Free Questions for AD0-E718

Shared by Bernard on 06-03-2023

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

An Architect wants to create an Integration Test that does the following:

* Adds a product using a data fixture

* Executes $this->someLogic->execute($product) on the product

* Checks if the result is true.

Sthis->someLogic has the correct object assigned in the setup () method-Product creation and the tested logic must be executed in the context of two different store views with IDs of 3 and 4, which have been created and are available for the test.

How should the Architect meet these requirements?

Options:

A- Create one test class with one test method. Use the \Magento\testFramework\ store\Executionstorecontext class once in the fixture and another time in the test.

B- Create two test Classes With one test method each. Use the @magentoExecuteInStoreContext 3 and @magentoExecuteInStoreContext 4 annotations on the class level.

C- Create one test class with two test methods. Use the @magentoStoreContext 3 annotation in one method and @magentoStoreContext 4 in the other one.

Answer:

C

Explanation:

The @magentoStoreContext annotation allows the test to run in the context of a specific store view. This annotation can be used on the method level to specify different store views for different test methods. This way, the product creation and the tested logic will be executed in the context of the same store view for each test method. Reference: https://devdocs.magento.com/guides/v2.4/test/integration/annotations/magento-store-context.html

Question 2

Due to a marketing campaign, a website is experiencing a very large number of simultaneously placed orders, which is affecting checkout performance. The website is in the production deploy mode.

Which two website settings can an Architect optimize to decrease the impact on checkout performance? (Choose two.)

Options:

A- Asynchronous indexing admin panel Setting (Stores > Settings > Ccr.f iguraticr. > Advanced > developer > Grid Settings > Asynchronous
indexing) can be enabled by executing the following CLI command: tm/magento config:set dev/grid/async_ini*xmg 1

B- Multithreaded checkout processing admin panel setting (stores > s ettmgs > Configuration > Sales > Checkout > General Settings > Asynchronous) can be set to a higher value representing the number of PHP threads used exclusively for checkout

C- Asynchronous email notifications admin panel Setting (stores > Settings > Configuration > Sales > Sales Emails > General Settings > Asynchronous) can be enabled

D- A new database can be created and the Split Database feature can be automatically configured with the following command: bin/Magento
setup:db-schema:split-sales-sales --host=''<checkout db host or ip>' ---dbname='<name>' ---username''<checkout db username>'' ---password='' '

E- The website deploy mode can be set to si-g- by executing the following CLI command: bin/magento deploy:mode:set siege. Provided that it will be changed back to production as soon as the number of simultaneously placed orders decreases to acceptable levels

Answer:

A, D

Explanation:

Asynchronous indexing allows Magento to process indexers in the background without affecting the storefront performance. Splitting the database allows Magento to use a separate database for checkout-related tables, which reduces the load on the main database and improves checkout performance. Reference:
https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#asynchronous-indexing https://devdocs.magento.com/guides/v2.4/config-guide/multi-master/multi-master.html

To decrease the impact on checkout performance, the Architect should do two things:

Enable asynchronous indexing admin panel setting. This will allow Magento to update indexes using cron jobs instead of doing it on the fly during checkout. This option can be found under Stores > Settings > Configuration > Advanced > Developer > Grid Settings > Asynchronous indexing. It can also be enabled by executing the following CLI command: bin/magento config:set dev/grid/async_indexing 1

Create a new database and use the Split Database feature. This will allow Magento to separate the sales data from the main database and reduce the load on the database server. This option can be configured with the following command: bin/magento setup:db-schema:split-sales --host='<checkout db host or ip>' --dbname='<name>' --username='<checkout db username>' --password=''

https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#m2devgde-asynchronous-indexing

https://devdocs.magento.com/guides/v2.4/config-guide/multi-master/multi-master.html

Question 3

Question Type: MultipleChoice

An Adobe Commerce Architect notices that the product price index takes too long to execute. The store is configured with multiple websites and dozens of customer groups.

Which two ways can the Architect shorten the full price index execution time? (Choose two.)

Options:

A- Enable price index customer group merging for products without tier prices
B- Set Customer Share Customer Accounts Option to Global
C- Edit customer groups to exclude websites that they are not using
D- Set MaGE_INDEXER_THREADS_COUNT environment variable to enable parallel mode
E- Move catalog price_index indexer to another custom indexer group

Answer:

A, D

Explanation:

The two best ways the Architect can shorten the full price index execution time are Option A. Enable price index customer group merging for products without tier prices, and Option D. Set MaGEINDEXERTHREADS_COUNT environment variable to enable parallel mode. Enabling customer group merging will help reduce the number of customer groups that need to be processed, while setting the environment variable will allow the indexer to use multiple threads and run in parallel mode, thus reducing the overall execution time.

Enabling price index customer group merging allows Magento to merge the price index rows for products that have the same price for all customer groups. This reduces the number of rows in the price index table and improves the performance of the indexer. Setting the MaGE_INDEXER_THREADS_COUNT environment variable allows Magento to run the indexer in parallel mode, which splits the index into multiple batches and processes them simultaneously. This reduces the execution time of the indexer. Reference: https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#customer-group-merging
https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html#parallel-mode

Question 4

Question Type: MultipleChoice

A client is migrating to Adobe Commerce Cloud and has approximately 800 existing redirects that must be implemented. The number of redirects cannot be reduced because all redirects are specific, and do not match any pattern.

How should the redirects be configured to ensure performance?

Options:
A- Use VCL snippets to offload the redirect to Fastly.
B- Add each redirect in the .magento/routes.yaml file.
C- Add each redirect as a URL rewrite via the admin Ul.

Answer:
A

Explanation:
The best option for configuring the redirects is to use VCL snippets to offload the redirects to Fastly. This is a Content Delivery Network (CDN) that can handle large numbers of requests

quickly and efficiently, ensuring that your redirects will be processed quickly and reliably. Furthermore, VCL snippets are easy to set up and can be reused for other redirects, making them an efficient and cost-effective solution for managing large numbers of redirects.

For Adobe Commerce on cloud infrastructure projects, configuring numerous non-regex redirects and rewrites in the routes.yaml file can cause performance issues. If your routes.yaml file is 32 KB or larger, offload your non-regex redirects and rewrites to Fastly.See Offload non-regex redirects to Fastly instead of Nginx (routes) in the Adobe Commerce Help Center1. Reference: https://experienceleague.adobe.com/docs/commerce-cloud-service/user-guide/configure/routes/redirects.html?lang=en21

Question 5

Question Type: MultipleChoice

Since the last production deployment, customers can not complete checkout. The error logs show the following message multiple times:

```
main.CRITICAL: Report ID: webapi-61b9fe83f0c3e; Message: Infinite loop detected, review the
trace for the looping path
```

The Architect finds a deployed feature that should limit delivery for some specific postcodes.

The Architect sees the following code deployed in/webapi_rest \di .xml and etc\frontend\di xml

```
<type name="Magento\Shipping\Model\Rate\Result">
    <plugin name="RestrictDeliveryMethods" type="Vendor\RestrictDeliveryMethods\Plugin\Shipping\LimitRates"/>
</type>
```

```
LimitRates.php:

public function __construct(
    \Magento\Checkout\Model\Session $session,
    ResultProvider $resultProvider
) {
    $this->session = $session;
    $this->resultProvider = $resultProvider;
}

public function afterGetAllRates(\Magento\Shipping\Model\Rate\Result $subject, array $result): array
{
    return $this->resultProvider->getLimitedRates($this->session->getQuote(), $result);
}
```

Which step should the Architect perform to solve the issue?

Options:

A- Inject an instance of \Magentro\Quote\API\CartRepostoryInterface and receive cart instance via$this->cartRepository->get($this-session->getQucteId())

B- Replace the injected dependency \Magento\Checkout\Model\Session\With\Magento\Framework\Session\SessionManagerInterface

C- Change 'after' plugin with 'around' plugin. The issue is being caused by calling the result provider code after the code of the original method.

Answer:

C

Explanation:

The 'after' plugin is not suitable for modifying the arguments or return value of the original method. The 'around' plugin allows the plugin method to wrap around the original method and modify its behavior and output. Reference: https://devdocs.magento.com/guides/v2.4/extension-dev-guide/plugins.html#around-methods

Question 6

Question Type: MultipleChoice

An Adobe Commerce Architect is reviewing api-functional test code. Some tests send errors to indicate that the customer address does not exist.

The test codes show the following:

```
/**
 * @magentoDataFixture Magento/Customer/_files/customer_one_address.php`
 * ...
 */
public function testMyUseCasTestForCartAddress(): void
```

Which steps should the Architect take to fix the test errors?

A)

Update the annotation to specify address_id @magentoDataFixture Magento/Customer/_files/customer_one_address.php with:
{"address_id":"$address.id$"}

B)

Change the annotation to use @magentoApiDataFixture Magento/Customer/_files/customer_one_address.php instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php

C)

Set the annotation to use @magentoPersistDataFixture Magento/Customer/_files/customer_one_address.php instead of @magentoDataFixture Magento/Customer/_files/customer_one_address.php

Options:

A- Option A

B- Option B

C- Option C

Answer:

B

Explanation:

The test errors are caused by using the wrong customer ID and address ID in the request. The correct customer ID and address ID should be obtained from the response of the previous request to create a customer and an address. The test code should use $this->customer->getId() and $this->address->getId() instead of hard-coded values. Reference: https://devdocs.magento.com/guides/v2.4/get-started/web-api-functional-testing.html

Question 7

Question Type: MultipleChoice

An Adobe Commerce Architect needs to scope a bespoke news section for a merchant's Adobe Commerce storefront. The merchant's SEO agency requests that the following URL structure:

news/{date}/{article_url_key}l where {date} is the publication date of the article, and {article_url_key} is the URL key of the article.

The Architect scopes that a news entity type will be created. The date and URL key data will be stored against each record and autogenerated on save. The values will be able to be manually overridden.

The Architect needs to manage routing this functionality and adhere to best practice.

Which two options should the Architect consider to meet these requirements? (Choose two.)

Options:

A- Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts.

B- Create an observer that listens to the controllers_front_send_response_before event, looks for the mm portion of the URL, and If it matches, loads the relevant news article by matching the

URL date and URL key parts.

C- Create a plugin that intercepts lu^jentoXFraBeworkUppXActien::executed, looks for the news portion of the URL and if it matches, loads the relevant news article by matching the URL date and URL key parts.

D- Create a standard controller route and mapping the internal URLs (such as news/article/view/id/1) to rewrites that are generated on save and then stored in the URL rewrites table.

E- Create a custom router that runs before the standard router and matches the news portion of the URL. then looks for and loads a news article by matching the date and URL key parts of the URL.

Answer:

A, E

Explanation:

To manage routing this functionality and adhere to best practice, you need to consider the following options:

Create a standard controller route and an Index/Index index controller class that loads the relevant news article by matching the URL date and URL key parts. This option will create a simple and straightforward way to handle the news requests using the standard Magento routing mechanism. The Index/Index controller class will receive the date and URL key parameters from the request and use them to load the news article model from the database.

Create a custom router that runs before the standard router and matches the news portion of the URL, then looks for and loads a news article by matching the date and URL key parts of the URL. This option will create a more flexible and customizable way to handle the news requests using a custom router class that implements \Magento\Framework\App\RouterInterface. The custom router class will check if the request path starts with news, then extract the date and URL key parts from the path and use them to load the news article model from the database.

: https://belvg.com/blog/how-to-create-custom-router-in-magento-2.html : https://devdocs.magento.com/guides/v2.4/extension-dev-guide/routing.html

Question 8

Question Type: MultipleChoice

An Adobe Commerce Architect is troubleshooting an issue on an Adobe Commerce Cloud project that is not yet live.

The developers migrate the Staging Database to Production in readiness to Go Live. However, when the developers test their Product Import feature, the new products do not appear on the frontend.

The developers suspect the Varnish Cache is not being cleared. Staging seems to work as expected. Production was working before the database migration.

What is the likely cause?

Options:

A- The Fastly credentials in the Production Database are incorrect.

B- A deployment should have been done on Production to initialize Fastly caching.

C- The site URLs in the Production Database are the URLs of the Staging Instance and must be updated.

Answer:

A

Explanation:

The Fastly credentials in the Production Database are incorrect. This means that the Varnish cache cannot be cleared by Commerce when new products are imported. The Fastly credentials should be updated to match the Production environment.See Configure Fastly credentials in the Adobe Commerce Help Center. Reference: https://experienceleague.adobe.com/docs/commerce-operations/configuration-guide/cache/use-varnish-cache.html?lang=en1https://support.magento.com/hc/en-us/articles/360006008192-Configure-Fastly-credentials

Question 9

Question Type: MultipleChoice

An Architect needs to review a custom product feed export module that a developer created for a merchant. During final testing before the solution is deployed, the product feed output is verified as correct. All unit and integration tests for code pass.

However, once the solution is deployed to production, the product price values in the feed are incorrect for several products. The products with incorrect data are all currently part of a content staging campaign where their prices have been reduced.

What did the developer do incorrectly that caused the feed output to be incorrect for products in the content staging campaign?

Options:

A- The developer forgot to use the getContentStagingValue() method to retrieve the active campaign value of the product data

B- The developer retrieved product data directly from the database using the entity_id column rather than a collection or repository.

C- The developer did not check for an active content staging campaign and emulates the campaign state when retrieving product data.

Answer:

C

Explanation:

Based on the given scenario, it is likely that option C - 'The developer did not check for an active content staging campaign and emulates the campaign state when retrieving product data' - is the correct answer. It appears that the developer did not take into account the active content staging campaign and did not properly adjust the product data when generating the product feed. As a result, the feed output is incorrect for products that are part of the staging campaign and have their prices reduced. The correct solution would be to check for an active content staging campaign and properly adjust the product data to reflect the campaign state.

Question 10

Question Type: MultipleChoice

A merchant notices that product price changes do not update on the storefront.

The index management page in the Adobe Commerce Admin Panel shows the following:

* All indexes are set to 'update by schedule'

* Their status is 'ready'

* There are no items in the backlog

* The indexes were last updated 1 minute ago

A developer verifies that updating and saving product prices adds the relevant product IDs into

the catalog_product_price_cl changelog table.

Which two steps should the Architect recommend to the developer to resolve this issue? (Choose two.)

Options:

A- Invalidate the catalog_product_price indexer in the Adobe Commerce Admin Panel so that it is fully reindexed next time the cron runs.

B- Manually reindex the catalog_product_price index from the Command line:bin\magentor indexer:reindex catalog_product_price.

C- Make sure that no custom or third-party modules modify the changelog and indexing process.

D- Make sure that the version_id for the price indexer in the mview_state table is not higher than the last entry for the same column in the changelog table and re-synchronize.

E- Reduce the frequency of the cron job to 5 minutes so the items have more time to process.

Answer:

C, D

Explanation:

To resolve the issue of product price changes not updating on the storefront, you need to take the following steps:

Make sure that no custom or third-party modules modify the changelog and indexing process. Some modules might interfere with the normal functioning of the indexing mechanism and cause data inconsistency or corruption. You can disable any custom or third-party modules that are related to indexing and check if the issue persists.

Make sure that the version_id for the price indexer in the mview_state table is not higher than the last entry for the same column in the changelog table and re-synchronize. The version_id column in the mview_state table indicates the current state of each indexer. If this value is higher than the last entry in the changelog table, it means that the indexer has skipped some records and needs to be re-synchronized. You can use the bin/magento indexer:reset command to reset the state of the price indexer and then run bin/magento indexer:reindex to reindex it.

: https://devdocs.magento.com/guides/v2.4/extension-dev-guide/indexing.html : https://devdocs.magento.com/guides/v2.4/config-guide/cli/config-cli-subcommands-index.html

To Get Premium Files for AD0-E718 Visit

https://www.p2pexams.com/products/ad0-e718

For More Free Questions Visit

https://www.p2pexams.com/adobe/pdf/ad0-e718