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.pass4itsure.com/ad0-e134.html

100% Passing Guarantee
100% Money Back Assurance

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

QUESTION 1

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 2

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?

A. update

B. replace

C. merge

Correct Answer: C

QUESTION 3

SPA components are connected to AEM components via the MapTo() method.

Which code should be used to correctly connect an SPA component called ItemList to its AEM equivalent?

A. (‘project/components/content/itemList’).MapTo(ItemList,ItemListEditConfig);

B. MapTo(‘project/components/content/itemList’)(ItemList,ItemListEditConfig);

C. ItemList.MapTo(‘project/components/content/itemList’);

D. MapTo(ItemList)(‘project/components/content/itemList’,ItemListEditConfig);

Correct Answer: B

https://experienceleague.adobe.com/docs/experience-manager-learn/getting-started-with-aem-headless/spa-editor/react/map-components.html?lang=en

QUESTION 4

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 5

An AEM application has a Header and Footer authored on every page.

The customer asks for the following:

1.

A centralized Header and Footer

2.

The ability to create a variation for both the Header and Footer

3.

Change the Header and Footer for specific time periods

4.

The ability to restore a previous version for both the Header and Footer

What should the developer use to meet the requirements?

A. Custom component

B. Content fragment

C. Static template

D. Experience fragment

Correct Answer: D

Explanation: An experience fragment is a group of one or more components including content and layout that can be referenced within pages. Experience fragments allow authors to create variations for different channels and modify them for specific time periods. Experience fragments also support versioning and restoring previous versions. References:https://experienceleague.adobe.com/docs/experience-manager- 65/authoring/authoring/experience-fragments.html?lang=en