DUMPSARENA

Oracle CX Commerce 2022 Implementation Essentials

Oracle 1z0-1068-22

Version Demo

Total Demo Questions: 9

Total Premium Questions: 90

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Oracle CX Commerce Overview	3
Topic 2, Oracle CX Commerce Administration	11
Topic 3, Catalog Management	4
Topic 4, Search and Navigation	4
Topic 5, Pricing and Promotions	3
Topic 6, Storefront Design	23
Topic 7, Extensibility	31
Topic 8, Integrations	10
Topic 9, Mix Questions	1
Total	90

QUESTION NO: 1

Consider the following code from the gateway.json file for a sample payment gateway extension:

```
{
  "provider": "General Payment Gateway",
  "paymentMethodTypes": ["card","cash","physicalGiftCard"],
  "transactionTypes": {
   "card": ["authorization", "void", "refund"],
   "cash": ["initiate", "cancel"],
   "physicalGiftCard": ["balanceInquiry", "authorize", "void", "refund"],
  }
}
```

For which paymentMethodTypes is the processors property (not shown in the code example) required to be defined in your descriptors file?

A. card, for both regular and 3D-Secure cards

B. card, only for 3D-Secure card support

C. card, cash, physicalGiftCard

D. cash, physicalGiftCard

E. physicalGiftCard, only for debit card support

ANSWER: B

Explanation:

card, only for 3D-Secure card support is correct. In an Oracle CX Commerce payment gateway extension descriptor, the `processors` definition is specifically required when a card payment method supports 3D Secure. Three-D Secure introduces an additional cardholder-authentication step in the payment flow, and the gateway descriptor must identify the applicable processor configuration so Commerce can route and manage that processing correctly. The descriptor therefore declares the processor capability as part of enabling the gateway's 3D Secure card integration, rather than requiring it for every ordinary card transaction. This distinction lets a gateway extension expose standard card-payment behavior without declaring processor metadata unless the 3D Secure flow is implemented. The gateway configuration and descriptor provide Commerce with the information needed to recognize the payment-method capabilities supported by the extension and invoke the appropriate payment processing behavior. Oracle's CX Commerce developer documentation covers extension development and payment integration concepts in the CX Commerce Developer Guide. Oracle's payment-provider integration reference also describes payment gateway implementation metadata and processing requirements in the Payment Plug-in Implementation Guide.

QUESTION NO: 2

You want a contact with an associated storefront role of Administrator to be able to manage contact

registration requests for their account.

What two configuration steps are necessary to implement this functionality? (Choose two.)

A. Add the Contact Login with Self Registration element to support B2B Contact Registration Requests in the Header Widget instance

B. Ensure that generated emails for registration requests are enabled to allow the administrator to accept or reject the request directly from the email

C. Ensure that the Profile Navigation – Account Shoppers widget instance supports Registration Requests on My Account for storefront administrators

D. Configure the Contact Registration Requests widget settings to specify the storefront role of admin

ANSWER: C D

Explanation:

Managing B2B contact registration requests is exposed to qualified account users through the My Account navigation and the Contact Registration Requests widget. The **Profile Navigation – Account Shoppers** widget instance must support Registration Requests so that a storefront administrator has a navigation path to the registration-request management experience after signing in. This is the storefront integration that makes the capability available in the account profile area.

The **Contact Registration Requests** widget must also be configured for the Administrator storefront role. Its role-based setting determines which signed-in contacts can access and act on the requests associated with their account. Together, these settings both surface the Registration Requests area and authorize contacts assigned the Administrator storefront role to manage the requests. Oracle Commerce separates storefront navigation and role-controlled widget access, so configuring both is required for the intended administrator workflow. Oracle's Commerce documentation describes widget configuration and account-based storefront behavior in the Oracle CX Commerce Using Commerce documentation and provides the product documentation library at Oracle CX Commerce Documentation.

QUESTION NO: 3

A company currently has two sites. The ID of the default site is siteA. The ID of the second site is siteB. They

want to create a third site, but have it copy the configurations of siteB rather than the default siteA.

Which endpoint, method, and URI do they use to accomplish this?

A. The createSiteFromForm endpoint, using POST with the URI set to /ccadmin/v1/siteCreateForms/
siteB

B. The updateSite endpoint, using PUT with the URI set to /ccadmin/v1/sites/siteB

C. The createSite endpoint, using POST with the URI set to /ccadmin/v1/sites/siteB

D. The getSite endpoint, using GET with the URI set to /ccadmin/v1/sites/siteB

ANSWER: A

Explanation:

The createSiteFromForm endpoint, using POST with the URI set to /ccadmin/v1/siteCreateForms/
siteB is correct because Oracle CX Commerce provides the site creation form resource specifically to create a new site from an existing site's configuration. The site identifier appended to the endpoint identifies the source site whose configuration is used as the basis for the new site. Therefore, specifying `siteB` directs Commerce to use the second site, rather than automatically using the default site, as the source template for the new site. The request is a `POST` because it initiates creation of a new site resource and submits the required site-creation form data. This approach is useful when an implementation needs a new site to inherit an established site's settings and structure, while allowing the new site to receive its own identity and subsequent configuration changes. Oracle's REST API reference documents the Site Create Forms resource and its POST operation for this purpose. See the Oracle Commerce REST API: Create Site from Form and the Oracle Commerce REST API reference.

QUESTION NO: 4

A storefront developer has just logged into the Commerce Cloud Administration UI and selects the Design

page to begin configuring layouts and components for their account-based implementation.

What two key concepts must they understand? (Choose two.)

A. A B2B Storefront implementation can include B2B-specific widgets as well as standard widgets that
support both B2B and B2C implementations.

B. Out-of-the-box B2B-specific storefront layouts and associated components must be used for the accountbased
implementation and cannot be customized.

C. “Display layout to account shoppers only” must be selected in the layout settings in order to restrict the layout to account-based shoppers only.

D. A B2B storefront implementation can only support one site per Account.

ANSWER: A C

Explanation:

An account-based Oracle Commerce implementation uses the B2B storefront capabilities, which extend the standard storefront framework rather than replacing it. Therefore, a developer working on the Design page needs to understand that B2B-specific widgets can be used alongside standard widgets that are applicable to both B2B and B2C storefronts. This makes it possible to construct pages that combine account-oriented functionality, such as organization and buyer workflows, with general catalog, navigation, content, and merchandising experiences.

The developer must also understand how layouts are targeted to the intended shopper population. Selecting “Display layout to account shoppers only” makes a layout available only when the shopper is operating in an account context. This is important when a page’s content, components, or interactions are intended for account buyers and should not be presented in a standard consumer-shopping experience. Layout visibility is therefore a key part of configuring an account-based storefront correctly. Oracle’s B2B storefront documentation describes the use of B2B and standard widgets, as well as B2B layout configuration and account-shopper layout targeting. See Configure layouts for a B2B storefront and B2B widgets.

QUESTION NO: 5

A company uses multiple sites and wants a custom integration to retrieve catalog information for the correct site by using the Storefront REST APIs.

Which two statements describe site context for Storefront REST API requests? (Choose two.)

A. The X-CCSite request header can identify the site for the request

B. The default site is used when no site context is specified

C. The site must always be included as a query parameter named siteId

D. The site context is selected by the current Administration UI site selector

E. The site is derived from the shopper's selected language

ANSWER: A B

Explanation:

The `X-CCSite` request header can specify the site context for a Storefront REST API request. For example, a client can send `X-CCSite: siteB` to identify the site whose catalog, pricing, and other site-specific configuration should be used.

When the header is not supplied, Oracle Commerce uses the default site context. An integration that must explicitly work with a nondefault site should therefore include the appropriate site header rather than assuming that the endpoint path selects a site. See the Oracle Commerce REST API reference.

QUESTION NO: 6

A merchandiser wants to create a collection that is populated automatically whenever products satisfy defined rules, such as belonging to a brand and having a specified property value.

Which type of collection should be created?

A. Dynamic collection

B. Static collection

C. Navigation collection

D. Price group collection

ANSWER: A

Explanation:

A dynamic collection is correct because it uses rules to determine which products belong to the collection. As catalog data changes, products that begin or cease to satisfy the configured rules can be included or removed without the merchandiser manually maintaining the collection membership.

A static collection requires products to be added or removed explicitly. Dynamic collections are therefore appropriate for rule-driven assortments and automated merchandising. See the Oracle Commerce Administration guide.

QUESTION NO: 7

An account buyer submits an order that exceeds the approval threshold configured for the buyer's business account. A different account contact must review the submitted order and either approve or reject it.

Which storefront role should be assigned to the reviewing contact?

A. Approver

B. Buyer

C. Profile Manager

D. Account Address Book User

ANSWER: A

Explanation:

The reviewing contact should be assigned the Approver storefront role. This role is intended for account contacts who review orders submitted for approval and make the approval or rejection decision. The buyer can submit an order for approval, while the Approver performs the approval-workflow action for the account.

Oracle Commerce uses B2B storefront roles to control account-specific capabilities, including contact administration and approval workflows. See the Oracle Commerce B2B Commerce documentation.

QUESTION NO: 8

An organization has implemented loyalty points as a payment currency by creating a payment gateway

extension with loyaltyPoints as one of the paymentMethodTypes and configuring the Custom Curency

Payment webhook.

What additional step does the organization need to do to add loyalty points as a payment option on the

Payments Method widget?

A. Create a custom widget extension called Loyalty Payment and add it to the checkout layout

B. Create a new checkout layout with custom widgets for payment

C. Nothing. The Payments Method widget already includes loyalty points as a payment option

D. Add the baseline Loyalty Payment widget to the checkout layout on the Design page

ANSWER: A

Explanation:

Create a custom widget extension called Loyalty Payment and add it to the checkout layout is correct because configuring the payment gateway extension and the Custom Currency Payment webhook establishes the server-side processing required to validate and apply loyalty points. It does not, by itself, provide the storefront user interface through which a shopper selects loyalty points during checkout. Oracle CX Commerce uses a dedicated Loyalty Payment widget extension to render that payment choice and connect the shopper interaction to the loyalty-payment functionality. After the widget extension is created, it must be placed in the checkout layout so that it is available in the checkout experience alongside the standard payment-method flow. This separation allows the implementation to support a custom currency while retaining the standard checkout architecture and payment integration lifecycle. Oracle's loyalty-widget guidance describes using the Loyalty Payment widget as part of the storefront implementation for loyalty points, while its payment extensibility documentation explains the role of payment gateway extensions and payment webhooks. See Oracle CX Commerce: Use Loyalty Widgets and Oracle CX Commerce documentation.

QUESTION NO: 9

Which three statements describe webhooks in Commerce Cloud? (Choose three.)

A. Function webhooks support synchronous calls to external applications

B. Webhooks are only configured using the Server Extensions REST API

C. Event webhooks support asynchronous, event-triggered outbound calls to custom integration endpoints

D. Webhooks can be defined separately for Storefront, Preview, and Agent instances

E. Webhooks use the POST HTTP method for the outgoing call

ANSWER: A C E

Explanation:

Function webhooks support synchronous calls to external applications because they are invoked as part of Commerce processing when an extension needs an immediate response from an external service before processing can continue. This makes them appropriate for real-time extension scenarios, such as obtaining data or performing validation through an external integration.

Event webhooks support asynchronous, event-triggered outbound calls to custom integration endpoints. Commerce Cloud publishes the applicable event information to the configured endpoint after an event occurs, allowing external systems to react to Commerce activity without making the originating Commerce request wait for the integration to finish.

Webhooks use the POST HTTP method for the outgoing call. Commerce Cloud sends the webhook payload in the body of an HTTP POST request to the configured external URL. The receiving endpoint therefore needs to be available to accept POST requests and process the event or function payload supplied by Commerce Cloud. Oracle documents function and event webhooks as the two webhook mechanisms used to integrate Commerce with external applications and services. See Oracle's Understand Webhooks documentation and the Create a Webhook documentation.