



Adobe AD0-E123 Practice Questions

Shared by Stevenson on 19-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

Which offloading-related service is used by each AEM instance to connect to AEM Topology Connector?

Options:

- A- Cluster Service
- B- Discovery Service
- C- Topology Connector Service

P2P
exams

Answer:

B

Explanation:

The Discovery Service is an offloading-related service used by each AEM instance to connect to AEM Topology Connector. The Discovery Service is part of the Sling Discovery framework, which is responsible for detecting and managing the topology of AEM instances. It enables instances to discover each other and establish a clustered topology, which is essential for load-balancing, failover, and other distributed operations.

Option A, 'Cluster Service,' is not a standard term used in AEM for connecting to the Topology Connector.

Option C, 'Topology Connector Service,' is a slightly misleading term; the correct term is Discovery Service, which performs the role described.

P2P
exams

Question 2

Question Type: MultipleChoice

Which plugin cleans the container project's target folder before Maven build?

Options:

- A- mvn-clean-plugin

- B- maven-scr-plugin
- C- maven-c lean-plugIn

Answer:

C

Explanation:

In Maven, the maven-clean-plugin is used to clean the project by removing files generated at build-time in the target directory. Before a new build, cleaning the project is a common practice to ensure that no stale artifacts affect the new build. The plugin's goal clean is bound to the clean phase of the build lifecycle and can be executed with the command mvn clean.

Option A, mvn-clean-plugin, is incorrectly named; the correct name is maven-clean-plugin.

Option B, maven-scr-plugin, is related to Service Component Runtime and not to cleaning build artifacts.

Question 3

Question Type: MultipleChoice

How would a developer define global project properties in Maven project structure?

Options:

- A- section of the 'core' package pom.xml
- B- section of the 'all' content package pom.xml
- C- section of the parent pom.xml

Answer:

C

Explanation:

Global project properties in a Maven project are typically defined in the section of the parent pom.xml file. This allows these properties to be shared across all modules of the project, ensuring consistency and easy maintenance of project-wide settings such as versions of dependencies, plugin configurations, and other properties that are common across multiple

modules of the project.

Question 4

Question Type: MultipleChoice

What would be the packageType for immutable code packages in Maven project structure?

Options:

- A- container
- B- content
- C- application

Answer:

C

Explanation:

In Maven for AEM, application is the packageType that is typically used for immutable code packages. These are packages that contain code that is not expected to change between deployments, such as OSGi bundles and certain content packages that do not contain user-editable content. This type of package is intended to be deployed to an AEM instance as part of the application's codebase, and the application packageType signifies this use case.

container is not a standard package type in AEM Maven project structures.

content is typically used for packages that contain mutable content, such as pages and configurations that can be edited by authors.

Question 5

Question Type: MultipleChoice

A developer wants to be able to execute the following query:

SELECT

*

FROM [ntbase] AS s

WHERE

s.status = 'STARTED'

Which two options are mandatory additions to the Index? (Choose two.)

Options:

- A- ntbase index rule
- B- status properly with property Index = false
- C- ntbase aggregate
- D- status property with propertyIndex = true

Answer:

A, D

Explanation:

When creating a custom query in AEM's JCR (Java Content Repository), the query's performance is highly dependent on the indexing configuration. For the given query that selects all nodes with a 'status' property equal to 'STARTED' from the 'nt:base' node type, the index must be set up correctly:

Option A, 'nt:base index rule', is correct. The index rule for 'nt:base' must be added to define which properties of nodes of this type are indexed.

Option D, 'status property with **propertyIndex = true**', **is also correct**. This index ensures that queries filtering on the 'status' **property are executed efficiently**. The property index should be set to true, which means that this property is indexed and the query will use this index to filter the results.

Option B is incorrect because setting the property index to false would mean that the property is not indexed, making the query less efficient since the repository would need to scan each node to find matches.

Option C, 'nt:base aggregate', is not mandatory. Aggregates are used to include properties of related nodes in the index, but they are not required for a simple property match like the one in the given query.

Question 6

Question Type: MultipleChoice

Review the following HTL code snippet:

