



Free Questions for PDII by dumpssheet

Shared by Baxter on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A company has a custom component that allows users to search for records of a certain object type by invoking an Apex Controller that returns a list of results based on the user's input. When the search is completed, a searchComplete event is fired, with the results put in a results attribute of the event. The component is designed to be used within other components and may appear on a single page more than once.

What is the optimal code that should be added to fire the event when the search has completed?

A)

```
var evt = component.getEvent("searchComplete");  
evt.setParams({results: results});  
evt.fire();
```

B)

```
var evt = $A.get("e.c.searchComplete");  
evt.set("v.results", results);  
evt.fire();
```

C)

```
var evt = component.getEvent("searchComplete");  
evt.set("v.results", results);  
evt.fire();
```

D)

```
var evt = $A.get("e.c.searchComplete");  
evt.setParams({results: results});  
evt.fire();
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

D

Question 2

Question Type: MultipleChoice

Universal Containers is using a custom Salesforce application to manage customer support cases. The support team needs to collaborate with external partners to resolve certain cases. However, they want to control the visibility and access to the cases shared with the external partners. Which Salesforce feature can help achieve this requirement?

Options:

- A- Role hierarchy
- B- Criteria-based sharing rules
- C- Apex managed sharing
- D- Sharing sets

Answer:

D

Question 3

Question Type: MultipleChoice

Assuming the CreateOneAccount class creates one account and implements the

Queueable interface, which syntax properly tests the Apex code?

A)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
Test.startTest();  
System.assertEquals(1, accts.size() );  
Test.stopTest();
```

B)

```
List<Account> accts;  
Test.startTest();  
System.enqueueJob( new CreateOneAccount() );  
Test.stopTest();  
accts = [SELECT Id FROM Account];  
System.assertEquals( 1, accts.size() );
```

C)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
Test.getFlexQueueOrder();  
System.assertEquals( 1, accts.size() );
```

D)

```
List<Account> accts;  
System.enqueueJob( new CreateOneAccount() );  
accts = [SELECT Id FROM Account];  
System.assertEquals(1, accts.size() );
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 4

Question Type: MultipleChoice

A developer used custom settings to store some configuration data that changes occasionally. However, tests are now Failing in some of the sandboxes that were recently refreshed.

What should be done to eliminate this issue going forward?

Options:

A- Set the setting type on the custom setting to List.

B- Replace custom settings with static resources.

C- Set the setting type on the custom setting to Hierarchy.

D- Replace custom settings with custom metadata.

Answer:

D

Question 5

Question Type: MultipleChoice

A developer wrote a class named `AccountHistoryManager` that relies on field history tracking. The class has a static method called `getAccountHistory` that takes in an `Account` as a parameter and returns a list of associated `AccountHistory` object records.

The following test fails:

```
BigTest
```

```
public static void testAccountHistory() {  
    Account a = new Account(name = 'test');  
    insert a;  
    a.name = a.name + '1';  
    update a;  
    List<AccountHistory> ahList =  
AccountHistoryManager.getAccountHistory(a);  
    System.assert (ahList.size() > 0);  
}
```

What should be done to make this test pass?

Options:

A- Create Accountdistory records manually in the test setup and write a query to get them.

- B-** Use @isTest (SeeAllData=true] to see historical data from the org and query for AccountHistory records.
- C-** Use Test.isRunningTest () in getAccountHistory() to conditionally return fake AccountHistory records.
- D-** The tact method should be deleted since this code cannot be tested

Answer:

C

Question 6

Question Type: MultipleChoice

Business rules require a Contact to always be created when a new Account is created.

What can be used when developing a custom screen to ensure an Account is not created if the creation of the Contact fails?

Options:

- A-** Use setSaverpoint (1) and rollback() with a try-catch block.
- B-** Use the Database.Insect method with a11orNone SEL LO alee,

- C- Use the Database. Delete method if the Contact insertion fails.
- D- Disable validation rules on Contacts and set default values with a trigger.

Answer:

A

Question 7

Question Type: MultipleChoice

For compliance purposes, a company is required to track long-term product usage in their org. The information that they need to log will be collected from more than one object and, over time, they predict they will have hundreds of millions of records.

What should a developer use to implement this?

Options:

- A- Setup audit trail
- B- Field audit trail
- C- Big objects

D- Field history tracking

Answer:

C

To Get Premium Files for PDII Visit

<https://www.p2pexams.com/products/pdii>

For More Free Questions Visit

<https://www.p2pexams.com/salesforce/pdf/pdii>

