Ultimate IT Test Preparation Material

Salesforce

B2B-Commerce-Developer
Salesforce Accredited B2B Commerce Developer

- **Up to Date products, reliable and verified.**
- **Questions and Answers in PDF Format.**

Full Version Features:

- **90 Days Free Updates**
- **30 Days Money Back Guarantee**
- **Instant Download Once Purchased**
- **24 Hours Live Chat Support**

For More Information:

https://www.testsexpert.com/

- **Product Version**

Visit us at

Latest Version: 12.0

Question: 1

Although Salesforce B2B Commerce and Salesforce recommend against using "without sharing classes" whenever possible, sometimes it is unavoidable. Which three items will open up a major security hole? (3 answers)

A. Executing dynamic SOQL inside a without sharing class with a bind variable fromPageReference.getParameters().
B. Executing dynamic SOQL inside a without sharing class with a bind variable from theUserInfo class.
C. Executing dynamic SOQL inside a without sharing class with a bind variable fromPageReference.getCookies().
D. Executing dynamic SOQL inside a without sharing class with a bind variable fromcc_RemoteActionContentex class.
E. Executing dynamic SOQL inside a without sharing class with a bind variable fromccAPI.CURRENT_VERSION.

Answer: A,C,D

Explanation:
Executing dynamic SOQL inside a without sharing class with a bind variable from PageReference.getParameters(), PageReference.getCookies(), or cc_RemoteActionContext class will open up a major security hole because these sources of input are not sanitized and can be manipulated by malicious users to inject SOQL queries that bypass the sharing rules and access data that they are not supposed to see. For example, a user can modify the URL parameters or cookies to include a SOQL query that returns sensitive data from the database. To prevent this, it is recommended to use static SOQL or escape the bind variables before executing dynamic SOQL.

Question: 2

The ccrz.cc_hk_UserInterface apex class, HTMLHead Include Begin and HTML Head Include End Cloudcraze Page Include sections allow additional content to be added to the HTML <head> tag. What are two reasons that is it preferred to use the ccrz.cc_hk_UserInterface extension over the Cloudcraze Page Include sections? (2 answers)

A. Salesforce apex:include is wrapped in tags.
B. HTML does not support tags inside the <head>
C. Salesforce apex:include is wrapped in tags.
D. HTML does not support tags inside the <head>

Answer: A,D

Explanation:
The ccrz.cc_hk_UserInterface apex class is preferred over the HTMLHead Include Begin and HTML Head Include End Cloudcraze Page Include sections because Salesforce apex:include is wrapped in tags, which are not valid inside the HTML <head> tag. This can cause rendering issues or unexpected behavior in some browsers. The ccrz.cc_hk_UserInterface extension allows adding content to the HTML <head> tag without using apex:include.

Question: 3

The ccUtil apex class in Salesforce B2B Commerce provides numerous utility functions that can be leveraged in subscriber classes.
What are two ways to check the input or return data of the Global API's? (2 answers)

A. ccrz.ccUtil.isNotEmpty(Map<String, Object>) andccrz.ccUtil.isNotEmpty(List<Object>)
B. ccrz.ccUtil.isNotValid(Map<String, Object>) andccrz.ccUtil.isNotValid(List<Object>)
C. ccrz.ccUtil.isValid(Map<String, Object>) and ccrz.ccUtil.isValid(List<Object>)
D. ccrz.ccUtil.isEmpty(Map<String, Object>) and ccrz.ccUtil.isEmpty(List<Object>)

Answer: A,D

Explanation:
The ccUtil apex class provides two methods to check the input or return data of the Global API's: ccrz.ccUtil.isNotEmpty(Map<String, Object>) and ccrz.ccUtil.isEmpty(Map<String, Object>). These methods return true if the map is not null and contains at least one entry, or if the map is null or empty, respectively. Similarly, ccrz.ccUtil.isNotEmpty(List<Object>) and ccrz.ccUtil.isEmpty(List<Object>) return true if the list is not null and contains at least one element, or if the list is null or empty, respectively. These methods are useful for validating the input parameters or the output results of the Global API's.

Question: 4

The ccUtil apex class in Salesforce B2B Commerce provides numerous utility functions that can be leveraged in subscriber classes. Which command will return the value in the given Map if found or a default value in the event that the Map is null, empty, or an object is not found for that key?

A. ccrz.ccUtil.defv (Map<String.Object> mp, String key , Object ob)
B. ccrz.ccUtil.defVal (Map<String.Object> mp, String key, Object ob)
C. ccrz.ccUtil.... (Map<String.Object> mp, String key, Object ob)
D. ccrz.ccUtil.defaultValue(Map<String.Object> mp, String key , Object ob)

Answer: B

Explanation:

The ccrz.ccUtil.defVal (Map<String.Object> mp, String key, Object ob) method will return the value in the given Map if found or a default value in the event that the Map is null, empty, or an object is not found for that key. This method is useful for providing fallback values for configuration settings or input parameters that may be missing or invalid. Salesforce Reference: B2B Commerce and D2C Commerce Developer Guide, ccUtil Class

Question: 5

A configuration value, CO.NewOrder, is set to TRUE. What is one way of
preventing anexisting payment page from being shown on the checkout payment page?

A. Delete the Visualforce page from the code base.
B. Remove the value matching the page name from the pmt.whitelist configurationsetting, then rebuild and activate a new Configuration cache
C. Remove the payment type associated with the payment page from CO.pmts, thenrebuild and activate a new cache.
D. Override the front end template and modify the way the embedded payment page getsloaded from the payment list configuration.

Answer: B

Explanation:
This approach effectively removes the payment page from the list of allowed pages, ensuring it is not displayed during the checkout process.
Reference: Salesforce B2B Commerce documentation on checkout process customization and configuration settings, specifically focusing on payment page handling and the pmt.whitelist setting.
The pmt.whitelist configuration setting in Salesforce B2B Commerce is used to manage the Visualforce pages that support all payment types on the storefront1. If you want to prevent an existing payment page from being shown on the checkout payment page, one way to do it is to remove the value matching the page name from the pmt.whitelist configuration setting. After doing this, you would need to rebuild and activate a new Configuration cache for the changes to take effect1. Please note that this information is based on the Salesforce B2B Commerce documentation and best practices1.

For More Information – **Visit link below:**

https://www.testsexpert.com/

16$ Discount Coupon: 9M2GK4NW

Features:

- Money Back Guarantee……..……...…. 

- 100% Course Coverage…… 

- 90 Days Free Updates…… 

- Instant Email Delivery after Order……