Free Questions for B2B-Commerce-Developer

Shared by Phillips on 04-10-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Universal Containers (UC) needs to display data from standard objects (entities) in a different format than what comes with B2B Commerce out of the box. In doing this, what is one advantage of using the Lightning Data Service vs using a custom Controller class?

Options:

A- Lightning Data Service translates the developer's component implementation to a VisualForce page for backward compatibility.

B- JavaScript proxies for transport objects are created in the developer's IDE automatically.

C- The developer can read, create, or modify single records or metadata without writing Apex code.

D- There is a Visual Studio add-in that accelerates the layout process

Answer:

C

Explanation:

Lightning Data Service (LDS) is a Salesforce UI API layer that empowers developers to perform CRUD operations and manage Salesforce data without the need for custom Apex controllers. LDS is designed to improve performance, user interface consistency, and developer productivity by handling data operations and sharing rules, thus reducing the need for server-side code. For more details, refer to the Salesforce documentation on Lightning Data Service: Salesforce Lightning Data Service Documentation.

Question 2

Question Type: **MultipleChoice**

A Northern Trail Qutfitters (NTO) developer made a tile component. To expose a click event and react to user input using the markup below, what should replace ?

```
<template>
  <div class="container">
    <a onclick=CLICK_EVENT>
      <div class="title">{product.fields.Name.value}</div>
      <img class="product-img" src={product.fields.Picture_URL__c.value}></img>
    </a>
  </div>
</template>
```

Options:

A- tileClick()

B- {event:tileClick}

C- javascript:avoid(0);tileClick();

D- {tileClick}

Answer:

A

Explanation:

To expose a click event and react to user input in a Lightning web component, the developer should use a method name as the value of the onclick attribute in the template. The method name should be followed by parentheses, as in tileClick(). This syntax indicates that the method is invoked when the element is clicked. The method should be defined in the JavaScript file of the component, and it can access the event object as a parameter. The other options are either invalid or incorrect. For example, {event:tileClick} is not a valid syntax for an onclick attribute, javascript:avoid(0);tileClick(); is unnecessary and outdated, and {tileClick} is a property binding expression, not a method invocation.Reference:B2B Commerce and D2C Commerce Developer Guide,Lightning Web Components Developer Guide

Question 3

Question Type: MultipleChoice

Which two blocks of code are needed to implement a custom getter in a Lightning web component?

A)

```
set rows(value) {
this.state.rows = value;
}
```

B)

```
get rows() {
return this.state.rows;
}
```

C)

```
@api
set rows(value) {
this.state.rows = value;
}
```

D)

```
@api
get rows() {
return this.state.rows;
}
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

A, D

Explanation:

custom getter in a Lightning web component is a JavaScript function that executes logic each time a public property is accessed. A custom getter must start with the keywordgetand be followed by a name for the property. A custom getter must also have a corresponding custom setter, which is a function that executes logic each time a public property is assigned a value. A custom setter must start with the keywordsetand have the same name as the getter. One of the getter or setter functions must be annotated with@api, which makes the property public and reactive.

Option A and Option D show the correct syntax for defining a custom getter and setter for a public property calledname. Option A shows the getter function, which returns the value of a private property called_name. Option D shows the setter function, which assigns the value of the parametervalueto the private property_name. The getter function is annotated with@api, which makes thenameproperty public and reactive.

Option B and Option C are incorrect because they do not follow the syntax for a custom getter and setter. Option B shows a regular function declaration, not a getter function. Option C shows a regular assignment statement, not a setter function. Neither option uses the@apidecorator, which is required for a public property.Reference:

Getters and Setters

Understand getter in Lightning Web component

Question 4

Question Type: MultipleChoice

How should a developer get the grand total amount, including shipping and tax, for items in the

cart and in the currency of the cart, when developing a new Lightning web component for an

Aura storefront cart page?

Options:

A- {!Cart.Details.grandTotal}

B- {ICart.Totals.grand Total}

C- {ICart.Details.Fields.grandTotal}

D- {!Cart.Fields.grandTotal}

Answer:

C

Explanation:

According to theB2B Commerce Developer Guide, the ICart interface provides access to the cart object and its related data. The Details property of the ICart interface returns an ICartDetails object, which contains information about the cart such as the currency, the subtotal, the shipping cost, the tax, and the grand total. The Fields property of the ICartDetails interface returns a map of field names and values for the cart object. Therefore, to get the grand total amount for items in the cart and in the currency of the cart, a developer should use the expression {ICart.Details.Fields.grandTotal}, which returns the value of the grandTotal field from the cart object.Reference:B2B Commerce Developer Guide,ICart Interface,ICartDetails Interface

Question 5

Question Type: MultipleChoice

Universal Containers (UC) needs to wrap a Lightning Web Component they have created called "lwcContainerComponent" inside an Aura component. Which set of tags is the correct approach?

A)

```
<!-- auraWrapper.cmp -->
<aura:component>
  <aura:lwcContainerComponent />
</aura:component>
```

B)

```
<!-- auraWrapper.cmp -->
<aura:component>
  <aura:c:lwcContainerComponent />
</aura:component>
```

C)

```
<!-- auraWrapper.cmp -->
<aura:component>
   <c:lwcContainerComponent />
</aura:component>
```

```
<!-- auraWrapper.cmp -->
<aura-lwc:component>
   <c:lwcContainerComponent />
</aura-lwc:component>
```

Options:

A- Option

B- Option

C- Option

D- Option

Answer:

C

Explanation:

To wrap a Lightning Web Component inside an Aura component, you need to use the<c:lwcContainerComponent>tag, wherecis the default namespace for custom components. This tag allows you to reference the Lightning Web Component by its name and use it as a child component of the Aura component. You also need to use thelwc:dom='manual'directive on the Aura component to indicate that you

are manually rendering the Lightning Web Component inside the Aura component. This directive prevents the Aura component from interfering with the rendering of the Lightning Web Component. Option C shows the correct syntax for wrapping a Lightning Web Component inside an Aura component. Option A is incorrect because it uses the<lightning:lwcContainerComponent>tag, which is not a valid tag for referencing a Lightning Web Component. Option B is incorrect because it uses thetag, which is also not a valid tag for referencing a Lightning Web Component. Option D is incorrect because it does not use thelwc:dom='manual'directive on the Aura component, which is required for manually rendering a Lightning Web Component inside an Aura component.Reference:Composition | Lightning Web Components Developer Guide,Using Lightning Web Components inside Aura Components,Aura Coexistence | Lightning Web Components Developer Guide

Question 6

Question Type: MultipleChoice

A developer has just deployed a new Lightning web component called myNewLwcComp to an

authorized org. The developer tries to find the component in the Lighting Page Builder, but it

does not come up in searches. Which two steps should the developer take next?

Options:

A- Ensure that the metadata isExposed property is set properly in source code

B- Redeploy the component

C- Close the browser and reopen the page

D- Ensure it has a target of lightning__FlowScreen

Answer:

A, D

Explanation:

To make a Lightning web component available in the Lightning Page Builder or Experience Builder, the developer needs to do two things: set the isExposed property to true in the component's metadata file, and define at least one target that specifies where the component can be used, such as a Lightning page type or a flow screen. Redeploying the component or closing and reopening the browser will not make the component appear in the searches if the metadata file is not configured properly.Reference:

XML Configuration File Elements

Supported Salesforce Targets and Tools

#8: Use Lightning Web Components in Salesforce Targets

Question 7

Question Type: MultipleChoice

An administrator has just provided a developer with a new org and username. Which two sets of steps can the developer use to authorize the org and begin deploying Lightning web components?

What should a developer do to expose a public property in a Lightning web component?

Options:

A- Decorate the field with @property

B- Decorate the field with @track

C- Decorate the field with @public

D- Decorate the field with @api

Answer:

D

Explanation:

To expose a public property in a Lightning web component, the developer should decorate the field with the @api decorator. This decorator marks the property as public, which means that it can be set by another component, such as a parent component. The @api decorator also makes the property reactive, which means that any changes to the property value are reflected in the component's template. The other decorators (@property, @track, and @public) are not valid for exposing public properties in Lightning web components.Reference:B2B Commerce and D2C Commerce Developer Guide, [Lightning Web Components Developer Guide]