



**Free Questions for B2B-Commerce-Administrator by
dumpssheet**

Shared by Cabrera on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Which two methods should a developer implement in a Lightning web component to successfully handle the subscription lifecycle of a Message Channel?

Options:

- A- Subscribe()
- B- stopListener()
- C- startListener()
- D- unsubscribe()

Answer:

A, D

Explanation:

To handle the subscription lifecycle of a Message Channel, a developer should implement two methods in a Lightning web component: `subscribe()` and `unsubscribe()`. The `subscribe()` method is used to register a callback function that handles the messages published on the message channel. The `unsubscribe()` method is used to remove the subscription and stop receiving messages. Both methods are part of the Lightning Message Service API, which is imported from the `lightning/messageService` module¹. For example:

```
// Import the Lightning Message Service API import { subscribe, unsubscribe, MessageContext } from 'lwc/messageService';

// Declare a message context @wire(MessageContext) messageContext;

// Declare a subscription variable subscription = null;

// Subscribe to the message channel in the connected callback connectedCallback() { this.subscribeToMessageChannel(); }

// Subscribe to the message channel using the LMS API subscribeToMessageChannel() { if (!this.subscription) { this.subscription =
subscribe( this.messageContext, MESSAGE_CHANNEL, // The name of the message channel (message) =>
this.handleMessage(message) // The callback function ); } }

// Unsubscribe from the message channel in the disconnected callback disconnectedCallback() {
this.unsubscribeFromMessageChannel(); }

// Unsubscribe from the message channel using the LMS API unsubscribeFromMessageChannel() { unsubscribe(this.subscription);
this.subscription = null; }
```

Question 2

Question Type: MultipleChoice

Which category can receive signaling from a Lightning Message Channel? 03m 42s

Options:

- A- only descendents (i.e. children)
- B- page (anywhere on it)
- C- only direct siblings
- D- only ancestors (i.e. parents)

Answer:

B

Explanation:

The Lightning Message Service (LMS) allows communication across different UI technologies on a Lightning page, such as Lightning Web Components, Aura Components, and Visualforce pages. LMS uses message channels to publish and subscribe to messages. A message channel is a custom metadata type that defines the shape and scope of the messages¹. The scope of a message channel determines which components can receive signaling from it. There are two possible scopes: application and component¹. The application scope means that any component on the page can receive signaling from the message channel, regardless of its position in the DOM hierarchy¹. The component scope means that only components that share a common ancestor can receive signaling from the message

channel1. Therefore, the correct answer is B. page (anywhere on it), because it corresponds to the application scope of a message channel.Reference:Use Lightning Message Service

Question 3

Question Type: MultipleChoice

A developer has created a custom Lightning web component for the Cart page that needs to react to changes to cart items from the standard cart component.

How should the developer implement the custom component so changes to cart items and quantities are reflected?

Options:

- A-** Subscribe to events on the lightning_commerce_cartChanged channel using the Lightning Message Service.
- B-** Add a listener for the cartItemUpdate Lightning event.
- C-** Listen for events on the lightning_commerce_cartChanged channel with the Lightning Event "Listener component.
- D-** Add an event listener for the cartchanged DOM (Document Object Model) event.

Answer:

A

Explanation:

The Lightning Message Service (LMS) is a feature that allows communication across different UI technologies, such as Lightning Web Components, Aura Components, and Visualforce pages. LMS uses message channels to publish and subscribe to messages. A message channel is a custom metadata type that defines the shape and scope of the messages¹. The lightning_commerce_cartChanged channel is a standard message channel that is used by the B2B Commerce Cloud platform to notify components of changes to the cart items and quantities². A developer can use the LMS API to subscribe to events on this channel and update the custom component accordingly. The developer needs to import the lightning/messageService module and the lightning_commerce_cartChanged channel in the JavaScript file of the custom component, and then use the subscribe method to register a callback function that handles the message payload³. For example:

```
// cartCustomComponent.js import { LightningElement, wire } from 'lwc'; import { subscribe, MessageContext } from
'lightning/messageService'; import CART_CHANGED_CHANNEL from
'@salesforce/messageChannel/lightning_commerce_cartChanged__c';

export default class CartCustomComponent extends LightningElement { // Declare a message context @wire(MessageContext)
messageContext;

// Declare a subscription variable subscription = null;

// Subscribe to the message channel in the connected callback connectedCallback() { this.subscribeToMessageChannel(); }
```

```
// Subscribe to the message channel using the LMS API subscribeToMessageChannel() { if (!this.subscription) { this.subscription =  
subscribe( this.messageContext, CART_CHANGED_CHANNEL, (message) => this.handleCartChange(message) ); } }
```

```
// Handle the message payload and update the component logic handleCartChange(message) { // Do something with the message  
payload, such as: // - Display the cart items and quantities // - Calculate the cart total // - Apply discounts or taxes // - etc. } }
```

Question 4

Question Type: MultipleChoice

Which Lightning web component path allows a developer to view or edit a record while maintaining control over specifying its layout and set of fields?

Options:

- A- lightning-record-edit-form
- B- lightning-record-imperative
- C- lightning-record-view-form
- D- lightning-record-form

Answer:

A

Explanation:

The lightning-record-edit-form component allows a developer to view or edit a record while maintaining control over specifying its layout and set of fields. It is a wrapper component that accepts a record ID and an object API name, and displays fields with their labels and current values. The developer can use lightning-input-field components inside the lightning-record-edit-form to create editable fields, and lightning-output-field components or other display components to show read-only information. The lightning-record-edit-form component also implements Lightning Data Service, which means it does not require additional Apex controllers to create or update records, and it handles field-level security and sharing automatically.

Question 5

Question Type: MultipleChoice

A developer needs to make a call to a long running web service which is critical to finalizing their checkout process. Which three items should the developer consider in their implementation?

Options:

- A- A new CORS entry may need to be created in Setup
- B- A new Named Credential may need to be created in Setup
- C- An Apex method returning a Continuation will need to be created
- D- Requests to the service should be brokered to prevent limit exceptions
- E- A new Remote Site may need to be created in Setup

Answer:

A, B, C

Explanation:

To make a call to a long running web service, a developer should consider the following items in their implementation:

A new CORS entry may need to be created in Setup. CORS stands for Cross-Origin Resource Sharing, and it is a mechanism that allows web browsers to make requests to servers on different origins¹. A CORS entry defines a trusted origin for an external web service and allows the browser to access its resources². If the web service is not on the same origin as the Salesforce org, a CORS entry is required to avoid cross-origin errors³.

A new Named Credential may need to be created in Setup. A Named Credential specifies the URL of an external web service and its authentication settings⁴. It can also store certificates or secrets for secure communication⁴. By using a Named Credential, a developer can simplify the callout code and avoid hardcoding sensitive information⁵.

An Apex method returning a Continuation will need to be created. A Continuation is a class in Apex that allows making long-running requests to external web services without blocking the user interface⁶. It works by sending the request asynchronously and returning a placeholder response that can be used to resume the processing later⁶. A Continuation can handle requests that take up to 120 seconds, which is longer than the normal limit of 10 seconds for synchronous callouts⁷. Reference: Cross-Origin Resource Sharing (CORS), Create a CORS Whitelist Entry, Make Long-Running Callouts with Continuations, Named Credentials, Apex Web Services and Callouts, Continuation Class, Apex Governor Limits

Question 6

Question Type: MultipleChoice

Which technique can be used with Lightning web components to expose them outside of an org in another web container?

Options:

- A- Slot elements
- B- Heroku

C- Lightning Out

D- Lightning Canvas

Answer:

C

Explanation:

Lightning Out because this is a technique that allows developers to use Lightning web components outside of Salesforce servers, such as in a Node.js app running on Heroku or a department server inside the firewall². Lightning Out uses a standalone Aura dependency app to host the Lightning web components and expose them to an external web container². The other options are incorrect because they are not techniques for exposing Lightning web components outside of an org. Slot elements are a way to create reusable components with dynamic content³. Heroku is a cloud platform that supports various languages and frameworks, but it does not directly expose Lightning web components⁴. Lightning Canvas is a framework that enables developers to integrate third-party applications into Salesforce⁵, but it does not support Lightning web components⁶. Reference: Use Components Outside Salesforce, Slot Elements, Heroku, Lightning Canvas Developer's Guide, Lightning Web Components in Visualforce, Aura, and Lightning Out

Question 7

Question Type: MultipleChoice

Which code statement should a developer use to import the ID of the current Lightning 04m 235 Experience Site?

Options:

- A- import id from '@salesforce/network/Id'
- B- import id from '@salesforce/site/Id'
- C- import id from '@salesforce/experience/Id'
- D- import id from '@salesforce/community/Id'

Answer:

C

Explanation:

The correct answer is C. import id from '@salesforce/experience/Id' because this is the syntax for importing information about the current Experience Builder site from the @salesforce/experience scoped module. The id property returns the ID of the current site¹. The other options are incorrect because they use different scoped modules or properties that do not exist or do not return the site ID. Reference: Current Community

Question 8

Question Type: MultipleChoice

A developer has the task to create custom Lightning web components (LWCs). 04m 30s

Which two steps must a developer take when creating custom LWCs?

Options:

- A- ClonealWC.
- B- Create an Apex class.
- C- Deploy a custom component.
- D- Authorize an org for an SFDX project.

Answer:

C, D

Explanation:

To create custom Lightning web components, a developer needs to follow these steps:

Authorize an org for an SFDX project. This allows the developer to use Salesforce CLI commands to create, test, and deploy code to the org2.

Create a Lightning web component using Visual Studio Code or Salesforce CLI. A Lightning web component consists of a folder with three files: an HTML template file, a JavaScript file, and a metadata file3.

Deploy a custom component to the org using Visual Studio Code or Salesforce CLI. This makes the component available for use in the org4.

Optionally, test the component using Jest framework or browser tools4. Reference: Create Lightning Web Components, Build Lightning Web Components, Create a Hello World Lightning Web Component

Question 9

Question Type: MultipleChoice

What is the purpose of `connectedCallback` in a Lightning web component? "04m 385.

Options:

A- It performs actions when a network request is made.

- B-** It perform actions when a component makes a call to a Connect APL.
- C-** It performs actions when a component is removed from the DOM.
- D-** It performs actions when a component is added to the DOM.

Answer:

D

Explanation:

The `connectedCallback` is a lifecycle hook in Lightning web component. It fires when a component is inserted into the DOM. It runs once when the component is inserted. `connectedCallback` in Lightning Web Component flows from parent to child. It can be used to perform actions such as initializing variables, fetching data, or setting event listeners¹. Reference: `connectedCallback()` and `disconnectedCallback()`, `connectedCallback()` in Lightning Web Component, What is `connectedCallback` in LWC?

Question 10

Question Type: MultipleChoice

A developer needs to create an event listener on a parent component 04m 49s

programmatically. With the script below, what should replace the text

?

```
import { LightningElement } from 'lwc';
export default class Parent extends LightningElement {
  constructor() {
    super();
    <EVENT_LISTENER_LINE>
  }
  handleNotification = () => {};
}
```

Options:

- A- this.template.addEventListener(this.handleNotification);
- B- this.template.addEventListener('notification', this.handleNotification);
- C- this.template.addEventListener(handleNotification);
- D- addEventListener('notification', this.handleNotification);

Answer:

B

Explanation:

The correct answer is B. `this.template.addEventListener('notification', this.handleNotification);` because the event listener needs to be created on the parent component programmatically. The event listener should be added to the template of the parent component, and the event type should be 'notification'. The event listener should also have a callback function, which is `handleNotification` in this case. Reference: [Create a Lightning Web Component]

Question 11

Question Type: MultipleChoice

Which two scoped modules should a developer import in Lightning web components to check 04m 595 user permissions?

Options:

- A- `@salesforce/customPermission`
- B- `@salesforce/userPermission`

C- @salesforce/hasPermission

D- @salesforce/permission

Answer:

A, C

Explanation:

To check user permissions in Lightning web components, a developer can import two scoped modules: @salesforce/customPermission and @salesforce/hasPermission. The @salesforce/customPermission module allows the developer to access the custom permissions assigned to the current user. The @salesforce/hasPermission module allows the developer to check if the current user has a specific permission, such as View Setup or Customize Application. These modules can be used to conditionally render or enable components based on the user's permissions. Reference:

[Check User Permissions in Lightning Web Components]

[Custom Permissions]

[User Permissions and Access]

Question 12

Question Type: MultipleChoice

Northern Trail Outfitters (NTO) has a requirement to schedule a reusable data import across 05m 06s multiple orgs for customer demo purposes. NTO also has a requirement to seed data of related objects---ProductCatalog, ProductCategory, Product2, and ProductCategoryProduct--- while preserving its relationships and without purchasing additional licenses or using thirdparty tools. What is the recommended tool a developer should select to address the requirement?

Options:

- A- Dataloader.io
- B- Bulk Import Using Command Line Interface
- C- Commerce Product Data Import
- D- Data Import Wizard

Answer:

C

Explanation:

Commerce Product Data Import is a tool that allows developers to import product data from a CSV file into the B2B Commerce Cloud platform. It supports importing data for related objects, such as ProductCatalog, ProductCategory, Product2, and ProductCategoryProduct, while preserving their relationships. It also supports scheduling the import process to run at a specified time or frequency, which can be useful for demo purposes. Commerce Product Data Import does not require any additional licenses or third-party tools to use. Reference:

[Import Products](#)

[Commerce Product Data Import](#)

To Get Premium Files for B2B-Commerce-Administrator Visit

<https://www.p2pexams.com/products/b2b-commerce-administrator>

For More Free Questions Visit

<https://www.p2pexams.com/salesforce/pdf/b2b-commerce-administrator>

