

## AD0-E134<sup>Q&As</sup>

Adobe Experience Manager Developer Exam

**Pass Adobe AD0-E134 Exam with 100% Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.leads4pass.com/ad0-e134.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by Adobe  
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



**QUESTION 1**

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?

- 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

Correct 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. References: <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**

Which configuration must be applied to enable re-fetching of cached items based on Cache Headers sent by AEM?

- A. /autoInvalidate true
- B. /autoInvalidate "1"
- C. /enableTTLtrue
- D. /enableTTL "1"

Correct Answer: D

Explanation: The /enableTTL "1" configuration must be applied to enable re-fetching of cached items based on Cache Headers sent by AEM. This configuration enables the Time To Live (TTL) feature of dispatcher, which allows dispatcher to check the Cache-Control and Expires headers of the cached files and re-fetch them from AEM if they are expired. References: <https://experienceleague.adobe.com/docs/experience-manager-dispatcher/using/configuring/page-invalidation.html?lang=en#time-based-cache-invalidation>

---

**QUESTION 3**

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?

- A. Static template
- B. Content fragment
- C. Experience fragment
- D. Custom component

Correct 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. References:

<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 4

A snippet throws an exception at runtime:

```
@Model(adaptables = {Resource.class}) public class MyCustomModel {  
(SlingObject  
private Resource resource;
```

```
@Inject  
private Page currentPage;  
  
private String currentPagePath;  
  
@PostConstruct  
protected void init() {  
    this.currentPagePath = currentPage.getPath();  
}
```

What should the developer add to fix it?

- A. defaultInjectionStrategy = DefaultInjectionStrategy property to @Model Class annotation

- B. (?ptional annotation to page field
- C. throws Exception at the end of the init method declaration
- D. SlingHttpServletRequest.class to adaptables property of ?odel Class annotation

Correct Answer: A

The developer should add the defaultInjectionStrategy = DefaultInjectionStrategy property to the @Model Class annotation to fix the snippet. The defaultInjectionStrategy property defines how the Sling Model handles missing or null values for the injected fields. By default, the Sling Model uses the REQUIRED injection strategy, which means that all fields must have a non-null value or else an exception is thrown. By setting the defaultInjectionStrategy property to OPTIONAL, the Sling Model allows null values for the injected fields and does not throw an exception. This way, if the page field is null because the resource is not a page, the Sling Model can still work without errors. References: <https://sling.apache.org/documentation/bundles/models.html><https://sling.apache.org/documentation/bundles/models.html#optional-injection>

---

## QUESTION 5

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?

- A. maven-enforcer-plugin
- B. femaven-assembly-plugin
- C. content-package-maven-plugin
- D. aemanalyser-maven-plugin

Correct Answer: D

Explanation: The aemanalyser-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. References: <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/aemanalysermaven-plugin>