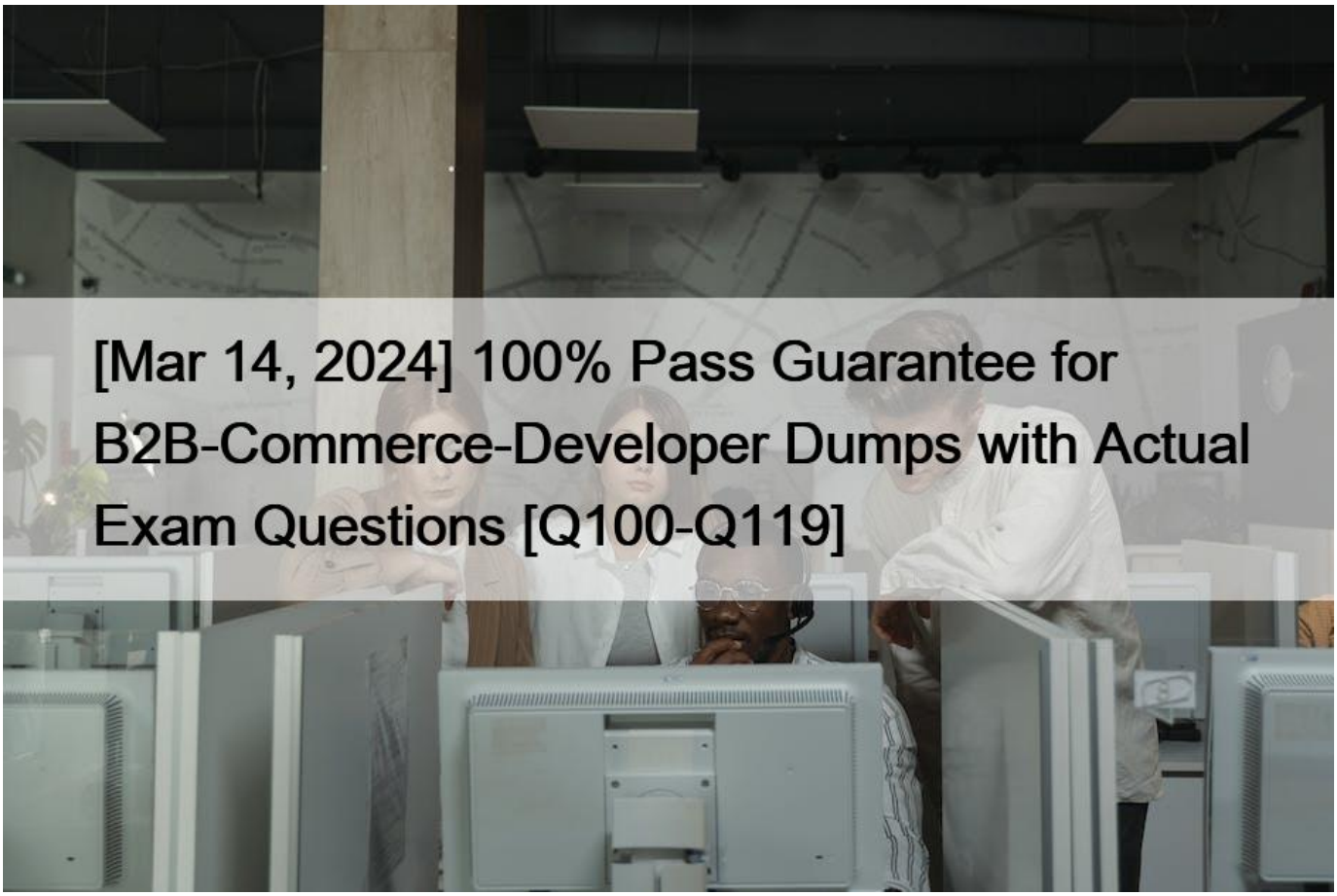
[Mar 14, 2024 100% Pass Guarantee for B2B-Commerce-Developer Dumps with Actual Exam Questions [Q100-Q119



[Mar 14, 2024 100% Pass Guarantee for B2B-Commerce-Developer Dumps with Actual Exam Questions Today Updated B2B-Commerce-Developer Exam Dumps Actual Questions

Salesforce B2B-Commerce-Developer Certification Exam is a valuable credential for developers who work in B2B commerce. It demonstrates to employers and clients that the developer has a deep understanding of the Salesforce platform and can develop solutions that meet the unique needs of B2B businesses. Additionally, certified developers have access to a range of resources and support from Salesforce, including exclusive events, training, and certification maintenance.

Preparing for the Salesforce B2B-Commerce-Developer certification exam requires a thorough understanding of the Salesforce B2B commerce architecture, data modeling, and integration with third-party systems. The candidate should have hands-on experience in developing, deploying, and maintaining B2B commerce solutions using the Salesforce platform. B2B-Commerce-Developer exam covers various topics such as data modeling, Apex programming, and integration with external systems.

NEW QUESTION 100

How should data for Lightning web components be provided?
* A few properties that contain sets (objects) of data
* One property that contains all data in one set (object)
* A single property object that contains sets (objects) of data
* Independent properties that take simpler, primitive values (e.g. String, Number, Boolean, Array)

Explanation

Data for Lightning web components should be provided as independent properties that take simpler, primitive values (e.g. String, Number, Boolean, Array). Providing data as independent properties allows the developer to expose data as public or private properties of the Lightning web component and communicate data between components or services. Providing data as simpler, primitive values allows the developer to use data types that are supported by JavaScript and Lightning web components and avoid unnecessary or complex conversions or transformations. Providing data as a few properties that contain sets (objects) of data is not a good way to provide data for Lightning web components, as it can create confusion or inconsistency in data structure and access. Providing data as one property that contains all data in one set (object) is not a good way either, as it can create complexity or inefficiency in data management and manipulation. Providing data as a single property object that contains sets (objects) of data is not a good way either, as it can create redundancy or duplication in data storage and retrieval. Salesforce References: Lightning Web Components Developer Guide: Communicate with Properties, Lightning Web Components Developer Guide: Data Types

NEW QUESTION 101

Which event is invoked by any CCRZ Salesforce B2B CommeceView after the view is rendered?
* view:*:load
* view:*:refresh
* view:*:onload
* view:*:rendered

NEW QUESTION 102

What is a valid way of referencing the global cc_api_CartExtentsion apex class via subscriber code?
* ccrz__cc_api_CartExtension
* c__cc_api_CartExtension
* cloudcraze.cc_api_CartExtension
* ccrz.cc_api_CartExtension

NEW QUESTION 103

In which three ways should useful debugging information in Salesforce B2B Commerce implementation be garnered? (3 answers)
A) Enabling the logging token via
* Admin andsubsequently inspecting the logs via the browser console.
* Logging a case with Salesforce support to enable advanced debugging options.
* Enabling debugging options for the current user and visually inspecting the Salesforce debug logs.
* Placing a System.debug() statement anywhere in the class being debugged.
* Logging into the community as a system administrator to identify any potential permissions or Visualforce exceptions.

Useful debugging information in Salesforce B2B Commerce implementation can be garnered in three ways:

Enabling the logging token via Admin and subsequently inspecting the logs via the browser console. This will enable logging messages and errors to the browser console, which can be viewed by opening the Developer Tools in the browser. The logging token can be enabled by setting the value of CO.logToken to true in CCAdmin.

Enabling debugging options for the current user and visually inspecting the Salesforce debug logs. This will enable logging messages and errors to the Salesforce debug logs, which can be viewed by opening the Debug Logs page in Salesforce Setup. The debugging options can be enabled by creating a Debug Level and a Trace Flag for the current user in Salesforce Setup.

Logging into the community as a system administrator to identify any potential permissions or Visualforce exceptions. This will allow viewing any errors or warnings that may occur on the community pages due to insufficient permissions or Visualforce issues. The system administrator can also access CCAdmin and other tools from within the community. Salesforce Reference: B2B Commerce and D2C Commerce Developer Guide, Logging, Debug Your Code

NEW QUESTION 104

Numerous flags when set, have a direct impact on the result set provided by the Global API’s. Which conversion flag allows for sObjects to be returned from the Global API’s when provided as a Boolean parameter with a value of true?
* ccrz.ccAPISizing.SKIPTRZ
* ccrz.ccAPISizing.SOBJECT
* ccrz.ccAPI.SZ_SKIPTRZ
* ccrz.ccAPI.SZ_SOBJECT

NEW QUESTION 105

Where is the API-based record creation generally handled in Salesforce B2B Commerce?
* In the methods available in extension hooks
* The service-layer responsible for the entity
* Data creation is not allowed
* Logic classes that implement the businesslogic for create operations

The API-based record creation is generally handled in the service-layer responsible for the entity in Salesforce B2B Commerce. The service-layer is a set of classes that provide methods for interacting with the data layer and performing business logic. Each entity, such as product, cart, or order, has a corresponding service class that handles the create, read, update, and delete operations for that entity. For example, ccrz.ccServiceProduct provides methods for creating and retrieving products. Salesforce Reference: B2B Commerce and D2C Commerce Developer Guide, Service Classes

NEW QUESTION 106

Based on error emails flowing in, a developer suspects that recent edits made to a checkout flow have created a defect. The developer has data points available to use as inputs in reproducing the scenario.

What should the developer do next?
* Open the flow, select Debug, provide the session ID for replay, and select Run.
* Open the flow, select Attach to Live Session, provide the session ID, and select Attach.
* Open the flow, select Debug, provide the Input values, and select Run.
* Open the flow, select Debug with Inputs, provide the Input values, and select Run.

Explanation

The next step that the developer should do after suspecting that recent edits made to a checkout flow have created a defect and having data points available to use as inputs in reproducing the scenario is to open the flow, select Debug, provide the Input values, and select Run. A flow is a type of application that automates a business process by collecting data and performing actions in Salesforce or an external system. A flow can be used to customize the checkout process in the storefront by defining the steps and logic that are executed when a customer places an order. A flow can be edited or modified using Flow Builder, a point-and-click tool that allows developers to create and manage flows. Flow Builder also provides debugging and testing tools that allow developers to run and troubleshoot flows before deploying them. To debug or test a flow, the developer can open the flow in Flow Builder, select

Debug from the toolbar, provide the Input values for the flow variables, and select Run. This will execute the flow in debug mode, which simulates how the flow runs in the org with real data. The developer can use debug mode to verify if the flow works as expected or if there are any errors or issues with the flow logic or actions. Open the flow, select Attach to Live Session, provide the session ID, and select Attach is not a valid next step, as it is not a feature or option available in Flow Builder or Salesforce CLI. Attach to Live Session is a feature that allows developers to attach a debugger to a running Apex session and inspect the state of the code execution. Open the flow, select Debug with Inputs, provide the Input values, and select Run is not a valid next step either, as it is not a feature or option available in Flow Builder or Salesforce CLI. Debug with Inputs is a feature that allows developers to debug an Apex class or trigger with predefined input values and breakpoints. Open the flow, select Debug, provide the session ID for replay, and select Run is not a valid next step either, as it is not a feature or option available in Flow Builder or Salesforce CLI. Replay is a feature that allows developers to replay an Apex log file and inspect the state of the code execution at each line. Salesforce References: [B2B Commerce Developer Guide: Customize Checkout Flows], [Salesforce Help: Flow Builder], [Salesforce Help: Debug Your Flows], [Salesforce Developer Blog: Apex Replay Debugger]

NEW QUESTION 107

Which wire adapter should a developer use to retrieve metadata about a specific picklist?
* getPicklistMetadataValues
* getPicklistMetadata
* getPicklistValues
* getPicklist
Explanation

To retrieve metadata about a specific picklist, a developer should use the getPicklistValues wire adapter. The getPicklistValues wire adapter imports data from the @salesforce/ui-api module and returns an object that contains information such as the picklist’s label, value, default value, validity, and controlling field values.

The getPicklistMetadataValues wire adapter does not exist. The getPicklistMetadata wire adapter does not exist either. The getPicklist wire adapter does not exist either. Salesforce References: [Lightning Web Components Developer Guide: getPicklistValues], [Lightning Web Components Developer Guide: Import User Interface API]

NEW QUESTION 108

What is a valid way of referencing the CC Cart Object whose API name is E_Cart__c in a SOQL query?
* _Cart__c
* c.E_Cart__c
* ccrz__E_Cart__c
* cloudcraze__E_Cart__c
Explanation

A valid way of referencing the CC Cart Object whose API name is E_Cart__c in a SOQL query is to use ccrz__E_Cart__c. This is the transformed name of the object that is used by the Salesforce B2B Commerce framework. All custom objects and fields that are part of the cloudcraze managed package have the prefix ccrz__ in their API names. For example, SELECT Id, Name FROM ccrz__E_Cart__c will query the CC Cart Object records. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Query Transformation

NEW QUESTION 109

Which code statement should a developer use to import the ID of the current Lightning Experience
* import id from ‘@salesforce/network/ld’
* import id from ‘@salesforce/experience/ld’

* import id from ‘@salesforce/site/ld’
* import id from ‘@salesforce/community/ld’

Explanation

To import the ID of the current Lightning Experience community, a developer should use the following code statement:

import id from ‘@salesforce/community/Id’;

The @salesforce/community module allows the developer to access information about the current community, such as its ID, name, URL, and base path. The other modules do not exist or are not related to the community ID. The @salesforce/network module is used to access information about the current network, such as its ID and name. The @salesforce/experience module is used to access information about the current user experience, such as whether it is standard or custom. The @salesforce/site module is used to access information about the current site, such as its name and prefix. Salesforce References: [Lightning Web Components Developer Guide: Import Community Information], [Lightning Web Components Developer Guide: Import Salesforce Modules]

NEW QUESTION 110

Numerous flags, when set, have a direct impact on the result set provided by the Global API’s. What is the default Global API DataSizing convention flag that is used by the API’s unless otherwise specified?

* CCRZ.ccPAI.SZ_XL
* CCRZ.ccPAI.SZ_M
* CCRZ.ccPAI.SZ_L
* CCRZ.ccPAI.SZ_S

NEW QUESTION 111

Which two statements are true regarding the cc_CallContext class in Salesforce B2B Commerce? (2 answers)

* The Salesforce session is accessible via the getSession method
* The class can be used internally within Salesforce B2B Commerce and in subscriber code to access context level parameters
* The userLocale variable returns the current Locale for storefront.
* The current storefront is accessible via this class

NEW QUESTION 112

A configuration value, CO.NewOrder, is set to TRUE. What is one way of

preventing an existing payment page from being shown on the checkout payment page?

* Delete the Visualforce page from the code base.
* Remove the value matching the page name from the pmt.whitelist configuration setting, then rebuild and activate a new Configuration cache
* Remove the payment type associated with the payment page from CO.pmts, then rebuild and activate a new cache.
* Override the front end template and modify the way the embedded payment page gets loaded from the payment list configuration.

NEW QUESTION 113

Which two Salesforce B2B Commerce visualforce pages must be enabled at a Salesforce Community level to make the out of the box SEO functionality available? (2 answers)

* CCSizeIndex
* SizeMap
* CCCatSiteMap

* ProductMap

Explanation

Two Salesforce B2B Commerce Visualforce pages that must be enabled at a Salesforce Community level to make the out of the box SEO functionality available are:

* CCSizeIndex: This page generates a sitemap.xml file, which is a file that lists all the pages and resources on a site that can be crawled by web crawlers. The page uses the configuration settings CO.SiteMapIncludeProducts and CO.SiteMapIncludeCategories to specify which products and categories should be included in the sitemap.

* CCCatSiteMap: This page generates a category sitemap file, which is a file that lists all the categories on a site that can be crawled by web crawlers. The page uses the configuration setting

* CO.SiteMapCategoryDepth to specify how many levels of subcategories should be included in the category sitemap. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Sitemap Files

NEW QUESTION 114

A Developer created a custom field that a project wants to expose on a given page.

How does the Developer ensure that the field is available to display on a given page?

* Override the Service Class that the page uses and update the Service Management in CCAdmin for the given storefront to use this new Service Class.
* Override the Logic Class that the page uses and update the Service Management in CCAdmin for the given storefront to use this new Service Class
* Create a new Service Class that the page uses and update the Service Management in CCAdmin for the given storefront to use this new Service Class
* Create a new Logic Class that the page uses and update the Service Management in CCAdmin for the given storefront to use this new Service Class

NEW QUESTION 115

Which handlebars helper function is used on Salesforce B2B Commerce pages and components for formatting price values?

* formatPrice
* priceAbs
* showprice
* price

Explanation

The handlebars helper function that is used on Salesforce B2B Commerce pages and components for formatting price values is formatPrice. This function will format a numeric value as a price according to the currency settings and locale of the storefront. For example, {{formatPrice price}} will format the price value as $1,234.56 for US dollars or ?1.234,56 for euros. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Handlebars Helpers

NEW QUESTION 116

What tool can a developer use to investigate errors during development?

* Commerce Diagnostics Event Logging
* Checkout Flow Log
* Support cases

* Browser dev tools

Explanation

Browser dev tools are a set of web authoring and debugging tools built into most modern browsers. They allow developers to inspect, edit, and debug the HTML, CSS, JavaScript, and network activity of a web page.

They can also provide useful information about errors, warnings, performance, and accessibility issues.

Browser dev tools are especially helpful for developing and testing Lightning web components, as they can display the component hierarchy, attributes, events, and slots.

The other options are not correct because:

* A. Commerce Diagnostics Event Logging is a feature that enables developers to capture and analyze events that occur during the execution of B2C Commerce code. It can help identify performance bottlenecks, memory leaks, and unexpected behavior. However, it is not a tool that can be used directly by the developer, but rather a service that requires a support request to enable and access.

* B. Checkout Flow Log is a log file that shows the details of the checkout flow execution, such as the input and output parameters, the pipeline steps, and the errors and warnings. It can help troubleshoot issues related to the checkout process, such as payment, shipping, or tax calculation. However, it is not a tool that can be used during development, but rather a log file that can be accessed after the checkout flow has run.

* C. Support cases are requests for assistance from the Salesforce support team. They can help resolve technical issues, provide guidance, or escalate bugs. However, they are not a tool that can be used to investigate errors during development, but rather a communication channel that can be used after the developer has exhausted other resources.

References:

* Browser Dev Tools

* Debug Your Lightning Web Components

* Commerce Diagnostics Event Logging

* [Checkout Flow Log]

NEW QUESTION 117

How can a developer establish communication between components that are not in the same DOM (Document Object Model) tree?

* Use publish-subscribe pattern.
* Configure targets property.
* Use dispatch events.
* Use @api decorators.

Explanation

To establish communication between components that are not in the same DOM (Document Object Model) tree, a developer can use the publish-subscribe pattern. The publish-subscribe pattern is a messaging pattern that allows components to communicate with each other without being directly connected or aware of each other. The components can publish events to a common channel and subscribe to events from that channel.

The channel acts as a mediator that delivers the events to the subscribers. The developer can use a custom library or a Salesforce platform service, such as Lightning Message Service or Platform Events, to implement the publish-subscribe pattern. Configuring the targets property is not a way to communicate between components that are not in the same DOM tree, as it only defines where a component can be used in an app.

Using dispatch events is not a way either, as it only works for components that are in the same DOM tree or have a parent-child relationship. Using @api decorators is not a way either, as it only exposes public properties or methods of a component to other components that use it. Salesforce References: Lightning Web Components Developer Guide: Communicate Across Salesforce UI Technologies, Lightning Web Components Developer Guide: Communicate with Events, [Lightning Web Components Developer Guide:

Communicate with Properties]

NEW QUESTION 118

Numerous flags, when set, have a direct impact on the result set provided by the Global API’s. What is the default Global API DataSizing convention flag that is used by theAPI’s unless otherwise specified?
* CCRZ.ccPAI.SZ_XL
* CCRZ.ccPAI.SZ_M
* CCRZ.ccPAI.SZ_L
* CCRZ.ccPAI.SZ_S

Explanation

The default Global API Data-Sizing convention flag that is used by the API’s unless otherwise specified is CCRZ.ccAPI.SZ_M. This flag indicates that the medium sizing block should be used for retrieving data, which includes the most commonly used fields and related entities. For example, ccrz.ccServiceProduct.getProducts() will use the SZ_M sizing block by default, unlessanother sizing block is specified as a parameter. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Data Sizing Conventions

NEW QUESTION 119

A developer is working in Visual Studio Code on a previously deployed project which is rather large and deployments are time consuming. The developer wants to know if a CSS file containing small changes was actually deployed to the org. What is one way this can be accomplished?
* Right-click the CSS file and choose Diff File Against Org
* Click the Tools menu and select Diff Styles Against Org…
* Right-click the folder for the component and choose Diff Styles Against Org
* Right-click the folder for the component and choose Diff Files Against Org

Explanation

To know if a CSS file containing small changes was actually deployed to the org, one way that a developer can accomplish this is by right-clicking the CSS file and choosing Diff File Against Org. Diff File Against Orgis an option that allows the developer to compare a local file with its remote version in the org using Salesforce CLI commands. The developer can use Visual Studio Code to execute these commands by right-clicking on files or folders in the project and choosing from various diff options. Right-clicking the CSS file and choosing Diff File Against Org allows the developer to see the differences between the local CSS file and the remote CSS file in the org side by side in Visual Studio Code. This way, the developer can verify if their changes were deployed successfully or not. Clicking the Tools menu and selecting Diff Styles Against Org… is not a valid way to know if a CSS file was deployed to the org, as there is no such option in Visual Studio Code or Salesforce CLI. Right-clicking the folder for the component and choosing Diff Styles Against Org is not a valid way either, as there is no such option in Visual Studio Code or Salesforce CLI.

Right-clicking the folder for the component and choosing Diff Files Against Org is not a valid way either, as it will compare all the files in the folder, not just the CSS file, which may not be efficient or necessary.

Salesforce References: [Salesforce CLI Command Reference: force:source:diff], [Salesforce Developer Tools for Visual Studio Code]

B2B-Commerce-Developer exam dumps with real Salesforce questions and answers:
https://www.examslabs.com/Salesforce/Salesforce-Developer/best-B2B-Commerce-Developer-exam-dumps.html]