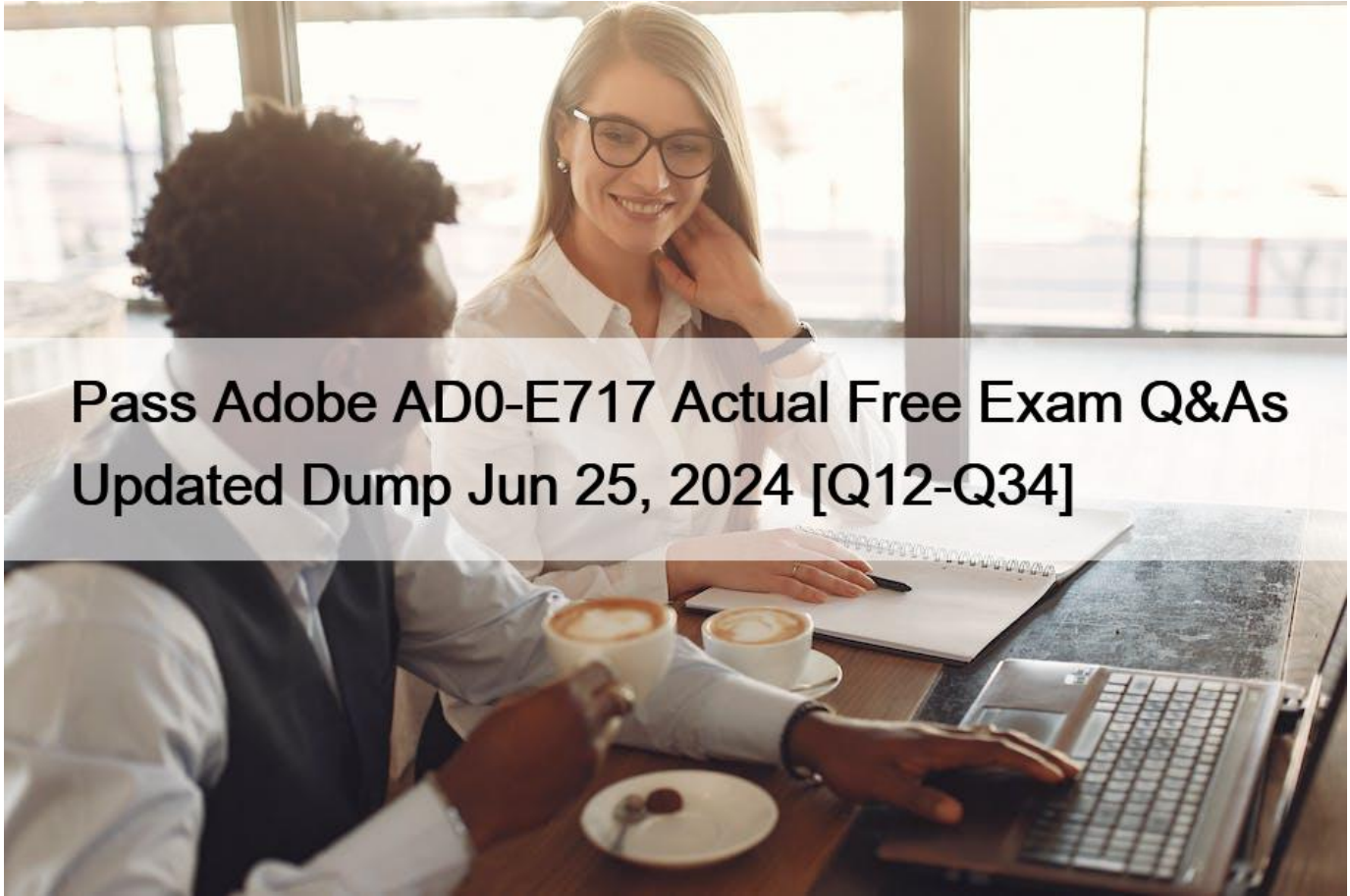


## Pass Adobe AD0-E717 Actual Free Exam Q&As Updated Dump Jun 25, 2024 [Q12-Q34]



## Pass Adobe AD0-E717 Actual Free Exam Q&As Updated Dump Jun 25, 2024 [Q12-Q34]

### Pass Adobe AD0-E717 Actual Free Exam Q&As Updated Dump Jun 25, 2024 Latest AD0-E717 Actual Free Exam Updated 79 Questions

#### Adobe AD0-E717 Exam Syllabus Topics:

TopicDetailsTopic 1- Identify when to call support \*Yaml files and limitations (DIY vs Support tickets)- Identify the basics of category management and products managementTopic 2- Describe different types of attributes- Describe how the ACL works with roles and resourcesTopic 3- Apply changes to existing product types and create new ones- Describe branching using the Adobe Commerce Cloud CLI toolTopic 4- Identify Adobe commerce Cloud Plan capabilities- Demonstrate the ability to manage Indexes and customize price outputTopic 5- Describe plugin, preference, event observers, and interceptors- Given a scenario, use a DB schema to alter a database table

#### QUESTION 12

A merchant wants to include taxes In an Adobe Commerce store. Which option can have a tax class assigned to it?

\* Order

- \* Shipping
- \* Category

According to the Adobe Commerce User Guide, a tax class can be assigned to either a product or a customer group in Adobe Commerce. A product tax class determines how a product is taxed, while a customer tax class determines how a customer is taxed based on their location and group membership. Shipping is considered as a product tax class in Adobe Commerce, and it can be assigned to different shipping methods or rates. The other options are not valid for assigning a tax class.

In Adobe Commerce, tax classes can be assigned to products and shipping. Categories, however, do not have tax classes assigned to them directly. Tax classes applied to shipping allow merchants to specify how taxes should be calculated for shipping costs, making option B the correct answer. Orders and categories do not have direct associations with tax classes in the same way products and shipping do.

### QUESTION 13

Which two attribute input types can be used for a date? (Choose two.)

- \* Date
- \* Timezone
- \* Date and Time
- \* Schedule

Explanation

The two attribute input types that can be used for a date are Date and Time and Date. These input types allow the user to select a date or a date and time from a calendar widget.

The Timezone and Schedule input types do not exist in Adobe Commerce.

Verified References: [Adobe Commerce User Guide &#8211; Create a product attribute]

### QUESTION 14

Which is a correct CMS content element in Adobe Commerce?

- \* Widget
- \* Sheet
- \* Image

A widget is a CMS content element that can be used to display dynamic content on a page. Widgets can be used to display things like product reviews, social media feeds, or even custom content.

In Adobe Commerce, widgets are a correct CMS content element. Widgets allow merchants to add dynamic data or content blocks to CMS pages, static blocks, and various other locations throughout the store's layout without needing to directly edit the site's code. Options B (Sheet) and C (Image) are not recognized CMS content elements in the context of Adobe Commerce's terminology, making option A the correct answer.

### QUESTION 15

How should a developer display a custom attribute on the category edit page in the admin panel when a new module Vendor.Category is created?

- \* Create view/adminhtml/layout/catalog\_category\_edit.xml in the module, and then define a block that would display the field for the attribute.
- \* The field for the attribute will appear automatically.
- \* Create view/adminhtml/ui\_component/category\_form.xml file in the module, and then define the field for the attribute.

## Explanation

To display a custom attribute on the category edit page in the admin panel, a developer should create a view/adminhtml/layout/catalog\_category\_edit.xml file in the module and define a block that would display the field for the attribute.

## QUESTION 16

On the Adobe Commerce Cloud Project Web Interface, what will be performed when clicking on the **Delete** button of an integration environment?

- \* The environment is marked as **inactive**, the git branch is still present but the database is deleted.
- \* The environment is completely deleted. Including git branch and database.
- \* The environment is marked as **inactive**, the git branch and the database are still present.

On the Adobe Commerce Cloud Project Web Interface, clicking on the **Delete** button of an integration environment will completely delete the environment, including the associated git branch and database. This action is irreversible and is used to remove an environment that is no longer needed. The environment, once deleted, frees up resources for the project and cannot be restored.

## QUESTION 17

A developer found a bug inside a private method of a third party module class. How can the developer override the method?

- \* Create a custom class with corrected logic, and define the class as preference in the preferences.xml.
- \* Create a custom class with the corrected logic, and define the class as a preference for original one in the di.xml.
- \* Create a plugin, implement correct logic in the **after** method, and then define the plugin in the di.xml.

## Explanation

To override a private method in a third party module class, a developer can create a plugin. The plugin should implement the **afterMethod**, and the correct logic should be placed in the **afterMethod**. The plugin should then be defined in the **di.xml** file.

## QUESTION 18

An Adobe Commerce Cloud developer wants to check the staging environment deployments history (i.e. branch, git, merge, sync). Where can the developer look up the history of the staging environment?

- \* Project Web Interface
- \* New Relic
- \* Adobe Commerce admin panel

The Project Web Interface is the main dashboard for managing Adobe Commerce Cloud projects. This includes the ability to check the staging environment deployments history.

The developer can look up the history of deployments to the staging environment, including branch, git merge, and sync operations, in the Project Web Interface. This interface provides a detailed log of all actions taken on the project, including deployments, enabling developers to track changes and troubleshoot issues that may arise.

## QUESTION 19

Which file on an Adobe Commerce Cloud project allows a developer to upgrade the PHP version and enable/disable a PHP extension?

- \* **magento.app.yaml**
- \* **.magento.env.yaml**
- \* **php.ini**

The `.magento.env.yaml` file is used on an Adobe Commerce Cloud project to customize the environment configuration, including the PHP version and enabling/disabling PHP extensions. This YAML configuration file provides the ability to manage service configurations and is essential for customizing the Cloud environment.

## QUESTION 20

Which characteristic is associated with a persistent cart?

- \* By default, a persistent cookie will become inactive in 30 days.
- \* While using the persistent cart, guest users do not need to log in or register to checkout
- \* While the customer is logged in, if the session cookie expires, the persistent cookie will remain active

A persistent cart is a cookie that is stored on the customer's computer. This cookie allows the customer to continue shopping even if they close their browser. If the customer is logged in, the persistent cookie will remain active even if the session cookie expires.

Associated with a persistent cart in Adobe Commerce is the characteristic that while the customer is logged in, if the session cookie expires, the persistent cookie will remain active. This ensures that the customer's shopping cart is preserved even if they have been inactive and the session has expired. The persistent cookie allows the cart to be restored when the customer returns to the store.

## QUESTION 21

A developer defined a new table in `db.schema.xml` while creating a new module.

What should be done to allow the removal of columns from the database when deleting them from `db.schema.xml`?

- \* The removable columns should be defined in `db_schema_whitelist.json`.
- \* The columns should have `removable` attribute set to `true` in the `db.schema.xml`.
- \* The removable columns should be defined in `db.schema_blacklist.json`.

Explanation

If a developer wants to allow the removal of columns from the database when deleting them from `db.schema.xml`, they need to define the removable columns in the `db_schema_whitelist.json` file. This file will tell Magento which columns can be removed from the database.

## QUESTION 22

Which property allows multiple cron jobs to share the same configuration?

- \* `name`
- \* `group`
- \* `schedule`

Explanation

The `group` property allows multiple cron jobs to share the same configuration. The `group` property defines the name of the cron group that the cron job belongs to. A cron group can have common settings such as `schedule`, `status`, and `error_email_recipients`.

[Configure a custom cron job and cron group \(tutorial\) | Adobe Commerce](#)

## QUESTION 23

How should a grid or form be included in an admin page layout using the UI Component?

- \* `<referenceContainer name="content"><uiComponent name="example_listing.xml"/>`

</referenceContainer>

\* <referenceContainername=&#8217;contentf> q <uiComponent name=&#8221;example\_listing7> </referenceContainer>

\* <referenceContainername=&#8217;content&#8221;>

<uiComponentname=&#8221;Vendor\_Module::ul\_component/example\_listing.xml7> </referenceContainer>

Explanation

To include a grid or form in an admin page layout using the UI Component, a developer should use the following code:

```
<referenceContainer name=&#8221;content&#8221;>
```

```
<uiComponent name=&#8221;Vendor_Module::ul_component/example_listing.xml&#8221; />
```

</referenceContainer>

## QUESTION 24

In which two directories are third-party modules located by default? (Choose two.)

- \* vendor/
- \* app/packages/
- \* app/modules/
- \* app/code/

Explanation

By default, third-party modules are located in vendor/ or app/code/ directories. The vendor/ directory contains modules that are installed using Composer, while the app/code/ directory contains modules that are manually copied or cloned from a repository.

## QUESTION 25

What does a URL Rewrite do?

- \* It updates the URL that is stored on the server.
- \* It changes the way a URL appears in the browser
- \* It updates the URL to a domain that is not being Indexed.

A URL Rewrite in Magento changes the way a URL appears in the browser. This is particularly useful for improving the readability and SEO of a URL. For example, a URL rewrite can be used to transform a long and complex URL into a shorter and more user-friendly version. It's important to note that while a URL rewrite changes the URL's appearance in the browser, it doesn't change the actual location of the resource on the server. This distinction is crucial for understanding how Magento handles URL rewrites and redirects, facilitating a more intuitive navigation structure within the store without altering the underlying server resources.

## QUESTION 26

What is one purpose of a customer data JS library?

- \* It stores the customers credit card info for usage in the checkout.
- \* It stores private customer data in local storage
- \* It stores the customer's username and password for easier frontend login.

Explanation

The customer data JS library is used to store private customer data in local storage. This data can be used to improve the

customer's experience on the store, such as by remembering their shipping address or their preferred payment method.

### QUESTION 27

What does a URL Rewrite do?

- \* It updates the URL that is stored on the server.
- \* It changes the way a URL appears in the browser
- \* It updates the URL to a domain that is not being Indexed.

Explanation

A URL rewrite changes the way a URL appears in the browser. For example, a URL rewrite could be used to change the URL for a product from `/product/1234to/product/my-product`.

### QUESTION 28

A seller would like to offer an electronic version of an album by selling each song individually. Which layout can be used to customize a product page layout for this item?

- \* `catalog_product_view_type_downloadable`
- \* `catalog_product_view_type_configurable`
- \* `catalog_product_view_category`

Explanation

The `catalog_product_view_type_downloadable` layout can be used to customize a product page layout for a downloadable product. This layout includes the product details, the product reviews, and the download links for the product's files.

### QUESTION 29

Which attribute option restricts Catalog EAV attributes to only certain product types?

- \* `show_in`
- \* `apply_to`
- \* `allowed_in`

Explanation

The `allowed_in` attribute restricts Catalog EAV attributes to only certain product types. This attribute can be used to prevent certain attributes from being displayed on certain product types.

### QUESTION 30

Under which section should the soft dependency for a module be listed in `app/code/<Vendor>/<Module>/composer.json` file?

- \* `suggest`: {
- \* }

- \* `optional`: {
- \* }

- \* `soft`: {
- \* }

Explanation

The soft dependency for a module should be listed in the `soft` section of

theapp/code/<Vendor>/<Module>/composer.jsonfile.

```
{  
  
    "name": "Vendor/Module",  
  
    "description": "This is a sample module",  
  
    "type": "magento2-module",  
  
    "version": "1.0.0",  
  
    "require": {  
  
        "php": "~7.3.0",  
  
        "magento/framework": "2.4.0",  
  
        "soft": {  
  
            "magento/module-catalog": "2.4.0";  
  
        }  
  
    }  
  
}
```

### QUESTION 31

A developer is creating a class VendorModuleModelMyModel. How should that class be defined as transient in di.xml?

- \* <type name="VendorModuleModelMyModel" transient="true">
- \* <type name="VendorModuleModelMyModel" singleton="false">
- \* <type name="VendorModuleModelMyModel" shared="false">

Explanation

To define a class as transient in di.xml, the transient attribute should be set to true.

Code snippet

```
<type name="VendorModuleModelMyModel" transient="true">
```

### QUESTION 32

What is the correct way to inject CMS block in a layout?

- \* <block class="Magento\Cms\Block\Block" name="block\_identifier"> <arguments> q  
<argument name="block\_id" xsi:type="string">my\_cms\_block\_identifier</argument> </arguments>  
</block>
- \* <block class="Magento\Cms\Block\Block" name="block\_identifier"> q

```
<actionmethod=&#8221;setBlock&#8217;>my_cms_block_identifier</action> </block>
```

```
* <referenceBlock name=&#8221;content&#8221;> <block class=&#8221;MagentoCmsBlockBlock&#8221;
name=&#8221;block_identifier&#8217; identifier=&#8221;my_cms_block_Identifier&#8221; /> </referenceBlock>
```

The correct way to inject a CMS block into a layout in Adobe Commerce is by using the <block> element with the class MagentoCmsBlockBlock and specifying the block identifier through an <argument> element with the name &#8220;block\_id&#8221;. This is shown in option A. The <block> tag defines the block class and name, and the <arguments> tag contains child <argument> tags for each argument, where the &#8220;block\_id&#8221; argument specifies the identifier of the CMS block to be injected.

### QUESTION 33

A developer would like to initialize a theme in Adobe Commerce. Which two files are required to complete this task? (Choose two.)

- \* theme.less
- \* registration.php
- \* composer.json
- \* themexml

Explanation

The two files that are required to initialize a theme in Adobe Commerce are registration.php and theme.xml.

The registration.php file registers the theme in the system and assigns a unique name to it. The theme.xml file declares the theme's title, parent theme, media directory, and preview image.

The theme.less file is not required to initialize a theme, but it can be used to define global styles for the theme.

The composer.json file is not required to initialize a theme, but it can be used to manage dependencies and packages for the theme.

Verified References: [Adobe Commerce Developer Guide &#8211; Create a theme]

### QUESTION 34

Which file is used to add a custom router class to the list of routers?

- \* routes.xml
- \* di.xml
- \* config.xml

Explanation

The routes.xml file is used to define the list of routers for Adobe Commerce. A custom router class can be added to the list of routers by adding a new entry to the routes.xml file.

**Online Questions - Valid Practice AD0-E717 Exam Dumps Test Questions:**

[https://www.dumpleader.com/AD0-E717\\_exam.html](https://www.dumpleader.com/AD0-E717_exam.html)