



Adobe AD0-E716 Practice Questions

Shared by Woodard on 19-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

On an Adobe Commerce Cloud platform, at what level is the variable `env: composer_auth` located in the Project Web Interface?

Options:

- A- In the Environment-specific variables.
- B- In the Integration variables.
- C- In the Project variables.

Answer:

C

Explanation:

The variable `env: composer_auth` is located in the Project variables section in the Project Web Interface. This variable is used to store the authentication credentials for Composer repositories that require access keys or tokens. The developer can set this variable at the project level to apply it to all environments, or override it at the environment level if needed. Verified Reference: [Magento 2.4 DevDocs]²

Question 2

Question Type: MultipleChoice

How would a developer enable the magnification of CSS files on an Adobe Commerce Cloud Staging environment?

Options:

- A- Locally from the command line
`bin/magento config:set --lock-config dev/css/minify_files 1`
Commit the `app/etc/config.php` file and redeploy.
- B- Update the stores > setting > configuration > Advanced > Developer > css configuration in the Admin Panel.
- C- SSH to the Adobe Commerce Staging environment. From the command line

```
ece-tools config:set dev/css/minify_files 1  
bin/magento setup:static-content:deploy
```

Answer:

A

Explanation:

The developer can enable the magnification of CSS files on an Adobe Commerce Cloud Staging environment by locally running the command `bin/magento config:set --lock-config dev/css/minify_files 1` from the command line. This will set the configuration value in the `app/etc/config.php` file and lock it from being changed in the Admin Panel. The developer then needs to commit the `app/etc/config.php` file and redeploy the environment. Verified Reference: [Magento 2.4 DevDocs]2

Question 3

Question Type: MultipleChoice

A developer is working on an Adobe Commerce Cloud project and wants to get connection data for the environment's deployed services. The developer has all of the necessary permissions to do this.

Which two options would the developer take to get the connection credentials? (Select Two.)

Options:

- A- Run the `magento-cloud relationships CLI Command`.
- B- Get the data from the Project Web Interface dedicated section.
- C- Execute `ece-tools env:config:show services Command`.
- D- Connect to server via SSH and read `$_ENV['services']` variable.

Answer:

A, D

Explanation:

Two options to get the connection credentials for the environment's deployed services are to run

the magento-cloud relationships CLI command and to connect to the server via SSH and read `$_ENV['services']` variable. The magento-cloud relationships CLI command displays information about the relationships between an environment and its services, such as database, cache, search, etc. The developer can use this command to get the connection data for each service in JSON format. Alternatively, the developer can connect to the server via SSH and read the `$_ENV['services']` variable, which contains the same information as the CLI command output. Verified Reference: [Magento 2.4 DevDocs]³

Question 4

Question Type: MultipleChoice

There is the task to create a custom product attribute that controls the display of a message below the product title on the cart page, in order to identify products that might be delivered late.

The new EAV attribute `is_delayed` has been created as a boolean and is working correctly in the admin panel and product page.

What would be the next implementation to allow the `is_delayed` EAV attribute to be used in the `.phtml` cart page such as `$block->getProduct()->getIsDelayed()`?

A)

Create a new file `etc/catalog_attributes.xml`:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Catalog:etc/catalog_attributes.xsd">
    <group name="quote_item">
        <attribute name="is_delayed" />
    </group>
</config>
```

B)

Create a new file `etc/extension_attributes.xml`:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:framework:Api/etc/extension_attributes.xsd">
    <extension_attributes for="Magento\Catalog\Api\Data\ProductRenderInterface">
        <attribute code="is_delayed" type="bool" />
    </extension_attributes>
</config>
```

Create a new file `etc/eav_attributes.xml`:

```
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Eav:etc/eav_attributes.xsd">
    <entity type="quote_item">
        <attribute code="is_delayed">
            <field code="is_visible" locked="true" />
        </attribute>
    </entity>
</config>
```

Options:

A- Option A

B- Option B

C- Option C

Answer:

A

Explanation:

To allow the is_delayed EAV attribute to be used in the .phtml cart page, the developer needs to create a new file called etc/catalog_attributes.xml. This file will contain the definition of the is_delayed attribute.

The following code shows how to create the etc/catalog_attributes.xml file:

XML

```
<?xml version='1.0'?>
```

```
<catalog_attributes>
```

```
<label>Is Delayed</label>
```

```
<note>This attribute indicates whether the product is delayed.</note>
```

```
<sort_order>10</sort_order>
```

```
<required>>false</required>
```

```
</catalog_attributes>
```

Once the etc/catalog_attributes.xml file has been created, the is_delayed attribute will be available in the .phtml cart page. The attribute can be accessed using the getIsDelayed() method of the Product class.

PHP

```
$product = $block->getProduct();
```

```
$isDelayed = $product->getIsDelayed();
```

The isDelayed variable will contain the value of the is_delayed attribute. If the value of the attribute is 1, then the product is delayed. If the value of the attribute is 0, then the product is not delayed.

Question 5

Question Type: MultipleChoice

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?

Options:

- A- Add an Uninstall.php file extending \Magento\Framework\Setup\UninstallInterface to the module's Setup directory and implement the uninstall method.
- 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- Make the Data Patch implement \Magento\Framework\setup\Patch\PatchRevertableInterface and implement the revert method to remove the product attribute.

Answer:

C

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 Reference: <https://devdocs.magento.com/guides/v2.3/extension-dev-guide/declarative-schema/data-patches.html>

Question 6

Question Type: MultipleChoice

During database migration in the Adobe Commerce Cloud integration environment, a developer experienced a disk space error causing the database import to fail.

How would the developer fix this issue?

Options:

- A- Increase the disk space of the database service.
- B- Add a new database node and enable split database.
- C- Change the database engine to PostgreSQL that has no disk space limit.

Answer:

A

Explanation:

The developer can fix this issue by increasing the disk space of the database service. The database service is one of the services that run on the Adobe Commerce Cloud platform and provide functionality for the application. The database service uses MySQL as the database engine and stores data for products, customers, orders, etc. The disk space of the database service determines how much data can be stored and processed by the database. If the disk space is insufficient, the database import can fail with a disk space error. The developer can increase the disk space of the database service by modifying the .magento/services.yaml file and redeploying the environment. Verified Reference: [Magento 2.4 DevDocs]

Question 7

Question Type: MultipleChoice

An Adobe Commerce developer added a new API method to search and retrieve a list of Posts for a custom Blog functionality. This is the content of the module's etc/webapi.xml file:

```
<?xml version="1.0" ?>
<routes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="urn:magento:module:Magento_Webapi:etc/webapi.xsd">
  <route url="/V1/myvendor-blog/post/search" method="GET">
    <service class="MyVendor\Blog\Api\PostRepositoryInterface" method="getList"/>
    <resources>
      <resource ref="MyVendor_Blog::Post_view"/>
    </resources>
  </route>
</routes>
```

The new code has been deployed to production and the merchant is using https://merchant.domain.com/swagger to review the new endpoint, but it is not visible in swagger.

What would be a reason for this?

Options:

- A- The webapi.xml file should be moved into the etc/webapi_rest/webapi.xml file.
- B- Since the new endpoint is not anonymous, the merchant needs to enter a valid integration token in swagger in order to see the new method.
- C- The greturn annotation is missing in the MyVendor\Blog\Api\PostRepositoryInterface class.

Answer:

B

Explanation:

The reason why the new endpoint is not visible in swagger is that since the new endpoint is not anonymous, the merchant needs to enter a valid integration token in swagger in order to see the new method. The webapi.xml file specifies that the resource for the new endpoint is MyVendor_Blog::post, which means that only authorized users with this permission can access it. To generate an integration token, the merchant needs to create an integration in the admin panel and activate it. Then they can copy the token and paste it in swagger's authorization field. Verified Reference: [Magento 2.4 DevDocs] [Magento Stack Exchange]

Question 8

Question Type: MultipleChoice

An Adobe Commerce developer has installed a module from a third-party vendor. This module fires a custom event named `third_party_event_after` and also defines an observer named `third_party_event_after_observer` that listens to that event. The developer wants to listen to this custom event in their own module but wants to execute their observer's logic after the `third_party_event_after_observer` observer has finished executing.

What would the developer do to ensure their observer runs after the observer defined by the third-party module?

Options:

- A- Ensure the third-party module is listed in the `<sequence>` node of the developer's module.xml file.
- B- Set the sort order of the new observer to be less than that of the third-party vendor's observer.
- C- This is not possible as observers listening to the same event may be invoked in any order.

Answer:

B

Explanation:

To ensure that the developer's observer runs after the observer defined by the third-party module, they need to set the sort order of the new observer to be less than that of the third-party vendor's observer.

The sort order is a number that is assigned to each observer. The lower the number, the earlier the observer will be executed.

For example, if the third-party vendor's observer has a sort order of 10, the developer's observer would need to have a sort order of 9 or lower.



To Get Premium Files for AD0-E716 Visit

<https://www.p2pexams.com/products/ad0-e716>

For More Free Questions Visit

<https://www.p2pexams.com/adobe/pdf/ad0-e716>

20%
DISCOUNT

P2P
exams