

# 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 application is expected to export a content fragment in JSON format without any customization for a headless implementation.

What is the recommended approach?

- A. Use AEM Assets HTTP API
- B. Use Core components to export JSON
- C. Use Sling Exporter framework

Correct Answer: A

Explanation: AEM Assets HTTP API is a RESTful API that allows access to content fragments in JSON format without any customization. The API supports CRUD operations on content fragments and their variations, as well as querying and searching for content fragments based on metadata or full-text search.

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

---

**QUESTION 2**

A developer needs to create a new Title component. The requirements are:

1.

The layout must be the same as the Title core component

2.

The text property must have the page title as prefix (e.g., Page Title - )

3.

The component must be reusable

Which approach is recommended?

A. 1. Create a Proxy Component of Title core component

2.

Create a Custom Sling Model that overrides the default behavior

3.

Customize the component template

B. 1. Create a custom component from scratch

2.

Create a Custom Sling Model for the component that follows the requirement

3.

Create a Model Exporter

C. 1. Create a Proxy Component from Title core component

2. Create a Custom Sling Model that overrides the default behavior

Correct Answer: A

Explanation: A proxy component is a site-specific component that inherits from a core component and allows customization of the component name, group, dialog, and behavior. A proxy component can refer to any version of the core component by changing the sling:resourceSuperType property. A custom sling model can be used to implement the logic for adding the page title as prefix to the text property. A component template can be used to define the layout of the component. Reference: 1 Using Core Components section 2 Create Proxy Component in AEM section 3 AEM-Create Proxy Component section 4 Proxy Components in AEM 6.4 section 5 AEM Proxy Component Pattern and Component Versioning section

---

### QUESTION 3

An AEM Developer receives requirements for Sling Models in a human-readable yaml format. A custom application needs to be built. The dependency is as shown:

```
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.8.4</version>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.dataformat</groupId>
  <artifactId>jackson-dataformat-yaml</artifactId>
  <version>2.8.4</version>
</dependency>
```

A. 1. Create OSGI models to export as yaml

2. Configure mime type in Apache Sling MIME Type Service

B. 1. Create OSGI models to export as yaml

2. Configure mime type in Apache Sling Servlet/Script Resolver and Error Handler

C. 1. Create Sling models to export as yaml

2. Configure mime type in Apache Sling MIME Type Service

D. 1. Create Sling models to export as yaml

## 2. Configure mime type in Apache Sling Referrer Filter

Correct Answer: C

Explanation: To create Sling Models that can export data in a human-readable yaml format, the following steps are required: Create Sling models to export as yaml. Sling models are Java classes that can be used to represent resources in AEM. Sling models can use annotations to define how they can be adapted from a resource and how they can export data in different formats. To export data in yaml format, the Sling model class needs to use the @Model annotation with the resourceType parameter set to the resource type of the resource that the model represents. The Sling model class also needs to implement the org.apache.sling.models.annotations.Exporter annotation with the name parameter set to "jackson" and the extensions parameter set to "yaml". The Sling model class also needs to use the @JsonProperty annotation on the fields or methods that need to be exported in yaml format. Configure mime type in Apache Sling MIME Type Service. The Apache Sling MIME Type Service is an OSGi service that maps file extensions to MIME types and vice versa. To enable the yaml format for Sling models, the MIME Type Service needs to be configured with a new entry for the yaml extension and its corresponding MIME type, which is "application/x-yaml". This can be done by creating an OSGi configuration for the org.apache.sling.commons.mime.internal.MimeTypeServiceImpl service and adding the entry "yaml=application/x-yaml" to the mime.types property. References: <https://sling.apache.org/documentation/bundles/models.html> <https://sling.apache.org/documentation/bundles/mime-type-support-commons-mime.html>

---

### QUESTION 4

A developer has to create a Logger and Writer pair for the company's application logging. Which OSGi configurations should the developer use?

- A. Apache Sling Logging Logger Configuration and Apache Sling Logging Configuration
- B. Apache Sling Request Logger and Apache Sling Logging Writer Configuration
- C. Apache Sling Logging Logger Configuration and Apache Sling Logging Writer Configuration

Correct Answer: C

Explanation: The Apache Sling Logging Logger Configuration and Apache Sling Logging Writer Configuration are the OSGi configurations that the developer should use to create a Logger and Writer pair for the company's application logging. The Logger Configuration defines the log level and the log file name for a given logger name or category. The Writer Configuration defines the file size, number of files, and file location for a given log file name.

References: <https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/configure-logging.html?lang=en#configuring-log-files>

---

### QUESTION 5

An AEM application development team is assigned a task to create an Event-Driven Data Layer implementation for an Analytics solution. Which Adobe recommended best practice should the developer choose?

- A. Use Adobe Experience Platform's data layer to integrate with AEM.
- B. Create a custom data layer and add each component template, and its properties to the data layer
- C. Use Adobe Client Data Layer and integrate with Core components.
- D. Create an Adobe Cloud Service configuration to use third-party tool's data layer.

Correct Answer: C

Adobe Client Data Layer is a JavaScript library that provides a standardized way to collect, structure, and manage data on a web page. It can be used to implement an event-driven data layer for analytics solutions. It integrates with Core components and allows authors to configure data layer properties for each component. It also supports custom events and data sources. References: <https://experienceleague.adobe.com/docs/experience-manager-core-components/using/developing/data-layer.html?lang=en><https://github.com/adobe/adobe-client-data-layer>

## QUESTION 6

A developer needs to create sling models for two fields name and occupations. The dialog has two fields, name - a single value field, and occupations - a multi value field. The following code is included in sling models inherited from interface com.adobe.aem.guides.wknd.core.models.Byline

```
package com.adobe.aem.guides.wknd.core.models.impl;
.....
public class BylineImpl implements Byline {
    .....
    @Override
    public List<String> getOccupations() {
        if (occupations != null) {
            Collections.sort(occupations);
            return new ArrayList<String>(occupations);
        } else {
            return Collections.emptyList();
        }
    }
    .....
}
```

- Ⓐ 

```
<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>
```
- Ⓑ 

```
<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline.impl"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>
```
- Ⓒ 

```
<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name}</h2>
    <p class="cmp-byline__occupations">${byline.occupations }</p>
</div>
```
- Ⓓ 

```
<div data-sly-use.byline="com.adobe.aem.guides.wknd.core.models.Byline"
    data-sly-use.placeholderTemplate="core/wcm/components/commons/v1/templates.html"
    data-sly-test.hasContent="${!byline.empty}"
    class="cmp-byline">

    <h2 class="cmp-byline__name">${byline.name @ join=', '}</h2>
    <p class="cmp-byline__occupations">${byline.occupations @ join=', '}</p>
</div>
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

Explanation: Option C is the correct implementation for the Sling Model. Option C uses the @Model annotation with the adaptables parameter set to Resource.class. This allows the Sling Model to adapt from a resource object and access its properties using the ValueMap interface. Option C also uses the @Inject annotation with the name parameter set to "/name" and "/occupations" to inject the values of the name and occupations properties into the name and occupations fields. Option C also uses the @Named annotation with the value parameter set to "byline" to specify the name of the Sling Model that can be used in HTL scripts. References:

<https://sling.apache.org/documentation/bundles/models.html><https://experienceleague.adobe.com/docs/experience-manager-htl/using-htl/htl-block-statements.html?lang=en#use>

---

## QUESTION 7

What two types of testing are available OOB in AEM Cloud Manager Pipeline? (Select Two.)

A. Code Quality testing

B. Performance testing

C. UI testing

D. Penetration testing

E. Integration testing

Correct Answer: AC

Explanation: Code Quality testing and UI testing are two types of testing that are available OOB in AEM Cloud Manager Pipeline. Code Quality testing checks the code quality of the project using SonarQube and reports any issues or

vulnerabilities. UI testing checks the functionality and usability of the project using Selenium WebDriver and reports any errors or failures.

References: <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/testing/testing-overview.html?lang=en#testing-types> <https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/testing/code-quality-testing.html?lang=en>

<https://experienceleague.adobe.com/docs/experience-manager-cloud-service/implementing/testing/ui-testing.html?lang=en>

---

## QUESTION 8

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>

---

## QUESTION 9

A client is having issues with some query results:

Many of the client's industry terms have the same meaning, and users do not always search the exact wording. Many users search by typing in short phrases instead of exact keywords, ex: "cats and dogs"

What index analyzers should the AEM developer recommend?

- A. 1. Add a Mapping filter to the current indexes
- 2. Add a Stop filter to the current indexes
- B. 1. Tokenize the current indexes with a Keyword tokenizer
- 2. Add a Mapping filter to the current indexes
- C. 1. Add a Synonym filter to the current indexes
- 2. Add a Stop filter to the current indexes
- D. 1. Add a Synonym filter to the current indexes
- 2. Add a LowerCase filter to the current indexes

Correct Answer: D

A Synonym filter can help to map different terms that have the same meaning, such as "cat" and "feline". A LowerCase filter can help to normalize the case of the terms, so that "cats and dogs" and "Cats and Dogs" are treated the same. Reference: 1 Lucene Analyzers section

---

## QUESTION 10

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>
```

A. Option A



B. Option B

C. Option C

D. Option D

Correct Answer: D

[AD0-E134 VCE Dumps](#)

[AD0-E134 Study Guide](#)

[AD0-E134 Exam Questions](#)