AD0-E134 Q&As

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.certbus.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

What is Out of Scope for the Pattern Detector tool, while doing an AEM upgrade?

A. OSGi bundles exports and imports mismatch

B. Backward Compatibility with the previous AEM Version

C. Definitions of Oak indices for compatibility

D. rep:User nodes compatibility (in context of OAuth configuration)

Correct Answer: B

Explanation: Backward Compatibility with the previous AEM Version is out of scope for the Pattern Detector tool, while doing an AEM upgrade. The Pattern Detector tool is a tool that scans an existing AEM code base and identifies patterns that are incompatible with AEM as a Cloud Service or newer versions of AEM on-premise. The tool does not check for backward compatibility with older versions of AEM, as this is not a requirement for upgrading. References: https://experienceleague.adobe.com/docs/experience-manager- cloud-service/moving/cloud-migration/pattern-detector.html?lang=enhttps://experienceleague.adobe.com/docs/experience-manager- cloud-service/moving/cloudmigration/pattern-detector/patterns-detected.html?lang=en

QUESTION 2

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 3

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-variableshttps://experienceleague.adobe.com/docs/experience-manager-cloud

service/implementing/deploying/environment-variables.html?lang=en

QUESTION 4

Which type of Cloud Manager tests are enabled for all Cloud Manager production pipelines and cannot be skipped?

A. Code Quality Testing

B. Experience Audit Testing

C. Ul Testing

D. Functional Testing

Correct Answer: A

Explanation: Code Quality Testing is a type of Cloud Manager tests that are enabled for all Cloud Manager production pipelines and cannot be skipped. Code Quality Testing checks the code quality of the project using SonarQube and reports

any issues or vulnerabilities. Code Quality Testing can fail the pipeline if the code quality does not meet the minimum standards defined by Adobe.

References: https://experienceleague.adobe.com/docs/experience-manager-cloud- service/implementing/testing/testing-overview.html?lang=en#testing- typeshttps://experienceleague.adobe.com/docs/experience-manager-cloud- service/

implementing/testing/code-quality-testing.html?lang=en

QUESTION 5

A custom component has one dialog field:

```
-> Title
-fieldLabel = Title
-sling:resourceType = granite/ui/components/coral/foundation/form/textfield
-name = ./title
```

The developer needs to implement a Sling Model to perform a business logic on the authored value. The developer writes the following HTL snippet.

```
<sly data-sly-use.display="com.adobe.aem.guides.certification.core.models.HelloWorldModelImpl">
<h1>${display.messageText}</h1>
</sly>
```

Which two implementations will support this HTL snippet? (Choose two.)

A.

```
@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class HelloWorldModelImpl {
@ScriptVariable
private String authoredVal;
private String messageText;

@PostConstruct
public void init() {
        if (StringUtils.isNotBlank(authoredVal)) {
                setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, authoredVal));
        }
}

public void setMessageText(String messageText) {
        this.messageText = messageText;
}

public String getMessageText() {
        return messageText;
}

}
```

B.

```
public void init() {
        if (StringUtils.isNotBlank(title)) {
                setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));
        }
}

public void setMessageText(String messageText) {
        this.messageText = messageText;
}

public String getMessageText() {
        return messageText;
}
```

```
}

@Model(adaptables = SlingHttpServletRequest.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class HelloWorldModelImpl {
@Inject
@Via("resource")
private String title;
private String messageText;
```

C.

```
@PostConstruct
public void init() {
        if (StringUtils.isNotBlank(title)) {
                setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));
        }
}

public void setMessageText(String messageText) {
        this.messageText = messageText;
}

public String getMessageText() {
```

```
}

@Model(adaptables = Resource.class, defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)
public class HelloWorldModelImpl {
@ValueMapValue
@Named("title")
private String authoredVal;
private String messageText;
```

D.

```
@PostConstruct
public void init() {
        if (StringUtils.isNotBlank(title)) {
                setMessageText(StringUtils.join("Welcome", StringUtils.SPACE, title));
        }
}

public void setMessageText(String messageText) {
        this.messageText = messageText;
}

public String getMessageText() {
        return messageText;
}
```

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: BD

Explanation: Option B and Option D are two implementations that will support the HTL snippet. Option B 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 B also uses the @Inject annotation with the name parameter set to "./text" to inject the value of the text property into the text field. Option D uses the @Model annotation with the defaultInjectionStrategy parameter set to OPTIONAL. This allows the Sling Model to use optional injection for all fields and avoid null pointer exceptions if a property is missing. Option D also uses the @Inject annotation without any parameters to inject the value of the text property into the text field, using the field name as the default property name. References:
https://sling.apache.org/documentation/bundles/models.htmlhttps://experienceleague.adob e.com/docs/experience-manager-htl/using-htl/htl-block-statements.html?lang=en#use

QUESTION 6

A snippet throws an exception at runtime:

@Model(adaptables = {Resource.class}) public class MyCustomModel {

(SSlingObject

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.htmlhttps://sling.apache.org/docum entation/bundles/models.html#optional-injection

QUESTION 7

An AEM application wants to set up multi-tenancy using Adobe-recommended best practices and bind multiple configurations to it. Which of the following options is recommended?

A. import org.apache.felix.scr.annotations.Component; @Component(label = "My configuration", metatype = true, factory= true)

B. import org.osgi.service.component.annotations.Component; @Component(service = ConfigurationFactory.class)

C. import org.osgi.service.metatype.annotations.AttributeDefinition; import org.osgi.service.metatype.annotations.ObjectClassDefinition; @ObjectClassDefinition(name = "My configuration")

D. @Component(service = ConfigurationFactory.class) @Designate(ocd = ConfigurationFactorylmpl.Config.class, factory=true)

Correct Answer: D

Explanation: The @Component(service = ConfigurationFactory.class) @Designate(ocd = ConfigurationFactorylmpl.Config.class,factory=true) option is recommended for creating a multi-tenancy configuration and binding multiple configurations to it. This option uses the OSGi R6 annotations to define a component that provides a service of type ConfigurationFactory and designates a class that contains the configuration properties. The factory=true attribute indicates that multiple configurations can be created for this component. References:https://experienceleague.adobe.com/docs/experience-manager- 65/deploying/configuring/osgi-configuration-settings.html?lang=en#creating-factory- configurations

QUESTION 8

Which attribute must be present in all filter rules in AEM dispatcher configuration?

A. /type

B. /selectors

C. /url

D. /glob

Correct Answer: A

https://experienceleague.adobe.com/docs/experience-manager-dispatcher/using/configuring/dispatcher-configuration.html?lang=en#configuring-access-to-content-filter

QUESTION 9

If multiple configurations for the same PID are applicable, which configuration is applied?

A. The last modified configuration is applied.

B. The configuration with the highest number of matching run modes is applied.

C. The one that occurs first in the repository is applied.

D. A configuration factory is created and all configurations are applied.

Correct Answer: B

When multiple configurations for the same PID are applicable, the configuration with the highest number of matching runmodes is applied. This is because the runmodes act as a filter to select the most specific configuration for a given environment. If there is a tie between two or more configurations with the same number of matching runmodes, the one that occurs first in the repository is applied. References:https://experienceleague.adobe.com/docs/experience-manager-65/deploying/configuring/configure-runmodes.html?lang=en#configuring-osgi-settings-per- runmode

QUESTION 10

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