}

* Cache and Cache Eviction Policy

## Spring Boot JPA

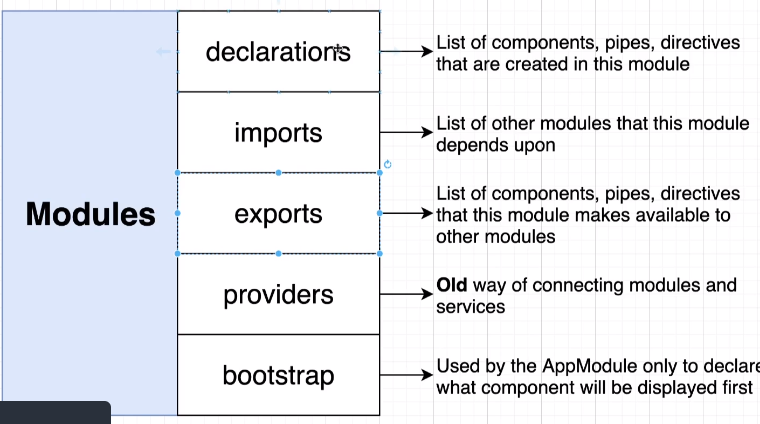
* What is JPA?
* Steps to create a CRUD based Application in Spring boot?
* How to save datetime, Not save and primary key
* What is the main difference between JPA and Hibernate?

## Spring Microservices

* What is service Registry and Discovery?
* Microservice communication?
* What is a Gateway?
* Fault tolerance?

## Angular

* Parent to child and vice versa?
* Communication between 2 disconnected component ?
* How to add a library in Angular project like boot strap?
* Local Reference ?
* How to create a resusable component?
* Content Projection?
* Observables?
* ViewChild and View Children?
* View Encapsulation?
* Angular Module



* Lazy Loading Routing?
* Query String in Routing . How to grab it?
* Pipes in Angular?

## JavaScript

1. What is Hoisting?
2. Let and var?
3. Difference between Spread and rest operator?
4. OOPS Concept in JS
   1. Prototype?
5. Apply(), call() and bind() methods
6. Promises and how to write promise?
7. Asycn and Await?
8. Why typescript?
9. What is Module in JS and advantage?
10. What is webpack?
11. Event Loop?

## NODE

* What is node?
* What is package .lock.json?
* What kind of App is suited for Node.
* If Node.js is single-threaded, then how does it handle concurrency
* Sever Side rendering and client Side
* Explain libuv

## Express JS

3. Middleware and calling a middle ware

4. Serving a static File.

next() ?

5. Passing Data from one middleware to another

6.Calling the middleware async

7. Setting Env varibales

8.Creating child process

39. How does Node.js handle the child threads?

In general, Node.js is a single threaded process and doesn’t expose the child threads or thread management methods. But you can still make use of the child threads using spawn() for some specific asynchronous I/O tasks which execute in the background and don’t usually execute any JS code or hinder with the main event loop in the application. If you still want to use the threading concept in your application you have to include a module called ChildProcess explicitly.

31. Differentiate between spawn() and fork() methods in Node.js?

In Node.js, the spawn() is used to launch a new process with the provided set of commands. This method doesn’t create a new V8 instance and just one copy of the node module is active on the processor. When your child process returns a large amount of data to the Node you can invoke this method.

Syntax:

1

child\_process.spawn(command[, args][, options])

Whereas, the fork() in Node.js is a special instance of spawn() that executes a new instance of the V8 engine. This method simply means that multiple workers are running on a single Node code base for various task.

Syntax:

1

child\_process.fork(modulePath[, args][, options])

Log4j issue?

## Explain the Single Responsibility Principle (SRP)?

CREATING A CLASS WHICH RESEMBELS THE FACTORY PATTERN

## What does SOLID stand for? What are its principles?

**Mid**

[](https://www.fullstack.cafe/interview-questions/software-architecture" \o "Software Architecture Interview Questions)**[Software Architecture](https://www.fullstack.cafe/interview-questions/software-architecture" \o "Software Architecture Interview Questions)**[85](https://www.fullstack.cafe/interview-questions/software-architecture" \o "Software Architecture Interview Questions)

**Answer**

**S.O.L.I.D and Example** is an acronym for the first five object-oriented design (OOD) principles by Robert C. Martin.

* **S** - Single-responsiblity principle. A class should have one and only one reason to change, meaning that a class should have only one job.
* **O** - Open-closed principle. Objects or entities should be open for extension, but closed for modification.
* **L** - Liskov substitution principle. Let q(x) be a property provable about objects of x of type T. Then q(y) should be provable for objects y of type S where S is a subtype of T.
* **I** - Interface segregation principle. A client should never be forced to implement an interface that it doesn't use or clients shouldn't be forced to depend on methods they do not use.
* **D** - Dependency Inversion Principle. Entities must depend on abstractions not on concretions. It states that the high level module must not depend on the low level module, but they should depend on abstractions.

## What's the difference between Deadlock and Livelock?

## What is the difference between Cohesion and Coupling?

**Q: What is Optional in Java 8?  
A:**Java 8 introduced a new **container class java.util.Optional<T>**. It wraps a single value, if that value is available. If the value is not available an empty optional should be returned. Thus it represents null value with absent value. This class has various utility methods like isPresent() which helps users to avoid making use of null value checks. So instead of returning the value directly, a wrapper object is returned thus users can avoid the null pointer exception.  
[Java 8 Optional using example.](https://www.javainuse.com/java/java8_optional)

**Q01.Scenario**: You need to load stock exchange security codes from a database and cache them for performance. The security codes need to be refreshed say every 30 minutes. This cached data needs to be populated and refreshed by a single writer thread and read by several reader threads. How will you ensure that your read/write solution is scalable and thread safe?

