



**Free Questions for AD0-E134 by certsdeals**

**Shared by Garcia on 12-12-2023**

**For More Free Questions and Preparation Resources**

**Check the Links on Last Page**

# Question 1

---

## Question Type: MultipleChoice

---

An AEM development team is working on a new multi-country application using AEM as a Cloud Service. A developer has been assigned the task for building the integration with a third-party web service. A secret key is needed to connect with this web service. The website creators will provide this key. The key is different for each type of environment (dev, stage and production)

What is the recommended way to make the secret key available in the AEM application?

### Options:

---

- A- Use a context aware configuration
- B- Read the key value from a property file stored in the code base
- C- Use an environment variable which is then consumed by an OSGi configuration
- D- Read the key value from OSGi configuration stored in run nNdes

### Answer:

---

C

### Explanation:

---

The recommended way to make the secret key available in the AEM application is to use an environment variable which is then consumed by an OSGi configuration. This way, the secret key is not stored in the code base or in the repository, but rather in a secure and encrypted way in the Cloud Manager environment variables. The OSGi configuration can use a placeholder to reference the environment variable and inject its value at runtime. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/configuring-osgi.html?lang=en#environment-variables>  
<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/environment-variables.html?lang=en>

## Question 2

---

**Question Type:** MultipleChoice

---

Which Maven plugin checks if all the requirements declarations made in OSGi bundles are satisfied by the capabilities declarations of other bundles included in the Maven project?

**Options:**

---

**A-** maven-enforcer-plugin

**B-** femaven-assembly-plugin

C- content-package-maven-plugin

D- aeanalyser-maven-plugin

**Answer:**

---

D

**Explanation:**

---

The aeanalyser-maven-plugin is a Maven plugin that checks if all the requirements declarations made in OSGi bundles are satisfied by the capabilities declarations of other bundles included in the Maven project. This plugin ensures that the OSGi bundles are consistent and can be resolved at runtime. The plugin also checks for other issues such as API compatibility, package versioning, and bundle start order. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/developing/aem-project-content-package-structure.html?lang=en#build-analyzer-maven-plugin> <https://github.com/adobe/aeanalyser-maven-plugin>

## Question 3

---

**Question Type:** MultipleChoice

---

What is the recommended path to override /libs standard functionality?

### Options:

---

- A- /conf
- B- /apps
- C- /content
- D- /libs

### Answer:

---

B

### Explanation:

---

The recommended path to override /libs standard functionality is /apps. The /apps folder contains application-specific code and configuration that can extend or override the default functionality provided by /libs. The /apps folder has precedence over the /libs folder, meaning that if there are two resources with the same path under both folders, the one under /apps will be used. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/developing/aem-project-content-package-structure.html?lang=en#repository-restrictions> <https://experienceleague.adobe.com/docs/experience-manager-learn/foundation/development/set-up-a-local-aem-development-environment.html?lang=en#the-aem-repository>

## Question 4

---

**Question Type: MultipleChoice**

---

A development team is starting a new AEM project that is going to integrate with the Adobe Commerce platform. The developer needs to create a new AEM project using the Maven command line interface.

How can the 'mvn -B archetype:generate' command help the developer with the integration between AEM and Adobe Commerce?

**Options:**

---

- A-** Using the property 'commerceModule=AdobeCommerce' can provide a path to an external jar that integrates between the platforms.
- B-** Using the property ,aemVersion=cloudl automatically provides a report with integration guidelines.
- C-** Using the property 'includeCommerce=y'1 the command will generate specific Commerce Core Components.

**Answer:**

---

C

**Explanation:**

---

The includeCommerce property is a boolean flag that indicates whether to include Commerce Core Components in the project or not. If set to y, the command will generate a commerce module that contains specific Commerce Core Components and their dependencies. These components can be used to integrate AEM with Adobe Commerce platform or other commerce solutions. Reference: <https://experienceleague.adobe.com/docs/experience-manager-core->

[components/using/developing/archetype/using.html?lang=en#properties](https://experienceleague.adobe.com/docs/experience-manager-core-components/using/components/commerce.html?lang=en#properties) <https://experienceleague.adobe.com/docs/experience-manager-core-components/using/components/commerce.html?lang=en>

## Question 5

---

**Question Type:** MultipleChoice

---

Which AEM dependency provides APIs for all AEM environments?

A)

```
<dependency>  
<groupId>com.adobe.aem</groupId>  
<artifactId>aem-api</artifactId>  
<version>6.5.0</version>  
<classifier>apis</classifier>  
<scope>provided</scope>  
</dependency>
```

B)

```
<dependency>  
<groupId>com.adobe.cq</groupId>  
<artifactId>core.wcm.components.content</artifactId>  
<type>zip</type>  
</dependency>
```

C)

```
<repository>  
<id>adobe-public-releases</id>  
<name>Adobe Public Repository</name>  
<url>https://repo.adobe.com/nexus/content/groups/public/</url>  
<layout>default</layout>  
</repository>
```

D)



```
<dependency>  
<groupId>com.adobe.aem</groupId>  
<artifactId>uber-jar</artifactId>  
<version>6.5.0</version>  
<classifier>apis</classifier>  
<scope>provided</scope>  
</dependency>
```

**Options:**

---

- A- Option A
- B- Option B
- C- Option C
- D- Option D

**Answer:**

---

C

**Explanation:**

---

Option C is the AEM dependency that provides APIs for all AEM environments. Option C uses the uber-jar dependency with the provided scope and the classifier set to apis. The uber-jar dependency contains all the public Java APIs that are available in AEM. The provided scope indicates that the dependency is not required for runtime, as it is already provided by the AEM platform. The apis classifier indicates that only the API classes are included in the dependency, not the implementation classes. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/developing/aem-project-content-package-structure.html?lang=en#project-dependencies> <https://helpx.adobe.com/experience-manager/6-4/sites/developing/using/html/eclipse-setup.html>

## Question 6

---

**Question Type:** MultipleChoice

---

On package install content that is already present in the repos must not be overwritten and if not present in the repos it must not be removed.

Which import mode should the developer use?

**Options:**

---

**A-** update

**B-** replace

**C-** merge

**Answer:**

---

A

**Explanation:**

---

The merge import mode is used to import content that is already present in the repository without overwriting it and without removing content that is not present in the package. This mode preserves existing content properties and nodes while adding new ones from the package. This mode is useful for incremental updates or initial content seeding. Reference:

[https://jackrabbit.apache.org/filevault/vaultfs.html#Import\\_Modes](https://jackrabbit.apache.org/filevault/vaultfs.html#Import_Modes) <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/content-package-import-modes.html?lang=en>

## Question 7

---

**Question Type:** MultipleChoice

---

In an AEM as a Cloud Service environment, which repository areas are considered as immutable?

### Options:

---

- A- /content and /libs
- B- /content and /etc
- C- /apps and /libs
- D- /apps and /content

### Answer:

---

C

### Explanation:

---

In AEM as a Cloud Service environment, the repository areas /apps and /libs are considered as immutable, meaning that they cannot be modified at runtime by any means. These areas contain application code and configuration that are deployed via Cloud Manager and are read-only for security and performance reasons. Any changes to these areas require a new deployment via Cloud Manager. Reference: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/deploying/overview.html?lang=en#repository-restrictions> <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/developing/aem-project-content-package-structure.html?lang=en#repository-restrictions>

## Question 8

---

**Question Type: MultipleChoice**

---

A developer needs to create a project based on AEM Project Archetype with a specific AEM as a Cloud Service SDK version on the local environment.

Which two properties must be defined when creating this project? (Choose two.)

**Options:**

---

- A- aemVersion=cloud
- B- sdkVersion=2022.5.7575.20220530T152407Z-220401
- C- sdkVersion=latest
- D- aemVersion=latest
- E- aemVersion=2022.5.7575.20220530T152407Z-220401

**Answer:**

---

A, B

**Explanation:**

---

The aemVersion property specifies the target AEM version for the project. To create a project based on AEM as a Cloud Service SDK, the value should be set to cloud. The sdkVersion property specifies the AEM as a Cloud Service SDK version to use for local

development. To use a specific SDK version, the value should be set to the exact version number, such as 2022.5.7575.20220530T152407Z-220401. Reference: <https://experienceleague.adobe.com/docs/experience-manager-core-components/using/developing/archetype/using.html?lang=en#properties> <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/developing/aem-as-a-cloud-service-sdk.html?lang=en#sdk-versions>

## Question 9

---

**Question Type: MultipleChoice**

---

The following anchor tag is not resolving:

`{item.name}`

Upon further inspection the developer notices that the link has no .html appended to the end of the URL

What could be a potential fix for the issue?

A)

```
<a href="{item.path}@extension = 'html'" >{item.name}</a>
```

B)

`<a href="item.path@context = ' unsafe, fragment = item.name' >{item.name}</a>`

C)

`<a href="item.path@append = ' html'" >{item.name}</a>`

D)

`<a href="item.path@context = ' html'" >{item.name}</a>`

**Options:**

---

A- Option A

B- Option B

C- Option C

D- Option D

**Answer:**

---

B

## Explanation:

---

Option B is a potential fix for the issue. Option B uses the data-sly-attribute block statement to add the href attribute to the anchor tag. The data-sly-attribute statement uses an expression to append ".html" to the item.path value. This way, the link will have the correct extension and will resolve to the corresponding page. Reference: <https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-block-statements.html?lang=en#data-sly-attribute>

## Question 10

---

### Question Type: MultipleChoice

---

An application development team needs to create a multi-channel application that uses AEM as a headless CMS Which feature should be used to maintain structured data for this application?

### Options:

---

- A- Static template
- B- Content fragment
- C- Experience fragment



D- Custom component

**Answer:**

---

B

**Explanation:**

---

A content fragment is a reusable piece of content that can be authored and managed independently from a page. Content fragments are based on structured data models that define the elements and data types of the content. Content fragments can be delivered as JSON using various APIs, such as Assets HTTP API, Content Fragments REST API, or Content Fragments GraphQL API. Content fragments are suitable for creating multi-channel applications that use AEM as a headless CMS. Reference:

<https://experienceleague.adobe.com/docs/experience-manager-learn/foundation/headless/introduction-to-headless.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-65/assets/fragments/content-fragments.html?lang=en>

## Question 11

---

**Question Type:** MultipleChoice

---

Which two unit testing dependencies are generated by AEM archetype? (Select two.)

### Options:

---

- A- JUnit
- B- Selenium
- C- PowerMock
- D- Mockito
- E- Hobbes

### Answer:

---

A, D

### Explanation:

---

JUnit and Mockito are two unit testing dependencies that are generated by AEM archetype. JUnit is a framework for writing and running unit tests in Java. Mockito is a framework for creating and using mock objects in unit tests. AEM archetype also adds Apache Sling Mocks and AEM Mocks Test Framework by io.wcm as dependencies for unit testing. Reference:  
<https://experienceleague.adobe.com/docs/experience-manager-learn/getting-started-wknd-tutorial-develop/project-archetype/unit-testing.html?lang=en> <https://experienceleague.adobe.com/docs/experience-manager-core-components/using/developing/archetype/using.html?lang=en>

**To Get Premium Files for AD0-E134 Visit**

**<https://www.p2pexams.com/products/ad0-e134>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/adobe/pdf/ad0-e134>**

