



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.passapply.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 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 2

Which option should be used to synchronize user data across publish servers in a publish farm?

- A. Sling Content Distribution
- B. Vault plugin
- C. CURL
- D. Replication Agents

Correct Answer: A

Explanation: Sling Content Distribution is a feature that allows synchronizing user data across publish servers in a publish farm. Sling Content Distribution uses agents to distribute content from one instance to another using a publish-publish

topology. Sling Content Distribution can be configured to synchronize user data under /home/users and /home/groups paths, as well as their ACLs.

References: <https://experienceleague.adobe.com/docs/experience-manager-65/administering/security/sync.html?lang=en>
<https://experienceleague.adobe.com/docs/experience-manager-65/administering/security/sync.html?lang=en>



learn/sites/deploying/sling-content-distribution.html?

lang=en

QUESTION 3

Which configuration/section should be used to resolve the domain name by dispatcher?

- A. Configuration in vhosts file
- B. Configuration in filters.any
- C. Configuration in httpd.conf
- D. Configuration in DNS

Correct Answer: D

Explanation: The configuration in DNS (Domain Name System) should be used to resolve the domain name by dispatcher. The DNS resolves the domain names to the IP address of the web server that hosts the dispatcher. The dispatcher then matches the incoming request URL with the cached files or the AEM publish instances.

References: <https://experienceleague.adobe.com/docs/experience-manager-dispatcher/using/configuring/dispatcher-domains.html?lang=en#client-requests>

QUESTION 4

Refer to the exhibit.

```
<div class="list">
<ul data-sly-
list="currentPage.listChildren" >< li > This is a sample HTML snippet < /li >< /ul >< /div >< divclass = "r
{currentPage.listChildren}">
<li>This is a sample HTML snippet</li>
</ul>
</div>
```

The current page has three children.

What is the final rendered html output for the code snippet?



```
<div class='list'>  
<ul>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```

```
A. <div class='repeat'>  
<ul>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```

```
<div class='list'>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```

```
B. <div class='repeat'>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```

```
<div class='list'>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>
```

```
C. </ul>  
</div>  
<div class='repeat'>  
<ul>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>  
<div class="list">  
<ul>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>  
<div class="repeat">  
<ul>
```

```
D. <li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
<ul>  
<li>This is a sample HTL snippet</li>  
</ul>  
</div>
```



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

Correct Answer: D

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

[AD0-E134 PDF Dumps](#)

[AD0-E134 VCE Dumps](#)

[AD0-E134 Study Guide](#)