

DUMPS ARENA

Adobe Commerce Developer Expert

Adobe AD0-E709

Version Demo

Total Demo Questions: 10

Total Premium Questions: 50

Buy Premium PDF

<https://dumpsarena.co>

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

QUESTION NO: 1

An Adobe Commerce developer is tasked with adding a new export option for the order grid, they have added the following code for the export button within sales_order_grid.xml

```
<exportButton>
  <settings>
    <options>
      <option name="txt" xsi:type="array">
        <item name="value" xsi:type="string">txt</item>
        <item name="label" xsi:type="string" translate="true">TXT</item>
        <item name="url" xsi:type="string">vendor_module/sales/export/customExport</item>
      </option>
    </options>
  </settings>
</exportButton>
```

Upon testing, they are getting redirected, what would be a cause for this error?

- A. The Layout cache needs to be refreshed.
- B. The developer has to add a formkey for the new export option.
- C. The option's url attribute is not valid.

ANSWER: B**QUESTION NO: 2**

An Adobe Commerce developer is working on an Acme_Exceptions module which is supposed to overwrite logic inside some of Magento native exceptions such as \Magento\Framework\Exception\NoSuchEntityException or \Magento\Framework\GraphQL\Exception\GraphQLInputException, The module is open-source and will be available on packagist.org.

The build of the codebase of projects, including the module, will sometimes take place in docker containers with full access to filesystem. but then it is deployed to a read-only filesystem.

Which two approaches would the developer use to overwrite logic in those exceptions? (Choose two.)

- A. 1. Create a version of those exceptions inside the module using the original namespaces and classes, e.g. \Magento\Framework\Exception\NoSuchEntityException .
2. Use composer's extra > replace node to copy those exceptions to their original destinations like vendor/Magento/framework/exception/ to replace original files.
- B. 1. Create a version of those exceptions inside the module using new namespace, e.g. \Acme\Exceptions\Exception\NoSuchEntityException .

C. 1. Create a version of those exceptions inside the module using the original namespaces and Classes, e.g. `\Magento\Framework\Exception\NoSuchEntityException` .
2. Use composer's scripts > post-install-and and scripts > post-update-cmd nodes to copy those exceptions to their original destinations like `vendor/Magento/framework/exception/` to replace original files.

D. 1. Create a version of Those exceptions inside the module using the original namespaces and classes, e.g. `\Magento\Framework\Exception\NoSuchEntityException` .
2. Create a data patch copying those exceptions to their original destinations like `vendor/magento/framework/exception/` to replace original files.

ANSWER: A B

QUESTION NO: 3

An Adobe Commerce Developer is tasked with writing an importer for a custom entity. After the work is complete and deployed, they start receiving complaints from their client that the importer does not work and fails every time they use it.

The developer realizes that the client is importing a file which does not match the format required for the importer to process correctly.

What two features would the developer add to this importer to prevent this? (Choose two.)

- A.** Set up an example file and inject it into the `ExampleFileProvider` .
- B.** Set up a try/catch in the importer to display a warning the file is not in the correct format.
- C.** Set up validation by implementing a `validateRow` method in their importer.
- D.** Set `$needColumnCheck = true;` and add column names to the `$validColumnNames` class properties.

ANSWER: C D

QUESTION NO: 4

When paying by Bank Transfer, there is a requirement to send an email to customer service with payment details, after the order is placed successfully. Which two events can be used to send an email during the order placement process? (Choose two.)

- A.** `sales_order_payment_pay`
- B.** `sales_model_service_quote_submit_before`
- C.** `sales_model_service_quote_submit_success`
- D.** `sales_order_place_after`

ANSWER: A D

QUESTION NO: 5

Which two techniques can be used to protect a storefront POST action against Cross Site Request Forgery (CSRF) attacks? (Choose two.)

- A.** The form sending data to that action needs to include a `form_key` parameter with a valid form key and the action must instantiate `\Magento\Framework\Data\FormKey\Validator` to validate it.
- B.** The action needs to implement `\Magento\Framework\App\CsrfAwareActionInterface` and include the `validateForCsrf` method implementing the actual protection logic.
- C.** The form sending data to that action needs to include a `form_key` parameter with a valid form key which will automatically be used in a built-in CSRF validation.
- D.** The action needs to implement `\Magento\Framework\App\Action\HttpPostActionInterface` to trigger built-in CSRF validation.

ANSWER: B D**QUESTION NO: 6**

A Data Patch in a module being investigated implements `\Magento\Framework\Setup\Patch\PatchVersionInterface` and contains a `getVersion` method which returns the string `'2.4.2'`.

What is the implication of this when `php bin/magento setup:upgrade` is run?

- A.** The patch will only apply if the version specified in the `data_version` column of the `setup_module` table for our module is lower than `'2.4.2'`.
- B.** The patch will only apply if the version specified in the `data_version` column of the `setup_module` table for our module is higher than `'2.4.2'`.
- C.** The patch will only apply if the current version of the Magento Framework is `'2.4.2'`, and likely contains code that is incompatible with later versions.

ANSWER: A**QUESTION NO: 7**

An Adobe Commerce developer has created a process that exports a given order to some external accounting system. Launching this process using the Magento CLI with the command `php bin/magento "/.module:order:process --order_id-` is required.

Example: `php bin/magento ny_module:order:process --order_id=1245.`

What is the correct way to configure the command?

A)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    parent::configure();
}

protected function values()
{
    return [new InputValue('order_id', InputValue::REQUIRED, 'Order ID')];
}
```

B)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addOption('order_id', null, InputOption::VALUE_REQUIRED, 'Order ID');
    parent::configure();
}
```

C)

```
protected function configure()
{
    $this->setName('my_module:order:process');
    $this->setDescription('Processes an order');
    $this->addArgument('order_id', InputArgument::REQUIRED, 'Order ID');
    parent::configure();
}
```

A. Option

B. Option

C. Option

ANSWER: B**QUESTION NO: 8**

An Adobe Commerce developer wants to generate a list of products using ProductRepositoryInterface and search for products using a supplier_id filter for data that is stored in a standalone table (It. not in an EAV attribute).

Keeping maintainability in mind, how can the developer add the supplier ID to the search?

A. Add a custom filter to the virtual type

B. Magento\Catalog\models\Api\SearchCriteria\CollectionProcessor\ProductFilterProcessor for supplier_id field. In the custom filter, apply the needed join and filter to the passed \$collection.

C. Write a before plugin on `\Magento\Catalog\Model\ProductRepository::getList()` and register the search criteria passed. Write an event observer to listen for the event `catalog_product_collection_load_before` - Iterate through the registered search criteria, and if found, apply the needed join and filter to the event's `$collection` .

D. Write a before plugin on `\Magento\Framework\Api\SearchCriteria\CollectionProcessorInterface::process()` . Iterate through the `$searchCriteria` provided for `supplier_id` , and if found, apply the needed join and filter to the passed `$collection` .

ANSWER: B

QUESTION NO: 9

An Adobe Commerce developer is developing a class that is executed in both the frontend and crontab areas. When executed in the frontend `\Magento\Customer\Model\Session::getCustomerid()` is called but when executed in the crontab this section of code is skipped.

Keeping performance and best practices in mind, what would be the best approach to achieve this?

A. Inject the `\Magento\Customer\Model\Session\Proxy` class in the constructor of their class and call the `getCustomerid` function wherever required in the class.

B. Declare `\Magento\Customer\Model\Session` in the constructor dependency of their class and declare a proxy in the `di.xml`. Then call the `getCustomerid` function wherever required in the class.

C. Declare `\Magento\Customer\Model\Session` in the constructor of their class dependency and call the `getCustomerid()` function wherever required.

ANSWER: B

QUESTION NO: 10

An Adobe Commerce developer has just finished creating a new custom entity, a block that extends `\Magento\Framework\View\Element\AbstractBlock` that lists all of the existing entities and a controller with the appropriate handle to display the block.

The developer now wants to implement cache on the block so that when one of the custom entities is saved, the cache of the block is automatically invalidated.

According to best practices- what are the two steps to accomplish this? (Choose two.)

A. 1. Override `AbstractBlock::getCacheKeyInfo` and return an array containing the ids of all displayed entities.
2. Override `AbstractBlock::getCacheTags` and return an array containing, for all displayed entities, the value returned by the `getCacheTag` method of the model

B. 1. Create an around plugin on the `save()` method of the model of the entity.
2. Use the `cleanCacheByTags()` method of `\Magento\Framework\App\Cache\FlushCacheByTags` with a single argument containing the concatenation of a chosen key and the id of the entity.

C. 1. Implement `\Magento\Framework\DataObject\IdentityInterface` on the block that lists the entities.
2. Implement the `getIdentitiesO` method on the block to return an array containing, for all displayed entities, the value returned by the `getIdentities()` method of the model.

D. 1. Implement `\Magento\Framework\DataObject\IdentityInterface` on the model of the entity.
2. Implement the `getIdentities()` method to return the concatenation of a chosen key and the id of the entity.

ANSWER: A C