

AD0-E716 Exam Braindumps: Adobe Commerce Developer with Cloud Add-on & AD0-E716 Certification Training



Adobe

AD0-E716

Adobe Certified Expert - Adobe Commerce Developer with
Cloud Add-on

QUESTION & ANSWERS

<https://www.certsquestions.com/AD0-E716-pdf-dumps.html>

BTW, DOWNLOAD part of PassLeaderVCE AD0-E716 dumps from Cloud Storage: <https://drive.google.com/open?id=1qHNhObB56dNyFJaSzinzwnoJzVqtB2Ygo>

We respect different propensity of exam candidates, so there are totally three versions of AD0-E716 guide dumps for your reference. The PDF version of AD0-E716 practice materials helps you read content easier at your process of studying with clear arrangement and the PC Test Engine version of AD0-E716 real test allows you to take simulative exam. Besides, the APP version of our practice materials, you can learn anywhere at any time with AD0-E716 study guide by your electronic devices.

Adobe AD0-E716 Exam Syllabus Topics:

Topic	Details
Topic 1	• Manipulate EAV attributes and attribute sets programmatically • Demonstrate how to effectively use cache in Adobe Commerce
Topic 2	• Demonstrate the ability to import • export data from Adobe Commerce • Explain how the CRON scheduling system works

Topic 3	• Demonstrate the ability to create new APIs or extend existing APIs • Demonstrate the ability to manage Indexes and customize price output
Topic 4	• Demonstrate the ability to add and customize shipping methods • Demonstrate a working knowledge of cloud project files, permission, and structure
Topic 5	• Demonstrate knowledge of how routes work in Adobe Commerce • Describe how to use patches and recurring set ups to modify the database
Topic 6	• Demonstrate the ability to extend the database schema • Describe how to add and configure fields in store settings
Topic 7	• Identify how to access different types of logs • Demonstrate understanding of branching using CLI

>> Valid AD0-E716 Test Guide <<

Three Formats for AD0-E716 Practice Tests PassLeaderVCE Exam Prep Solutions

PassLeaderVCE provides the three most convenient formats to prepare for AD0-E716 exam dumps. It offers a desktop practice test, web based practice test and pdf file. Therefore, feel free to go through Adobe Commerce Developer with Cloud Add-on (AD0-E716) exam dumps. Each of the three formats is downloaded to all android devices. Therefore, there's no reason to download an additional application to access web-based or desktop-based practice tests.

Adobe Commerce Developer with Cloud Add-on Sample Questions (Q37-Q42):

NEW QUESTION # 37

An Adobe Commerce developer has created a module that adds a product attribute to all product types via a Data Patch. According to best practices, how would the developer ensure this product attribute is removed in the event that the module is uninstalled at a later date?

- A. Make the Data Patch implement `\Magento\Framework\Setup\Patch\PatchRevertableInterface` and implement the `revert` method to remove the product attribute.
- B. Add instructions to the module's README.md file instructing merchants and developers that they must manually remove this attribute if they want to uninstall the module.
- C. Add an Uninstall.php file extending `\Magento\Framework\Setup\UninstallInterface` to the module's Setup directory and implement the `uninstall` method.

Answer: A

Explanation:

According to the Develop data and schema patches guide for Magento 2 developers, data patches can also implement `PatchRevertableInterface` to provide rollback functionality for their changes. The `revert()` method contains the instructions to undo the data modifications made by the patch. To ensure that the product attribute is removed when the module is uninstalled, the developer should make the data patch implement `PatchRevertableInterface` and implement the `revert` method to remove the product attribute using `EavSetupFactory` or `AttributeRepositoryInterface`. Verified References:
<https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html>

NEW QUESTION # 38

An Adobe Commerce developer is being tasked with creating a new cron job to run a method that has already been written. What are the minimally required steps to accomplish this?

- A. Create a `crontab.xml` file and set a schedule for the new cron job.

- B. Create a crontab.xml file and a new system configuration in system.xml for the schedule.
- C. Create crontab.xml and cron_groups.xml files to assign the new job to a cron group.

Answer: A

Explanation:

According to the Configure and run cron guide for Magento 2 developers, the crontab.xml file is used to declare and configure cron jobs for a module. The file should specify the name, instance, method and schedule of the cron job. Therefore, creating a crontab.xml file and setting a schedule for the new cron job are the minimally required steps to accomplish this task. Verified

References:

<https://devdocs.magento.com/guides/v2.3/config-guide/cli/config-cli-subcommands-cron.html>

NEW QUESTION # 39

On an Adobe Commerce Cloud platform, in which order does the ECE-Tools package apply patches?

- A. 1. All required Magento patches included in the Cloud Patches for Commerce package.
2. Selected optional Magento patches included in the Quality Patches Tool.
3. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.
- B. 1. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.
2. All required Magento patches included in the Cloud Patches for Commerce package.
3. Selected optional Magento patches included in the Quality Patches Tool.
- C. 1. All required Magento patches included in the Cloud Patches for Commerce package.
2. Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.
3. Selected optional Magento patches included in the Quality Patches Tool.

Answer: A

Explanation:

The order in which the ECE-Tools package applies patches is as follows:

All required Magento patches included in the Cloud Patches for Commerce package.

Selected optional Magento patches included in the Quality Patches Tool.

Custom patches in the /m2-hotfixes directory in alphabetical order by patch name.

The ECE-Tools package is a set of scripts and tools designed to manage and deploy Adobe Commerce Cloud projects. The Cloud Patches for Commerce package is a dependency of ECE-Tools that provides a set of required patches for Magento core issues that affect Adobe Commerce Cloud functionality. The Quality Patches Tool is an optional tool that allows developers to apply individual patches for specific Magento issues without waiting for a full product release. The /m2-hotfixes directory is a directory where developers can place their own custom patches for their Adobe Commerce Cloud projects. Verified Reference: [Magento 2.4 DevDocs]

NEW QUESTION # 40

Which hashing algorithm will Adobe Commerce choose to hash customer passwords?

- A. It does not matter if the Sodium extension is installed or not, the Magento hashing default algorithm will be SHA256.
- B. If the Sodium extension is installed, SHA256 will be chosen, otherwise MD5 will be used as the Magento default hashing algorithm.
- C. If the Sodium extension is installed, Argon 2ID13 will be chosen, otherwise SHA256 will be used as the Magento default hashing algorithm.

Answer: C

Explanation:

If the Sodium extension is installed, Argon 2ID13 will be chosen as the Magento default hashing algorithm.

Otherwise, SHA256 will be used.

The Sodium extension is a PHP extension that provides cryptographic functions. Argon 2ID13 is a password hashing algorithm that is considered to be more secure than SHA256.

If the Sodium extension is installed, Magento will use Argon 2ID13 as the default hashing algorithm for customer passwords. If the Sodium extension is not installed, Magento will use SHA256 as the default hashing algorithm.

Adobe Commerce uses secure hashing algorithms for customer passwords. As of the more recent updates, Adobe Commerce defaults to using the Argon2ID13 hashing algorithm, provided that the Sodium PHP extension is available. Argon2ID is considered a

secure and modern hashing algorithm designed to protect against brute-force attacks.

If the Sodium extension is not available, Adobe Commerce falls back to using SHA256, which, while secure, is not as robust as Argon2ID13.

This functionality ensures that customer data is safeguarded with the highest level of security available based on the server configuration.

Additional Resources:

* Adobe Commerce Developer Guide: Hashing Algorithms

* PHP Documentation: Argon2 and Sodium

NEW QUESTION # 41

An Adobe Commerce developer is tasked to add a file field to a custom form in the administration panel, the field must accept only .PDF files with size less or equal than 2 MB. So far the developer has added the following code within the form component xml file, inside the fieldset node:

```
<field name="pdf_file" formElement="fileUploader">
  <formElements>
    <fileUploader>
      <settings>
        <uploaderConfig>
          <param xsi:type="string" name="url">myvendor_mymodule/customForm/uploadPdf</param>
        </uploaderConfig>
      </settings>
    </fileUploader>
  </formElements>
</field>
```

How would the developer implement the validations?

A)

Add the Validations Within the HyVendor\MyModule\Controller\Adminhtml\CustomEntity\UploadPdf Controller

```
public function execute()
{
    $file = $this->fileUploaderFactory->create($this->getRequest()->getPdfFile());
    if($file->getExtension() == 'pdf') {
        throw new InvalidFileException(__('The file must be PDF.'));
    }
    if($file->getSize() >= '2048000') {
        throw new InvalidFileException(__('The file size must be less or equal than 2MB'));
    }
    return $this->resultFactory->create(ResultFactory::TYPE_PAGE);
}
```

B)

Add a virtual type for MyVendor\MyModule\Model\customPdfuploader specifying the allowedExtensions and the maxFileSize for the constructor, within the module's di.xml:

```
<type name="MyVendor\MyModule\Model\CustomPdfuploader">
  <arguments>
    <argument name="allowedExtensions" xsi:type="string">pdf</argument>
    <argument name="maxFileSize" xsi:type="number">2048000</argument>
  </arguments>
</type>
```

C)

Add the following code inside the <settings> node:

```
<allowedExtensions>pdf</allowedExtensions>
<maxFileSize>2048000</maxFileSize>
```

- A. Option A
- B. Option C
- **C. Option B**

Answer: C

Explanation:

The developer can add a virtual type for MyVendor\MyModule\Model\customPdfuploader specifying the allowedExtensions and the maxFileSize for the constructor, within the module's di.xml. This way, the developer can reuse the existing file uploader class and customize it for the specific field without modifying the core code. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

• • • • •

AD0-E716 Latest Training: <https://www.passleadervce.com/Adobe-Commerce/reliable-AD0-E716-exam-learning-guide.html>

- DOWNLOAD the newest PassLeaderVCE AD0-E716 PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1qHNhObB56dNyFJaSzwnoJzVqtB2Ygo>