

# Lead2Passed



Lead2Passed

HOME

ALL VENDORS

★ GUARANTEE

? FAQ

TESTIMONIALS

Login / Register My Shopcart (1)

Input your exam code ...



## Try before you buy

Download a free sample of any of our exam questions and answers

- ✓ Online Test Engine: Online Tool, Convenient, easy to study. Instant Online Access. Supports All Web Browsers.
- ✓ PDF format: Easy to read and print learning materials, our products are available in PDF file format.
- ✓ Desktop Test Engine: Installable Software Application. Simulates Real Exam Environment. Practice Offline Anytime.



### Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.



### 365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.



### Money Back Guarantee

Full refund if you fail the corresponding exam in 60 days after purchasing. And Free get any another product.



### Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

<http://www.lead2passed.com>

Valid Certification Exam Dumps Materials and Study Guide -  
Lead2Passed

**Exam** : **AD0-E722**

**Title** : Adobe Commerce Architect  
Master

**Vendor** : Adobe

**Version** : DEMO

**NO.1** An Adobe Commerce Architect is supporting deployment and building tools for on-premises Adobe Commerce projects. The tool is executing build scripts on a centralized server and using an SSH connection to deploy to project servers.

A client reports that users cannot work with Admin Panel because the site breaks every time they change interface locale.

Considering maintainability, which solution should the Architect implement?

- A.** Modify project config.php file, configure 'admin\_locales\_for\_deploy' value, and specify all required locales
- B.** Edit project env.php file, configure 'adminLocales\_for\_build' value, and specify all required locales
- C.** Adjust the tools build script and specify required locales during \*setup:static-content:deploy' command

**Answer:** C

Explanation

The issue here is that the site breaks every time the users change interface locale in the Admin Panel. This is because the static content for the different locales is not generated during the deployment process. The solution is to adjust the tools build script and specify required locales during \*setup:static-content:deploy' command. This will ensure that the static content for all the needed locales is generated and deployed to the project servers. References:  
<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/static-content.htm>

**NO.2** An Adobe Commerce store owner sets up a custom customer attribute "my.attribute".

An Architect needs to display additional content on the home page, which should display only to Customers with "my.attribute" of a certain value and be the same content for all of them. The website is running Full Page Cache.

With simplicity in mind, which two steps should the Architect take to implement these requirements? (Choose two.)

- A.** Add a new context value of "my\_attribute" to Magento\Framework\App\Http\Context
- B.** Create a Customer Segment and use 'my.attribute' in the conditions
- C.** Add a custom block and a pHTML template with the content to the cmsjindexindex.xml layout
- D.** Add a dynamic block with the content to the Home Page
- E.** Use customer-data JS library to retrieve "my.attribute" value

**Answer:** A D

Explanation

To display additional content on the home page based on a custom customer attribute, the Architect needs to do the following steps:

Add a new context value of "my\_attribute" to Magento\Framework\App\Http\Context. This will allow the Full Page Cache to generate different versions of the page for customers with different values of

"my.attribute". The context value can be set using a plugin on the Magento\Customer\Model\Context class.

Add a dynamic block with the content to the Home Page. A dynamic block is a type of content block that can be configured to display only to specific customer segments or conditions. The Architect can use the 'my.attribute' in the conditions of the dynamic block and assign it to the Home Page in the Content > Blocks section of the Admin Panel. References:

Private content | Magento 2 Developer Documentation  
Dynamic Blocks | Adobe Commerce 2.3 User Guide - Magento

**NO.3** An Architect is investigating a merchant's Adobe Commerce production environment where all customer session data is randomly being lost. Customer session data has been configured to be persisted using Redis, as are all caches (except full page cache, which is handled via Varnish). After an initial review, the Architect is able to replicate the loss of customer session data by flushing the Magento cache storage, either via the Adobe Commerce Admin Panel or running `bin/magento cache:flush` on the command line. Refreshing all the caches in the Adobe Commerce Admin Panel or running `bin/magento cache:clean` on the command line does not cause session data to be lost. What should be the next step?

- A.** Check `app/etc/env.php` and make sure that the Redis configuration for caches and session data use different database numbers.
- B.** Educate the merchant to not flush cache storage and only refresh the caches in future.
- C.** Set the Stores > Configuration' option for Store Session Data Separately' to 'Yes' in the Adobe Commerce Admin Panel.

**Answer:** A

Explanation

The issue here is that the customer session data is randomly being lost when flushing the Magento cache storage. This is because the Redis configuration for caches and session data might be using the same database number, which causes the session data to be deleted along with the caches. The solution is to check the `app/etc/env.php` file and make sure that the Redis configuration for caches and session data use different database numbers. This will prevent the session data from being affected by the cache operations. References:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/develop/deploy/redis.html?lang=e>

**NO.4** A merchant is using a unified website that supports native Adobe Commerce B2B and B2C with a single store view.

The merchant's objective is to display the B2B account features, such as negotiable quotes and credit limits, in the header of the site on every page for logged-in users who belong to a B2B company account.

Each B2B company possesses its unique shared catalog and customer group, while numerous customer groups for non-B2B customers undergo changes. The merchant insists that this association should not be linked to customer groups.

Which two solutions should the Architect recommend for consideration, taking into account public data and caching? (Choose two.)

- A.** Create a Virtual Type that switches the theme when a user is part of a B2B company so the output can be modified accordingly in the alternate theme.
- B.** Create a new HTTP Context variable to allow for separate public content to be cached for users in B2B companies where the output can be modified accordingly.
- C.** Set whether the current user is part of a B2B company in the customer session and use that data directly to modify the output accordingly.
- D.** Create a new custom condition for customer segments that allow for choosing whether a user is part of a B2B company and then use this segment to modify the output accordingly.

**E.** Check if the current user is part of a B2B company within a block class and modify the output accordingly.

**Answer:** B D

Explanation

Option B is a valid solution because creating a new HTTP Context variable can allow for differentiating the public content cache for users who belong to a B2B company account. The HTTP Context variable can be used to modify the output of the header block accordingly, without affecting the performance or scalability of the site<sup>1</sup> Option D is also a valid solution because creating a new custom condition for customer segments can enable targeting users who are part of a B2B company account. The customer segment can be used to modify the output of the header block accordingly, using layout updates or dynamic blocks. This solution can also leverage the existing customer segment functionality and avoid custom coding<sup>2</sup> Option A is not a valid solution because switching the theme based on a virtual type can cause performance issues and increase the complexity of the site maintenance. Moreover, switching the theme can affect the entire site appearance, not just the header block<sup>3</sup> Option C is not a valid solution because using the customer session data directly to modify the output of the header block can prevent the public content cache from working properly. The customer session data is private and cannot be cached, so this solution can negatively impact the performance and scalability of the site<sup>4</sup> Option E is not a valid solution because checking if the current user is part of a B2B company within a block class can also prevent the public content cache from working properly. The block class logic is executed on every request, so this solution can negatively impact the performance and scalability of the site<sup>5</sup> References:

1:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/marketing/customer-segments.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/design/themes.html?lang=en> 4:

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

<https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/architecture/starter-architecture.htm>

**NO.5** An external system integrates functionality of a product catalog search using Adobe Commerce GraphQL API.

The Architect creates a new attribute `my_attribute` in the admin panel with frontend type `select`. Later, the Architect sees that `ProductInterface` already has the field `my_attribute`, but returns an `Int` value. The Architect wants this field to be a new type that contains both option id and label.

To meet this requirement, an Adobe Commerce Architect creates a new module and file `etc/schema.graphqls` that declares as follows:

```
interface ProductInterface {
  my_attribute: SelectableOption @resolver(class:"Vendor\\CatalogGraphQL\\Model\\Resolver\\SelectableOption")
}
```

After calling command `setup:upgrade`, the introspection of `ProductInterface` field `my_attribute` remains `Int`.

What prevented the value type of field `my_attribute` from changing?

**A.** The `Magento_CatalogGraphQL` module occurs later in sequence than the `Magento_GraphQL`

module and merging output of dynamic attributes schema reader overrides types declared in schema.graphqls

**B.** The fields of ProductInterface are checked during processing schema.graphqls files. If they have a corresponding attribute, then the backendjype of product attribute is set for field type.

**C.** The interface ProductInterface is already declared in Magento.CatalogGraphQL module. Extending requires use of the keyword extend before a new declaration of ProductInterface.

**Answer:** C

Explanation

According to the Adobe Commerce documentation, to extend an existing GraphQL interface, the keyword extend must be used before the interface name. This indicates that the new declaration is adding or modifying fields to the existing interface, rather than redefining it. If the keyword extend is omitted, the new declaration will be ignored and the original interface will be used. In this case, the Architect wants to change the type of the my\_attribute field in the ProductInterface interface, which is already declared in the Magento.CatalogGraphQL module. Therefore, the Architect should use the keyword extend before declaring the ProductInterface interface in the schema.graphqlsfile of the custom module. This will allow the Architect to override the type of the my\_attribute field from Int to MyAttributeType.

References:

Extend existing schema | Adobe Commerce Developer Guide

Schema language with GraphQL | Adobe Commerce