

## Topic 1 - Exam A

### Question #1

Topic 1

A company wants to build an Adobe Commerce website to sell their products to customers in their country. The taxes in their country are highly complex and require customization to Adobe Commerce. An Architect is trying to solve this problem by creating a custom tax calculator that will handle the calculation of taxes for all orders in Adobe Commerce.

Following best practices, how should the Architect add the taxes for all orders?

- A. Add a new observer to the event "sales\_quote\_collect\_totals\_before" and add the custom tax to the quote
- B. Write a before plugin to \Magento\Quote\Model\QuoteManagement::placeOrder() and add the custom tax to the quote
- C. Declare a new total collector in "etc/sales.xml" in a custom module

**Correct Answer:** C

### Question #2

Topic 1

An Adobe Commerce Architect is creating a new GraphQL API mutation to alter the process of adding configurable products to the cart. The mutation accepts configurable product ID. If the given product has only one variant, then the mutation should add this variant to the cart and return not nullable Cart type. If the configurable product has more variants, then the mutation should return not nullable ConfigurableProduct type.

The mutation declaration looks as follows:

```
type Mutation {
  addConfigurableToCart(product_id: Int!): AddToCartOutput!
  @resolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddConfigurableToCart")
}
```

How should the Adobe Commerce Architect declare output of this mutation?

- A. 

```
union AddToCartOutput
  @typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver")
  = ConfigurableProduct | Cart
```
- B. 

```
interface AddToCartOutput
  @typeResolver(class: "Vendor\\MyModule\\Model\\Resolver\\AddToCartOutputTypeResolver") {
  }
  type ConfigurableProduct implements AddToCartOutput {
  }
  type Cart implements AddToCartOutput {
  }
```
- C. 

```
type AddToCartOutput {
  product: ConfigurableProduct
  cart: Cart
}
```


**Correct Answer:** A

A third-party company needs to create an application that will integrate the Adobe Commerce system to get orders data for reporting. The integration needs access to the GET /V1/orders endpoint. It will call this endpoint automatically every hour around the clock. The merchant wants the ability to restrict or extend access to resources as well as to revoke the access using Admin Panel.

Which type of authentication available in Adobe Commerce should be used and implemented in a third-party system for this integration?

- A. Use token-based authentication to obtain the Admin Token. The third-party system will utilize the REST endpoint using the admin username and password to get the Admin Token, which will be used as the Bearer Token to authorize.
- B. Use token-based authentication to obtain an Integration Token. Integration will be created and activated in the admin panel using default integration token settings to get access to the token, which will be used as the Bearer Token to authorize.
- C. Use OAuth-based authentication to provide access to system resources. Integration will be registered by the merchant in the admin panel with an OAuth handshake during activation. The third-party system should follow OAuth protocol to authorize.

**Correct Answer:** B

  **abhilash198** 3 months, 3 weeks ago

Answer Should be C  
upvoted 1 times

In a custom module, an Architect wants to define a new xml configuration file. The module should be able to read all the xml configuration files declared in the system, merge them together, and use their values in PHP class.

Which two steps should the Architect make to meet this requirement? (Choose two.)

- A. Inject a "reader" dependency for "Magento\Framework\Config\Data" in di.xml
- B. Write a plugin for \Magento\Framework\Config\Data::get() and read the custom xml files
- C. Create a Data class that implements "\Magento\Framework\Config\Data"
- D. Append the custom xml file name in "Magento\Config\Model\Config\Structure\Reader" in di.xml
- E. Make a Reader class that implements "\Magento\Framework\Config\Reader\Filesystem"

**Correct Answer:** CE

  **abhilash198** 4 months ago

A and D are correct answers  
upvoted 1 times

An Adobe Commerce Architect creates a stopwords for the Italian locale named stopwordsJtJT.csv and changes the stopwords directory to the following:

```
<magento_root>/app/code/CustomVendor/Elasticsearch/etc/stopwords/
```

What is the correct approach to change the stopwords directory inside the custom module?

- A. Add stopwords to the stopwordsDirectory and CustomVendor\_Elasticsearch to the stopwordsModule parameter of the \Magento\Elasticsearch\SearchAdapter\Query\Preprocessor\Stopwords class via di.xml
- B. Add a new class implementing \Magento\Framework\Setup\Patch\PatchInterface to modify the default Value of elasticsearch\custom\stopwordspath in core\_config\_data table.
- C. Add stopwords to the stopwordsDirectory parameter of the \Magento\Elasticsearch\Model\Adapter\Document\DirectoryBuilder class via stopwords/it.xml and Adobe Commerce will automatically detect the current module.

**Correct Answer: A**

Currently there are no comments in this discussion, be the first to comment!

A client has multiple warehouses where orders can be fulfilled. The cost of shipping goods from each warehouse varies by day, due to the number of workers available. The Architect needs to make sure that when an order is shipped, it is shipped from the lowest cost warehouse that is open. How should this functionality be implemented?

- A. Create a new class as a preference for Magento\InventoryShipping\Plugin\Sales\Shipment\AssignSourceCodeToShipmentPlugin to set the lowest-cost warehouse on a shipment.
- B. Create a new class implementing Magento\InventorySourceSelectionApi\Model\SourceSelectionInterface, which returns open warehouses sorted by cost.
- C. Create an after plugin on Magento\InventoryDistanceBasedSourceSelection\Model\Algorithms\DistanceBasedAlgorithm to sort Warehouse sources by cost.

**Correct Answer: B**

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.

**Correct Answer:** *BE*

An Adobe Commerce Architect needs to customize the workflow of a monthly installments payment extension. The extension is from a partner who is contracted with the default website Payment Service Provider (PSP), which has its own legacy extension (a module using deprecated payment method).

The installment payment partner manages only initializing a payment, and then hands the capture to be executed by the PSP. Once the amount is successfully captured, the PSP notifies the website through a webhook. The goal of the webhook is only to create an "invoice" and save the "capture information" to be used later for refund requests through the PSP itself.

The Architect needs the most simple solution to capture the requested behavior.

Which solution should the Architect implement?

- A. Add a plugin before the `$invoice->capture()` and change its input to prevent the call of the `$payment->capture()`
- B. Change the `can_capture` attribute for the payment method under `config.xml` to be `<can_capture>0</can_capture>`
- C. Declare a capture command with type `Magento\Payment\Gateway\Command\NullCommand` for the payment method `CommandPool` in `di.xml`

**Correct Answer:** *C*

An existing Adobe Commerce website is moving to a headless implementation.

The existing website features an "All Brands" page, as well as individual pages for each brand. All brand-related pages are cached in Varnish using tags in the same manner as products and categories.

Two new GraphQL queries have been created to make this information available to the frontend for the new headless implementation:

```
type Query {
  brands {
    search: String @doc(description: "Search against brand names (partial matches)")
    pageSize: Int = 20 @doc(description: "Number of brand results to return")
    currentPage: Int = 1 @doc(description: "Page number to return")
  } : BrandsResult
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\Brands")
  brand {
    urlKey: String! @doc(description: "Match against brand url key")
  } : Brand
  @resolver(class: "ClientName\\ProductBrandsGraphQL\\Model\\Resolver\\BrandByUrlKey")
}
```

During testing, the queries sometimes return out-of-date information.

How should this problem be solved while maintaining performance?

- A. Specify a `$cache(cacheable: false)` directive for each GraphQL query, making sure that the data returned is not cached, and is up to date
- B. Specify a `@cache(cacheIdentity: Path\\To\\IdentityClass)` directive for each GraphQL query, corresponding to a class that adds cache tags for relevant brands and associated products
- C. Each GraphQL query's resolver class should inject `\Magento\GraphQLCache\Model\CacheableQuery` and call `setCacheValidity(true)` on it as part of the resolver's resolve function

**Correct Answer: B**

An Adobe Commerce Architect is investigating a case where some EAV product attributes are no longer updated.

The catalog is composed of 20,000 products with 100 attributes each.

The product updates are run by recurring Adobe commerce imports that happen multiple times a day.

The Architect finds an error in the logs that indicates an integrity constraint while trying to insert row with id 2147483647.

What is causing this error?

- A. Magento framework uses INSERT on DUPLICATE, which leads to reaching the max limit of the increment of the column.
- B. Integrity constraints were dropped after upgrading to the latest version, and the integrity checks were missed.
- C. EAV attribute import uses REPLACE, which leads to reaching the max limit of the increment of the column.

**Correct Answer: A**

An Adobe Commerce Architect is planning to create a new action that will add gift registry items to the customer's quote.

What should the Architect do to guarantee that private content blocks are updated?

- A. Mark the controller by setting no-cache HTTP headers
- B. Invalidate the status of gift registry indexers
- C. Specify a new action in a sections.xml configuration file

**Correct Answer: C**

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}`, 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.)

- A. 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.
- B. 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.
- C. Create a plugin that intercepts `Magento\Framework\App\Action::execute()`, 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 an `Index/Index` controller class that loads the relevant news article by matching the URL date and URL key parts.
- E. Create an observer that listens to the `controller_front_send_response_before` event, 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.

**Correct Answer:** AB

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 `backend_type` 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`.

**Correct Answer:** C

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

**Correct Answer:** AC

An Adobe Commerce Architect designs a data flow that contains a new product type with its own custom pricing logic to meet a merchant requirement.

Which three steps are required when adding a product type with custom pricing? (Choose three.)

- A. Content of the etc/product\_types.xml file
- B. Data patch to register the new product type
- C. Hydrator for attributes belonging to the new product type
- D. New price model extending \Magento\Catalog\Model\Product\Type\Price
- E. Custom type model extended from the abstract Product Type model
- F. A new class with custom pricing logic, extending the abstract Product model class

**Correct Answer:** ADE

An Adobe Commerce Architect needs to log the result of a ServiceClass::getData method execution after all plugins have executed. The method is public, and there are a few plugins declared for this method. Among those plugins are after and around types, and all have sortOrder specified.

Which solution should be used to meet this requirement?

- A. Declare a new plugin with the sortOrder value lower than the lowest declared plugin sortOrder and implement aroundGetData method.
- B. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement afterGetData method.
- C. Declare a new plugin with the sortOrder value higher than the highest declared plugin sortOrder and implement aroundGetData method.

**Correct Answer:** B

While developing a new functionality for a website in developer mode with all cache types enabled, an Adobe Commerce Developer needs to add `\Magento\Sales\Model\Service\InvoiceService $invoiceService` as a new dependency to an existing page action controller in `Vendor\CustomModule\Controller\Index\Index`. This is accomplished as follows:

```
[...]  
public function __construct(  
    \Magento\Framework\App\Action\Context $context,  
    \Magento\Sales\Model\Service\InvoiceService $invoiceService  
    \Magento\Framework\View\Result\PageFactory $resultPageFactory  
) {  
    [...]  
}
```

After cleaning the `full_page` cache and reloading the page, the developer encounters the following exception:

Recoverable Error: Argument 2 passed to `Vendor\CustomModule\Controller\Index\Index::__construct()` must be an instance of `\Magento\Sales\Model\Service\InvoiceService` [...]

Which action should the Architect recommend to the developer to fix this error?

- A. Clean the `block_html` cache along with `full_page` cache.
- B. Add the new `\Magento\Sales\Model\Service\InvoiceService $invoiceService` dependency at the end of the constructor signature.
- C. Remove the generated child class from `generated/code/Vendor/CustomModule/Controller/Index/Index`.

**Correct Answer: C**

A representative of a small business needs an Adobe Commerce Architect to design a custom integration of a third-party payment solution. They want to reduce the list of controls identified in their Self-Assessment Questionnaire as much as possible to achieve PCI compliance for their existing Magento application.

Which approach meets the business needs?

- A. Utilize the Advanced Encryption Standard (AES-256) algorithm to encrypt all customer-sensitive data from the payment module.
- B. Utilize the payment provider `iframe` system to isolate content of the embedded frame from the parent web page.
- C. Utilize a trusted signed certificate issued by a Certification Authority (CA) to secure each connection made by the payment solution protocol via HTTPS.

**Correct Answer: B**



A merchant asks for a new category attribute to allow uploading an additional mobile image against categories. The merchant utilizes the content staging and preview feature in Adobe Commerce and wants to schedule and review changes to this new mobile image field.

A developer creates the attribute via a data patch and adds it to view/adminhtml/ui\_component/category\_form.xml. The attribute appears against the category in the main form, but does not appear in the additional form when scheduled updates are made.

To change this attribute when scheduling new category updates, which additional action should the Architect ask the developer to take?

- A. The attribute must have its apply\_to field set to "staging" in the data patch file.
- B. The attribute must have <item name="allow\_staging" xsi:type="boolean">true</item> set in the category\_form.xml file under the attributes 'config' section.
- C. The attribute must also be added to view/adminhtml/ui\_component/catalogstaging\_category\_update\_form.xml.

**Correct Answer: C**

An Adobe Commerce Architect needs to create a new customer segment condition to enable admins to specify an 'Average sales amount' condition for certain segments.

The Architect develops the custom condition under Vendor\Module\Model\Segment\Condition\AverageSalesAmount with all of its requirements:

```
<?php
namespace Vendor\Module\Plugin;
use Magento\CustomerSegment\Model\Segment\Condition\Combine;
class AddNewChildOption
{
    public function afterGetNewChildSelectOptions(Combine $subject, array $result): array {
        $conditions = [
            [
                'value' => \Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount::class,
                'label' => __('Average sales amount'),
            ]
        ];
        return array_merge_recursive($result, $conditions);
    }
}
```

During testing, the following error appears:

```
"Class Magento\CustomerSegment\Model\Segment\Condition\Vendor\Module\Model\Segment\Condition\AverageSalesAmount does not exist at /var/www/vendor/magento/framework/Code/Reader/ClassReader.php"
```

What should the Architect do to fix the problem?

- A. Set the class to be \Vendor\Module\Model\Segment\Condition\AverageSalesAmount for the \$conditions value attribute
- B. Use a preference <preference for="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" type="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>
- C. Use a VirtualType <virtualType name="Magento\CustomerSegment\Model\Segment\Condition\AverageSalesAmount" types="Vendor\Module\Model\Segment\Condition\AverageSalesAmount"/>

**Correct Answer: A**

A single Adobe Commerce Cloud instance is set up with two websites (each with a single store view) with different domains.

- The default website is website\_one, with store view store\_one, and domain storeone.com.
- The second website is website\_two, with store view store\_two, and domain storetwo.com.

The magento-vars.php file is set up as follows to determine which website each request runs against:

```
<?php
function isHttpHost($host)
{
    if (!isset($_SERVER['HTTP_HOST'])) {
        return false;
    }
    return $_SERVER['HTTP_HOST'] === $host;
}

$_SERVER["MAGE_RUN_TYPE"] = "website";
if (isHttpHost("storetwo.com")) {
    $_SERVER["MAGE_RUN_CODE"] = "website_two";
} else {
    $_SERVER["MAGE_RUN_CODE"] = "website_one";
}
```

When testing a new GraphQL integration, all requests returned data relating to the default website, regardless of the domain.

What is causing this issue?

- A. The magento-vars.php file is not processed for any GraphQL requests, so the default website is always processed.
- B. \$\_SERVER["MAGE\_RUN\_CODE"] needs to be set to store and \$\_SERVER["MAGE\_RUN\_TYPE"] needs to be set to the store code instead.
- C. GraphQL requests are always run against the default store view unless a Store header or store cookie is provided.

**Correct Answer: C**

An Architect needs to integrate an Adobe Commerce store with a new Shipping Carrier. Cart data is sent to the Shipping Carrier's API to retrieve the price and display to the customer. After the feature is implemented on the store, the API hits its quota and returns the error "Too many requests". The Shipping Carrier warns the store about sending too many requests with the same content to the API.

In the carrier model, what should the Architect change to fix the problem?

- A. In \_doShipmentRequest(), call canCollectRates() before sending request to the API.
- B. Override getResponse(), save the response to a variable, check if the response exists, then return.
- C. Implement \_setCachedQuotes() and \_getCachedQuotes(), return the data if the request matches.

**Correct Answer: C**

An Architect working on a headless Adobe Commerce project creates a new customer attribute named `my_attribute`. Based on the attribute value of the customer, the results of GraphQL queries are modified using a plugin. The frontend application is communicating with Adobe Commerce through Varnish by Fastly, which is already caching the queries that will be modified. The Adobe Commerce Fastly extension is installed, and no other modifications are made to the application.

Which steps should the Architect take to make sure the `vcl_hash` function of Varnish also considers the newly created attribute?

- A. Create a new class inheriting from `Magento\GraphQLCache\Model\Cached\CachedFactorProviderInterface` and returning the value of `my_attribute` from the `getFactorValue` function and `my_attribute` from the `getFactorName` function. Then add this class through DI to the `idFactorProviders` array of `Magento\GraphQLCache\Model\Cached\CachedCalculator`.
- B. Create a new class inheriting from `Magento\Framework\GraphQL\Query\Resolver\IdentityInterface` and returning the value of `my_attribute` from the `getIdentities` function. Then specify a `@cache(cachedidentity: Path\To\IdentityClass)` directive for each GraphQL query to include the newly created `IdentityClass` to each query that adds the cache tags for each customer.
- C. Create a new class inheriting from `Magento\Customer\CustomerData\SectionSourceInterface` and returning the value of `my_attribute` from the `getSectionData` function. Then add this class through DI to the `sectionSourceMap` array of `Magento\Customer\CustomerData\SectionPoolInterface`.

**Correct Answer: A**

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.

`$this->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?

- A. Create two test classes with one test method each. Use the `@magentoExecuteInStoreContext 3` and `@magentoExecuteInStoreContext 4` annotations on the class level.
- B. Create one test class with two test methods. Use the `@magentoStoreContext 3` annotation in one method and `@magentoStoreContext 4` in the other one.
- C. Create one test class with one test method. Use the `\Magento\TestFramework\store\ExecuteInStoreContext` class once in the fixture and another time in the test.

**Correct Answer: C**

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.)

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

**Correct Answer:** AC