TestHorse

Certified IT practice exam authority

Accurate study guides, High passing rate!
Testhorse provides update free of charge in one year!

http://www.testhorse.com

Exam : PDII

Title : Salesforce Certified Platform Developer II

Version : DEMO

1.A developer created a Lightning web component that allows users to input a text value that is used to search for Accounts by calling an Apex method. The Apex method returns a list of AccountWrappers and is called imperatively from a JavaScript event handler.

```
01:
02:  public class AccountSearcher {
03:
04:      public static List<AccountWrapper> search(String term) {
05:          List<AccountWrapper> wrappers = getMatchingAccountWrappers(term);
06:          return wrappers;
07:      }
08:
09:
10:      public class AccountWrapper {
11:          public Account { get; set; }
12:          public Decimal matchProbability { get; set; }
13:      }
14:      // ...other methods, including getMatchingAccountWrappers implementation...
15: }
```

Which two changes should the developer make so the Apex method functions correctly? Choose 2 answers

A. Add @AuraEnafcle3 to line 09.

B. Add @AuraEnabled to line 03.

C. Add @AuraEnabled to lines 11 and 12.

D. Add @AuraEnat:ei to line 01.

Answer: B,C

2.A developer created a JavaScript library that simplifies the development of repetitive tasks and features and uploaded the library as a static resource called jsutils in Salesforce. Another developer is coding a new Lightning web component (LWC) and wants to leverage the library.

Which statement properly loads the static resource within the LWC?

A. import {jsUtilities} from '@salesforce/reaourceUrljsUtila';

B. import jUtilities from '@salesforce/reaourceUrljsUtila';

C. <lightning-require scripts=N { ! SReaource. j sUtils}''/>

D. const jsUtility = SA.get ('SReaource.jsUtils');

Answer: B

3.Refer to the following code snippets:

MyOpportunities.js

```
import { LightningElement, api, wire } from 'lwc';
import getOpportunities from '@salesforce/apex/OpportunityController.findMyOpportunities';

export default class MyOpportunities extends LightningElement {
    @api userId;
    @wire(getOpportunities, {oppOwner: '$userId'})
    opportunities;
}
```

OpportunityController.cls

```
public with sharing class OpportunityController {
    @AuraEnabled
    public static List<Opportunity> findMyOpportunities(Id oppOwner) {
        return [
            SELECT Id, Name, Amount
            FROM Opportunity
            WHERE OwnerId = :oppOwner
            WITH SECURITY_ENFORCED
            LIMIT 10
        ];
    }
```

A developer is experiencing issues with a Lightning web component. The component must surface information about Opportunities owned by the currently logged-in user
When the component Is rendered, the following message Is displayed: "Error retrieving data".
Which modification should be implemented to the Apex class to overcome the issue7
A. Ensure the OWD for the Opportunity object is Public.
B. Use the Cacheable-true attribute in the Apex method.
C. Edit the code to use the without sharing keyword in the Apex class.
D. Use the continuation-true attribute in the Apex method.
Answer: B

4.Part of a custom Lightning Component displays the total number of Opportunities in the org, which is in the millions. The Lightning Component uses an Apex Controller to get the data it needs.
What is the optimal way for a developer to get the total number of Opportunities for the Lightning Component?
A. SUM() SOQL aggregate query on the Opportunity object
B. SOQL for loop that counts the number of Opportunities records
C. COUNT() SOQL aggregate query on the Opportunity object
D. Apex Batch job that counts the number of Opportunity records
Answer: C

5.Refer to the test method below:

```
@isTest
static void testAccountUpdate() {
    Account acct = new Account(Name = 'Test');
    acct.Integration_Updated__c = false;
    insert acct;

    CalloutUtil.sendAccountUpdate(acct.Id);

    Account acctAfter = [SELECT Id, Integration_Updated__c FROM Account WHERE Id = :acct.Id][0];

    System.assert(true, acctAfter.Integration_Updated__c);
}
```

An external system with Account information and sets the Account's Integration-Updated___ c checkbox to True when it completes.

The test fails to execute and exits with an error: "Methods defined as TestMethod do not support Web service callouts."

What is the optimal way to fix this?

A. Add if ('Test.isRunningTest l)) around calloutUtil.sendAccountUpdate.

B. Add Test.startTest ()before and Test.stopTest () after CallcutoutUtil. sendAccoutUpdate.

C. Add Test.startTest before and Test.setMock and Test. stoptest () after CallloutUtil.sendAccountUpdate.

D. Add Test.startTest() and Test .setMock before and Test.stopTest () after CalloutUtil.sendAccountUpdate.

Answer: D