

KillTest

質量更高 服務更好



練習題

<https://www.killtest.net>

一年免費更新服務

Exam : AD0-E116

**Title : Adobe Experience Manager
Developer Expert**

Version : DEMO

1.How should a developer configure the replication agent to flush the dispatcher cache for a newlyactivated page?

- A. Create a dispatcher flush agent in publish instance
- B. Create a reverse replication agent on the author instance
- C. Create a new replication agent and set transport URI to point to the dispatcher
- D. Set the serialization type property of the defaultagent to dispatcher flush

Answer: A

2.An AEM site experiences slower page loads. A developer needs to identify the slow running requests. How should a developer analyze the requests with long response times?

- A. Use proxy.jar with the following command `java -jar proxy.jar <host> <remoteport> <localport>` to debug the webserver and AEM server communication
- B. Use rlog.jar with the following command `$ java -jar ../opt/helpers/rlog.jar -n 10 request.log` to identify long running requests
- C. Download Heapdumps from Tools > Operations > Diagnosis and analyze the Heapdumps using the Memory Analyzer Tool
- D. Embed `/libs/foundation/components/timing` component in the Page Component and verify the page load time

Answer: B

3.A developer has acomponent located under the path `/apps`. This component has a Client Library which is directly loaded onto a page. The publish instance loads the page correctly. The dispatcher also loads the page but the Client Library is missing.

How should the developerresolve this issue, while taking security into consideration?

- A. Change the ACLs for the Client Library.
- B. Move the Client Library under `/apps/<project>library`.
- C. Add the property `allowProxy` with a boolean value `true`.
- D. Allow the path to the clientlibson the dispatcher.

Answer: C

4.A developer is creating a new OSGi bundle `com.custom.package.b` to expose new services. `com.custom.package.a` is already installed and active in the system and has the following package definition:

```
Export-Package: com.custom.package.a;version="2.0"
Import-Package: com.sample.package.a;version="[1,2)"
Classpath: .,com.sample.package.b-3.0.jar
```

The system console shows the following package availability:

```
com.sample.package.a;version="1.5"
com.sample.package.c;version="3.0"
```

Bundle `com.custom.package.b` to be installed has the following packagedefinition:

```
Export-Package: com.custom.package.b;version="1.0"
Import-Package: com.custom.package.a;version=[1,2) ", com.sample.package.b;
version="[3.0,3.0] ", com.sample.package.c;version=[2,3)
```

What will happen when the developer uploads the bundle com.custom.package.b into the system?

- A. The bundle will install but fail the activation due to unsatisfied dependencies com.sample.package.b and com.sample.package.c.
- B. The bundle will install but fail the activation due to unsatisfied dependency com.sample.package.c.
- C. The bundle will install and activate successfully.
- D. The bundle will install but fail the activation due to unsatisfied dependency com.sample.package.b.

Answer: A

5. An application contains an OSGi configuration that contains a password.

How should a developer prevent this sensitive information from being stored in plain text in JCR?

- A. 1. Use console at /system/console/crypto to encrypt the value
- 2. Either create an encrypted value for each AEM instance and use runmodes to apply the different values or make sure relevant instances share the same master key
- 3. When loading the value in the code, call CryptoSupport.unprotect(_) before using the value
- B. 1. Use console at /system/console/configMgr and tick the checkbox "encrypt" before saving a configuration
- 2. Use encrypted values work across all instances
- 3. When loading the value in the code, call CryptoSupport.unprotect(...) before using the value
- C. 1. Use console at /system/console/crypto to encrypt the value
- 2. Either create an encrypted value for each AEM instance and use runmodes to apply the different values or make sure relevant instances share the same master key
- 3. Sensitive information is automatically decrypted using the CryptoSupport OSGi service before the value is returned
- D. 1. Use console at /system/console/configMgr and tick the checkbox "encrypt" before saving a configuration
- 2. Either create an encrypted value for each AEM instance and use runmodes to apply the different values or make sure relevant instances share the same master key
- 3. Sensitive information is automatically decrypted using the CryptoSupport OSGi service before the value is returned

Answer: A