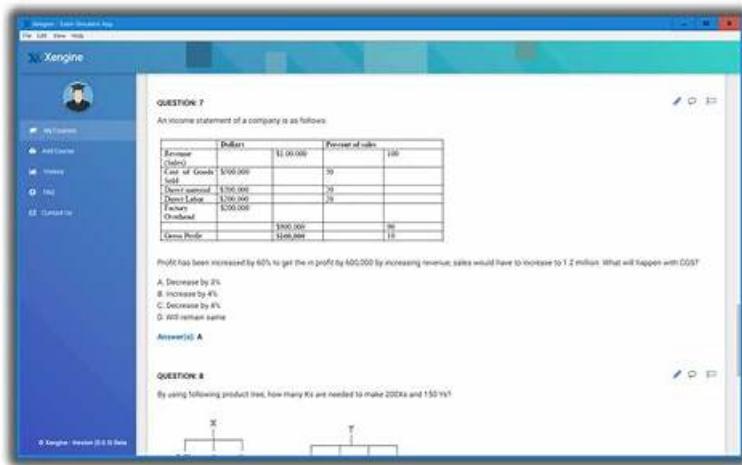


AD0-E725 Test Simulator - PDF AD0-E725 Cram Exam



What's more, part of that DumpsTests AD0-E725 dumps now are free: <https://drive.google.com/open?id=1gWgGkrWEZgK0IntmBpjU-g5-jkPxhkIJ>

Our company committed all versions of AD0-E725 practice materials attached with free update service. When AD0-E725 exam preparation has new updates, the customer services staff will send you the latest version. So we never stop the pace of offering the best services and AD0-E725 practice materials for you. Tens of thousands of candidates have fostered learning abilities by using our AD0-E725 Learning materials you can be one of them definitely.

Adobe AD0-E725 Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none">Section 1: Architecture: This section of the exam measures the skills of Backend Developers and covers the core structural concepts of Adobe Commerce. It involves demonstrating effective cache implementation, understanding key code components like plugins and observers, and managing multi-site configurations on a single instance. The domain also includes explaining the use of Git patches, critical security features, the CRON scheduling system, and how indexing functions within the platform.
Topic 2	<ul style="list-style-type: none">Cloud: This section of the exam measures the skills of Cloud Architects and covers the deployment and management of Adobe Commerce on cloud infrastructure. It requires explaining the fundamental cloud architecture, performing setup and configuration tasks, and utilizing the Adobe Commerce Cloud CLI tool to manage the environment effectively.
Topic 3	<ul style="list-style-type: none">Customizations: This section of the exam measures the skills of Solutions Engineers and involves modifying and extending platform functionality. This includes customizing core areas like the product catalog, checkout process, and admin panel, as well as manipulating data entities, customizing APIs, working with message queues, and writing integration tests to ensure code quality and functionality.
Topic 4	<ul style="list-style-type: none">External Integrations: This section of the exam measures the skills of Integration Specialists and focuses on connecting Adobe Commerce with external SaaS services. It encompasses the skills needed to customize data flows, utilize Adobe App Builder for extensibility, and implement Adobe I/O events and webhooks to create automated and connected business processes.

>> AD0-E725 Test Simulator <<

Well-Prepared AD0-E725 Test Simulator & Leading Offer in Qualification Exams & Updated Adobe Adobe Commerce Developer Expert

One of the most effective ways to prepare for the Adobe Commerce Developer Expert AD0-E725 exam is to take the latest Adobe AD0-E725 exam questions from DumpsTests. Many candidates get nervous because they don't know what will happen in the final Adobe Commerce Developer Expert AD0-E725 exam. Taking AD0-E725 exam dumps from DumpsTests helps eliminate exam anxiety. DumpsTests has designed this set of real Adobe AD0-E725 PDF Questions in accordance with the AD0-E725 exam syllabus and pattern. You can gain essential knowledge and clear all concepts related to the final exam by using these AD0-E725 practice test questions.

Adobe Commerce Developer Expert Sample Questions (Q30-Q35):

NEW QUESTION # 30

A customer asks for a new functionality which will allow them to see the total lifetime sales statistic of a product. This new data is to be made available for the API and traditional product exports. The technical architect of the company servicing the client decides that for this to be achieved, the first step is for an Extension Attribute to be created.

What are the next two steps that need to be taken? (Choose two.)

- A. Implement an after plugin on the Product Repository.
- B. **Declare the extension attribute in etc/webapi_rest/extension_attributes.xml.**
- C. Implement an around plugin on the Product Repository.
- D. **Declare the extension attribute in etc/extension_attributes.xml.**

Answer: B,D

NEW QUESTION # 31

An Adobe Commerce Developer is tasked to frequently send data to a third-party API. The API utilizes a JSON Web Token (JWT) that expires every hour. The developer decides to store the JWT in a custom cache.

Which step should the Developer take to implement this new custom cache type correctly?

- A. **Define the custom cache type in cache.xml with a unique name and instance.**
- B. Define the custom cache type in the di.xml, ensuring the cache model implements CacheInterface.
- C. Define the custom cache type directly in system.xml at the website level.

Answer: A

NEW QUESTION # 32

A Developer connects to the Magento-Cloud CLI and creates a new branch for environment using the following Git command:
git checkout -b <branch-name>

The developer notices the branch is not activated and does not automatically build when changes are pushed to the remote environment.

Which solution will fix this issue?

- A. magento-cloud environment:redeploy --no-wait should be executed after branch creation.
- B. git branch -b <branch-name> should be used instead of Git checkout.
- C. **magento-cloud environment:branch <branch-name> should be used instead of Git commands.**

Answer: C

Explanation:

In Adobe Commerce Cloud, simply creating a Git branch locally does not automatically create a corresponding Cloud environment. The correct way is to use:

magento-cloud environment:branch <branch-name>

This ensures the environment is created in Cloud, builds automatically, and is linked to the new Git branch.

B redeloys an existing environment but does not create one.

C is just another Git command and does not activate the environment.

Reference:

Adobe Commerce Cloud DevDocs - Manage environments

NEW QUESTION # 33

An Adobe Commerce Developer creates a before plugin for the save() method from the Magento\Framework\App\Cache\Proxy class to manipulate cache identifiers and data before it is saved to the cache storage.

An example of the class code is shown below:

```
namespace Magento\Framework\App\Cache;
use Magento\Framework\App\Cache\CacheInterface;
use Magento\Framework\ObjectManager\NoninterceptableInterface;
class Proxy implements
CacheInterface,
NoninterceptableInterface
{
...
public function save($data, $identifier, $tags = [], $lifeTime = null)
{
return $this->getCache()->save($data, $identifier, $tags, $lifeTime);
}
...
}
```

Why is the plugin not working as expected?

- A. An after plugin defined for the same function affects the results.
- B. An around plugin defined for the same function prevents the execution.
- C. The plugin cannot be created for this class.

Answer: C

Explanation:

Comprehensive and Detailed Explanation (with official references):

The correct answer is A. The plugin cannot be created for this class.

The reason is that Magento\Framework\App\Cache\Proxy implements the NoninterceptableInterface.

* Any class that implements the NoninterceptableInterface in Magento is excluded from the plugin system.

* This means no before, after, or around plugins can be applied to methods of such classes.

* Magento uses this mechanism to protect critical classes (like Proxy classes, Factories, and other infrastructure code) from being intercepted, as doing so could introduce performance or stability issues.

Therefore, the developer's plugin for the save() method does not work, because plugins are not allowed on this class by design.

Options B and C are incorrect because:

* Another plugin (after/around) does not block the execution in this case; the class itself is simply non- interceptable.

Official Documentation Extracts:

* "Plugins cannot be applied to final classes, final methods, non-public methods, or classes that implement

Magento\Framework\ObjectManager\NoninterceptableInterface."- Adobe Commerce DevDocs: Plugins limitations

* "Classes implementing NoninterceptableInterface cannot be intercepted. This interface is used to mark classes that must not be extended through the plugin mechanism"- Magento Framework Reference: NoninterceptableInterface

NEW QUESTION # 34

A Developer needs to subscribe to the customer_register_success event.

How should the observer be declared in the module?

- A. Declare in etc/events.xml:

```
<event name="customer_register_success">
<observer name="vendor_module_customer_register_observer" instance="Vendor\Module\Observer\CustomerRegisterSuccess" />
</event>
```
- B. Declare in etc/events.xml:

```
<observer name="customer_register_success">
<event name="vendor_module_customer_register_observer" instance="Vendor\Module\Observer\CustomerRegisterSuccess" />
</observer>
```
- C. Declare in etc/observer.xml:

```
<observer name="vendor_module_customer_register_observer">
<event name="customer_register_success" instance="
```

```
Vendor\Module\Observer\CustomerRegisterSuccess" />
</observer>
```

Answer: A

Explanation:

The correct declaration is in etc/events.xml with <event> as the parent node and <observer> as the child node.

Magento's event/observer framework requires the observer definition to follow this structure:

Option B is invalid because the `<observer>` tag cannot be the root node.

Option C is invalid because Magento does not use `observer.xml`; only `events.xml` is valid.

Reference:

Adobe Commerce DevDocs - Events and observers

NEW QUESTION # 35

• • • • •

Do you want to choose a lifetime of mediocrity or become better and pursue your dreams? I believe you will have your own pursuit. Perhaps you do not know how to go better our AD0-E725 learning engine will give you some help. The choice is like if a person is at a fork, and which way to go depends on his own decision. Our AD0-E725 Study Materials have successfully helped a lot of candidates achieve their certifications and become better. Our AD0-E725 learning guide will be your best choice.

PDF AD0-E725 Cram Exam: <https://www.dumpstests.com/AD0-E725-latest-test-dumps.html>

What's more, part of that DumpsTests AD0-E725 dumps now are free: <https://drive.google.com/open?id=1gWgGkrWEZgK0IntmBpjU-g5-jkPxhkIJ>

