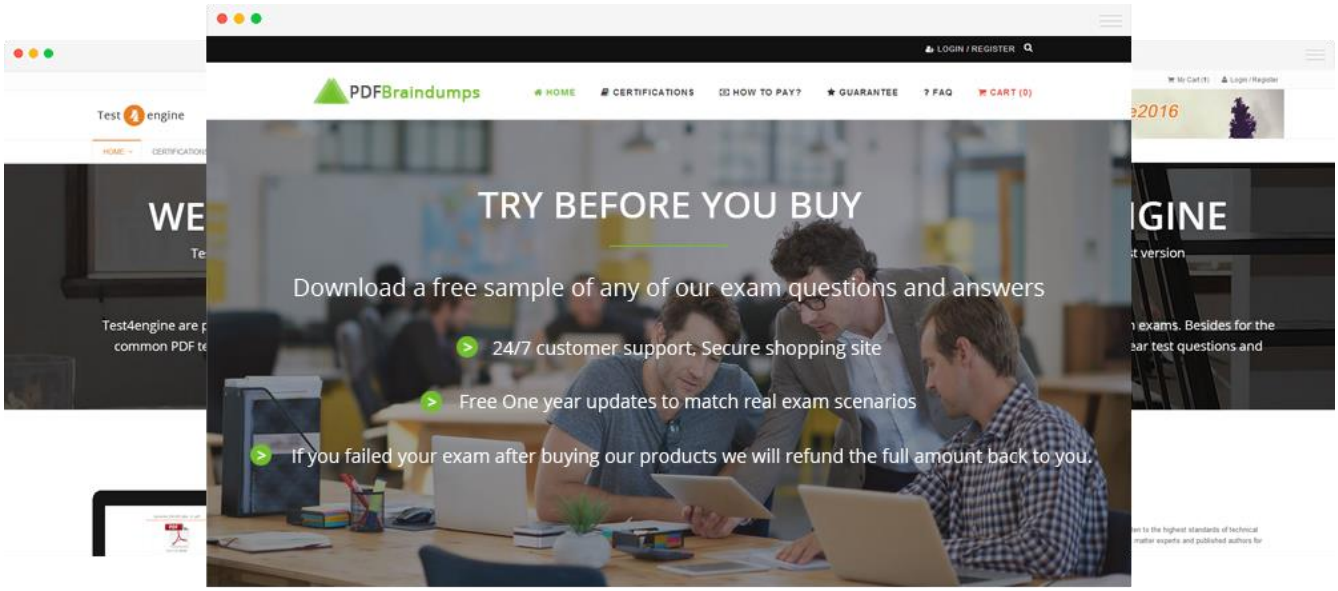
PDFBraindumps



Latest Pdf Braindumps

- LRP-614
- C-SRM-72
- BCABA
- ACMP-6.3
- JN0-740
- 1z0-100
- 250-405
- FM0-308
- DS-200
- C_SRM_72
- SDM_2002001040
- ACMP_6.3
- ST0-250
- 1z0-062
- H12-221
- A4120-784
- M2180-716

Top Certifications

- ISEB Certification
- HCNP
- OCE
- IFPUG Certifications
- NVIDIA Certifications
- dotMobi Certification
- Network+
- SCMA
- IBM Certified Integrat
- MCSD
- CCDH
- NCLP
- IBM Certified Advanc
- XMLMaster Certificat
- eserver Certified Spe
- CS5
- SAP-Certifications
- CHA
- Network Appliance N

Top Vendors

- ISEB
- Fortinet
- ASTQB
- Ericsson
- Aruba
- Liferay
- Data Center Universit
- Novell
- HRCI
- Huawei
- CIW
- RSA
- Patchlink
- MYSQL
- International Consorti
- ISM
- Acme-Packet
- CheckPoint

http://www.pdfbraindumps.com

Latest pdf braindumps provider, high pass rate

Exam : **B2B-Commerce-Developer**

Title : Salesforce Accredited B2B Commerce Developer

Vendor : Salesforce

Version : DEMO

NO.1 What is likely to happen if a developer leaves debug mode turned on in an environment?

A. The performance of the org will become slower each day

B. The user will begin getting JavaScript limit exceptions

C. Theorg will turn off debug mode after 72 hours

D. A banner will be displayed to the user indicating that the org is in debug mode

Answer: D

Explanation:

If a developer leaves debug mode turned on in an environment, the user will begin getting JavaScript limit exceptions.Debug mode is a setting that enables more detailed logging and error reporting for Lightning web components. However, it also increases the size and complexity of the JavaScript code that is delivered to the browser, which can cause performance issues andJavaScript limit exceptions. The JavaScript limit exceptions are errors that occur when the browser reaches its maximum capacity for executing JavaScript code, such as memory heap size or script execution time. The performance of the org will not become slower each day, as debug mode only affects the client-side performance, not the server-side performance. The org will not turn off debug mode after 72 hours, as debug mode is a persistent setting that can only be changed manually by an administrator. A banner will not be displayed to the user indicating that the org is in debug mode, as debug mode is a transparent setting that does not affect the user interface. Salesforce References: Lightning Web Components Developer Guide: Debug Your Code, Lightning WebComponents Developer Guide: JavaScript Limit Exceptions

NO.2 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?

A. Open the flow, select Debug, provide the session ID for replay, and select Run.

B. Open the flow, select Attach to Live Session, provide the session ID, and select Attach.

C. Open the flow, select Debug, provide the Input values, and select Run.

D. Open the flow, select Debug with Inputs, provide the Input values,and select Run.

Answer: A

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, selectDebug, 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 debuggingand 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]

NO.3 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?

A. Right-click the CSS file and choose Diff File Against Org

B. Click the Tools menuand select Diff Styles Against Org...

C. Right-click the folder for the component and choose Diff Styles Against Org

D. Right-click the folder for the component and choose Diff Files Against Org

Answer: A

Explanation:

To know if a CSS file containing small changes wasactually 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 Org is an option that allows the developer to compare a local file with its remote version inthe 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, notjust 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]

NO.4 What is a valid way of referencing the CC Cart Object whose API name is E_Cart__c in a SOQL query?

A. _Cart__c

B. c.E_Cart__c

C. ccrz__E_Cart__c

D. cloudcraze__E_Cart__c

Answer: C

Explanation:
A valid way of referencing the CC Cart Object whose API name is E_Cart__c in a SOQL query is to useccrz__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__ intheir API names. For example, SELECTId, Name FROM ccrz__E_Cart__c will query the CC Cart Object records. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Query Transformation

NO.5 What is a best practice when passing query parameters from user interface to an apex controller?

A. Query parameters should be properly sanitized by using JSINHTMLENCODE within the VisualForce Page or Component.

B. String parameters should be trimmed using String.trim().

C. Query parameters should be passed only to Salesforce B2B Commerce classes thatyou are extending.

D. Query parameters should be stored on a backbone model prior to passing them to the server

Answer: A

Explanation:
A best practice when passing query parameters from user interface to an apex controller is to query parameters should beproperly sanitized by using JSINHTMLENCODE within the VisualForce Page or Component. This function will encode any special characters in the query parameters to prevent cross-site scripting (XSS) attacks or SOQL injection attacks. For example, ccrz.ccRemoteActions.getProducts(' {!
JSINHTMLENCODE(searchTerm)} ') will encode the searchTerm parameter before passing it to the apex controller. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Security

NO.6 A developer is building a custom component in Lightning web components (LWC) that has a grandchild component that needs to pass information to the grandparent component. What is the correct way to demonstrate the passingof a message from the grandchild component to the grandparent component?

A.

Using an attribute on the child component to pass the message from the child to the grandparent

```
// child.js
this.message = 'Hello from Child';

// child.html
<template>
<c-grand-parent message={message}></c-grand-parent>
</template>

// grandParent.js
@api message;
```

Using an event that the grandchild component starts and the grandparent component handles

```
// grandChild.js
this.dispatchEvent(new CustomEvent('message', {
bubbles: true,
detail: { message: 'Hello from Grandchild' }
}));

// grandParent.js
handleMessage(event) {
console.log(event.detail.message); // 'Hello from Grandchild'
}
```

B.

Directly calling a method on the grandparent component from the grandchild component

```
// grandChild.js
this.template.querySelector('c-grand-parent').handleMessage('Hello from Grandchild');
```

C.

```
// grandParent.js
handleMessage(message) {
console.log(message); // 'Hello from Grandchild'
}
```

Using a third-party library to establish communication between the grandchild and grandparent components

```
// grandChild.js
PubSub.publish('message', 'Hello from Grandchild');

// grandParent.js
connectedCallback() {
```

D.

```
this.subscription = PubSub.subscribe('message', (message) => {
console.log(message); // 'Hello from Grandchild'
});
```

Answer: B
Explanation:
The correct way to demonstrate the passing of a message from the grandchild component to the grandparentcomponent in LWC is to use the PubSub library. The PubSub library is a utility that enables Lightning web components to communicate across the DOM tree without using intermediaries. The grandchild component can publish a message to a channel, and the grandparent component can subscribe to that channel and receive the message. This way, the communication is decoupled and does not depend on the component hierarchy. Option B shows the correct syntax for using the PubSub library to pass a message from the grandchild component to the grandparent component. Option A is incorrect because it uses the @api decorator to expose a property on the grandchild component, which is not a valid way to communicate with the grandparent component. Option C is incorrect becauseit uses the @wire decorator to wire a property on the grandchild component to a function on the grandparent component, which is also not a valid way to communicate with the grandparent component. Option D is incorrect because it uses the @track decorator to track a property on the grandchild component, which is not a valid way to communicate with the grandparent component. References: Communicate Across the DOM |Lightning Web Components Developer Guide, Child to grandparent communication in LWC, PubSub Library for Lightning Web Components

NO.7 Which method needsto be implemented when rendering a Salesforce B2B Commerce view in order to have it called after rendering has finished?
A. There are no methods called on the view after rendering has finished
B. onRender()
C. postRender()
D. afterRender()
Answer: C
Explanation:
The method that needs to be implemented when rendering a Salesforce B2B Commerce view in order to have it called after rendering has finished is postRender. This method is an optional hook that can be defined by the user to perform any actionsor logic that depend on the view being rendered, such as initializing widgets, binding events, or updating the user interface. The method will be called by the render method of the CCRZ.
View class, which is the base class for all views in the framework. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, View Class

NO.8 Which two event settings are required for a custom event called CustomEvent to firefrom the Lightning web component and propagate up to the DOM?
A. bubbles: true
B. composed: true
C. cancelable: true

D. composed: false

Answer: A B

Explanation:

To fire a custom event called CustomEvent from the Lightning web component and propagate it up to the DOM,the developer must set two event settings: bubbles and composed. The bubbles setting determines whether the event bubbles up through the component's ancestors in the DOM tree. The composed setting determines whether the event crosses the shadow boundary and reaches the light DOM. Setting both bubbles and composed to true allows the event to be handled by any element in the DOM that listens for it. The cancelable setting is not required for firing or propagating the event, as it only determines whether theevent can be canceled by calling preventDefault() on it. Setting composed to false would prevent the event from reaching the light DOM and limit its propagation to the shadow DOM. Salesforce References: Lightning Web Components Developer Guide: Create and Dispatch Events, Lightning Web Components Developer Guide:
Event Propagation

NO.9 Which two guidelines should a developer consider when migrating aura componentsto LWC?

A. Migrate one component and then determine whether additional effort would make sense

B. Start with migrating trees of components (components within components)

C. Force all developers to write any new components using Lightning web components

D. Start with simple components that only render UI

Answer: A D

Explanation:

When migrating aura components to LWC, a developer should consider two guidelines: migrate one component and then determine whether additional effort would make sense and start with simple components that only render UI. Migrating one component and then determining whether additional effort would make sense allows the developer to evaluate the benefits and costs of migration and decide whether to continue or stop. Migrating simple components that only render UI allows the developer to leverage the performance and modern features of LWC without much complexity or dependency on other components or services. Starting with migrating trees of components (components within components) is not a good guideline, as it can introduce more challenges and dependencies that can complicate the migration process. Forcing all developers to write any new components using Lightning web components is not a good guideline either, as it can create inconsistency and confusion among developers and users. Salesforce References: [Lightning Web Components Developer Guide: Migrate Aura Components to Lightning Web Components], [Lightning Web Components Developer Guide: Migration Considerations]

NO.10 What are three advantages of using ccLog over the Salesforce
standard System.debug class? (3 answers)

A. There is no need to use string concatenation to easily tag log statements with asubject.

B. ccLog can debug syntax errors found in the JavaScript.

C. There is no need to create a User Trace Flag.

D. Append #ccLog= < Logging Token Name > to the end of the storefront URL in order to get logs in the inspector console.

E. There is no need tomanually set a cookie to debug with the Site Guest User.

Answer: A D E

Explanation:
Three advantages of using ccLog over the Salesforce standard System.debug class are:
* There is no need to use string concatenation to easily tag log statements with asubject. ccLog allows passing a subject parameter to the log method, which will prepend the subject to the log message. For example, ccLog.log(' This is a message ' , ' Subject ') will log [Subject] This is a message.
* There is no need to create a User Trace Flag. ccLog can be enabled by setting the value of CO. logToken to true in CCAdmin, which will activate logging for all users who access the storefront.
* There is no need to manually set a cookie to debug with the Site Guest User. ccLog can be enabled for the Site Guest User by appending #ccLog= < Logging Token Name > to the end of the storefront URL in order to get logs in the inspector console. For example, https://my-storefront.com/#ccLog=debug will enable logging for the Site Guest User with the debug level.
Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Logging

NO.11 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?

A. CCRZ.ccPAI.SZ_XL

B. CCRZ.ccPAI.SZ_M

C. CCRZ.ccPAI.SZ_L

D. CCRZ.ccPAI.SZ_S

Answer: B

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

NO.12 Which three actions are applicable when modifying the number of steps required in the Salesforce Commerce Checkout flow? (3 answers)

A. Perform a template override on the Checkout page.

B. Add a page include to the checkout page.

C. Build and activate a new configuration cache setting via CC admin.

D. Set the value of the configuration setting defined as CO.useDef to TRUE

E. Set the value of the configuration setting defined as CO.overrideFlow to TRUE.

Answer: A C E

Explanation:
Three actions that are applicable when modifying the number of steps required in the Salesforce Commerce Checkout flow are:
* Perform a template override on the Checkout page. This action will allow you to change the structure and content of the Checkout page, such as adding or removing sections, widgets, or fields. For example, you can override the checkout.handlebars template and modify it according to your requirements.

* Set the value of the configuration setting defined as CO.overrideFlow to TRUE. This setting will enable you to use your own custom checkout flow instead of the default one. You need to set this value to true before you can modify the checkout flow.
* Set the value of the configuration setting defined as CO.useDef to TRUE. This setting will enable you to use a single-page checkout flow instead of a multi-step checkout flow. You need to set this value to true if you want to reduce the number of steps in the checkout flow to one. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Checkout Flow

NO.13 What does a developer need to do to modify the out-of-the-box checkout flow template?
A. Clone, modify, activate and refer in Experience Builder
B. Modify directly and save to activate
C. Create each flow from scratch
D. Clone, modify and rename to Checkout Flow
Answer: A
Explanation:
To modify the out-of-the-box checkout flow template in Salesforce B2B Commerce, a developer should clone the existing template, make the necessary modifications, activatethe modified template, and then reference it in the Experience Builder. This approach ensures that the original template remains intact and provides a fallback option. Salesforce documentation on customizing the checkout flow in B2B Commerce emphasizes the importance of using the Experience Builder for suchcustomizations, providing a visual interface to manage and reference different checkout flow templates.

NO.14 What is the fastest route to setting up a B2B Commerce Store as a developer?
A. Set up B2B Commerce on Lightning Experiencemanually
B. Create a new store in the Commerce app
C. Import a previously exported store archive
D. Use sfdx setup scripts
Answer: C
Explanation:
The fastest route to setting up a B2B Commerce store as a developer is to use sfdx setup scripts. Sfdx setup scripts arescripts that use Salesforce CLI commands to automate the creation and configuration of a B2B Commerce store. The scripts can perform tasks such as creating scratch orgs, installing packages, importing data, assigning permissions, and deploying code. The scripts can save time and effort for developers who need to set up a B2B Commerce store quickly and easily. Setting up B2B Commerce on Lightning Experience manually is not the fastest route to setting up a B2B Commerce store, as it involves many steps and actions that can be tedious and error-prone. Creating a new store in the Commerce app is not the fastest route either, as it also requires manual configuration and customization of various settings and features. Importing a previously exported store archiveis not the fastest route either, as it depends on the availability and quality of the store archive and may not reflect the latest changes or updates. Salesforce References: [B2B Commerce Developer Guide: Set Up Your Development Environment], [B2B CommerceDeveloper Guide: Create Your Store]

NO.15 Why is the use ofa standard Visualforce control such as apex:form discouraged in Salesforce B2B Commerce page includes and subscriber pages?

A. Visualforce " scopes " controls that are present on a page and scope of the control will be set to " ccrz "
B. Apex:form render DOMcomponents slowly
C. The CCRZ Javascript object is not accessible within an apex:form control.
D. Javascript events are not supported within an apex:form control
Answer: C
Explanation:
The use of a standard Visualforce control such as apex:form is discouraged in Salesforce B2B Commerce page includes and subscriber pages because the CCRZ JavaScript object is not accessible within an apex:form control. The CCRZ JavaScript object is a global object that provides access to various B2B Commerce functions and properties, such as logging, utilities, events, and configuration. Using an apex:form control would prevent the developer from using the CCRZ JavaScript object and its features. Visualforce does not
"scope" controls that are present on a page and scope of the control will be set to "ccrz", as this is not a valid statement. Apex:form does not render DOM components slowly, as this is not a performance issue. JavaScript events are supported within an apex:form control, as this is not a limitation. Salesforce References: B2B Commerce Developer Guide: CCRZ JavaScript Object, B2B Commerce Developer Guide: Page Includes

NO.16 Which three pages should be enabled for the Guest user profile for a storefront to have anonymous checkout?
(3 answers)
A. CCPaymentInfo
B. CheckoutNew
C. OrderView
D. Checkout
E. OrderConfirmation
Answer: A B E
Explanation:
Three pages that should be enabled for the Guest user profile for a storefront tohave anonymous checkout are:
* CCPaymentInfo: This page allows the guest user to enter their payment information, such as credit card number, expiration date, and security code. The page also displays the order summary and total amount.
* CheckoutNew: Thispage allows the guest user to enter their shipping and billing information, such as name, address, phone number, and email. The page also displays the cart items and shipping options.
* OrderConfirmation: This page displays the confirmation message and ordernumber after the guest user places their order. The page also provides a link to view the order details or print the invoice.
Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Anonymous Checkout

NO.17 What is a methodto resolve if the current storefront customer is a Salesforce B2B Commerce guest user in an apex class?
A. ccrz.cc_CallContext.currUser.isGuest
B. ccrz.cc_CallContext.isGuest

C. UserInfo.getUserType()
D. ... UserType
Answer: B
Explanation:
A method to resolve if the current storefront customer is a Salesforce B2B Commerce guest user in an apex class is to use ccrz.cc_CallContext.isGuest. This property will return true if the current user is a guest user, or false otherwise. For example, if(ccrz.cc_CallContext.isGuest){ // do something for guest user } will execute some logic only for guest users. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Call Context

NO.18 A developer is debugging a flow and needs to watch all the variables changing asthe checkout process is executed, but nothing is displaying. Which two features did the developer forget to enable?
A. Set up a debug tog to show the details of what is executed
B. Show the details of what is executed and render flow in Lightning Runtime
C. Run the latest version of each flow called by subtle w elements
D. Show the details of what is executed and render flow in Lightning Experience.
Answer: B D
Explanation:
To debug a flow and watch all the variables changing as the checkout process is executed, thedeveloper needs to enable two features: show the details of what is executed and render flow in either Lightning Runtime or Lightning Experience. These features are available in the debug options in Flow Builder, and they allow the developer to see real-time details of the flow actions, inputs, outputs, and outcomes in a panel on the right. The developer can also set input variables, restart the flow, and convert the debug run to a test. Option A is incorrect because there is no such thing as a debug tog in Flow Builder. Option C is incorrect because running the latest version of each flow called by subflow elements is not a feature that the developer can enable or disable, but rather a default behavior of Flow Builder. References: Debug a Flow in Flow Builder, B2B Commerce Checkout Flow (Aura), B2B Commerce Checkout Flow Core Actions

NO.19 Which three steps are ne cessary to have subscriberpage added to Salesforce B2B Commerce after creating a custom Visualforce page? (3 answers)
A. Create a new CC Subscriber Page record that points to your custom Visualforce page.
B. Create a new Visualforce page, and manually import the Salesforce B2BCommerce JavaScript libraries.
Run in Anonymous Apex ccrz.cc_util_Reflection.upsertPageUIKey([arg1],[arg2] ,[arg3]);
C. Refresh the Page Keys Index in CC Admin.
D. Enable the Subscriber Page in CC Admin.
Answer: A C D
Explanation:
Three steps that are necessary to have a subscriber page added to Salesforce B2B Commerce after creating a custom Visualforce page are:
* Create a new CC Subscriber Page record that points to your custom Visualforce page. This record will store information about the subscriberpage, such as the name, description, URL, and Visualforce page.

For example, you can create a new record named MySubscriberPage that points to your custom Visualforce page named MyPage.
* Refresh the Page Keys Index in CC Admin. This action will update the page keys index, which is a cache that stores the mapping between the page keys and the subscriber pages. You need to do this whenever you create or modify a subscriber page record.
* Enable the Subscriber Page in CC Admin. This action will allow you to select the subscriber page from the CC Page Settings configuration and assign it to a CC Page. For example, you can enable MySubscriberPage and assign it to the Home page. Salesforce References: B2B Commerce and D2C Commerce Developer Guide, Subscriber Pages

NO.20 Which two statement are true for Mass Order (2 answers)
A. MassOrder pricing is done via a batch job.
B. Mass order works with thedefault wishlists
C. The variation product is leveraged for SKUs.
D. Mass Order is mobile ready with the ccrz templates.
Answer: A C
Explanation:
Mass Order pricing is done via a batch job, which means that the prices are not calculated in real time, but rather at a scheduled time. Mass order works with the variation product, which is a product that has multiple SKUs with different attributes, such as size or color. Mass order does not workwith the default wishlists, which are used to store products that the customer wants to buy later. Mass order is not mobile ready with the ccrz templates, which are the defaulttemplates for the storefront pages. Salesforce References: B2B Commerce Developer Guide: Mass Order, B2B Commerce Developer Guide: Variation Products

NO.21 How is a price group dynamically set?
A. By overriding the ccLogicProductPrice class
B. By using contract pricing
C. By extending the ccApiPriceList API
D. By extending the cc_hk_priceing hook
Answer: D
Explanation:
A price group can be dynamically set by extending the cc_hk_pricing hook. This hook allows modifying the price group based on various factors, such as the user, cart, product, orstorefront. For example, the hook can apply a different price group for a specific product category or for a specific user segment.

NO.22 Which three considerations should a developer keep in mind when creating a tax provider?
A. What events to fire in the Lightning Web Component
B. Whether to use JSON or XML
C. Success criteria
D. Whether an AppExchange package already exists
E. How to handle errors
Answer: B C E
Explanation:

When creating a tax provider in SalesforceB2B Commerce, developers should consider the data format (JSON or XML) for interoperability with the tax service, define clear success criteria to ensure accurate tax calculations, and implement robust error handling to manage exceptions and failures gracefully. Salesforce B2B Commerce documentation emphasizes the importance of these considerations for integrating external services, ensuring reliability and consistency in tax calculations across different jurisdictions and scenarios.