MB-820 Q&As

Microsoft Dynamics 365 Business Central Developer

Pass Microsoft MB-820 Exam with 100% Guarantee

Free Download Real Questions & Answers **PDF** and **VCE** file from:

https://www.leads4pass.com/mb-820.html

100% Passing Guarantee
100% Money Back Assurance

Following Questions and Answers are all new published by Microsoft Official Exam Center

- **Instant Download** After Purchase
- **100% Money Back** Guarantee
- **365 Days** Free Update
- **800,000+** Satisfied Customers

QUESTION 1

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear on the review screen.

A company creates a Business Central app and a table named MyTable to store records when sales orders are posted.

Users report the following issues:

1.

The users receive permission errors related lo MyTable.

2.

Users are no longer able to post sales orders since installing the new app.

3.

The users cannot access the list page created in MyTable.

You need to resolve the user issues without creating new permission sets. You must use the principle of least privilege.

Solution: In the MyTable object add the property InherentPermissions = Rl. Does the solution meet the goal?

A. Yes

B. No

Correct Answer: B

The property InherentPermissions is used to automatically grant permissions to the table object it is applied to, but setting it to Rl (which seems to be a typo and should likely be \\'RL\\' for Read and Insert permissions) is not sufficient in this scenario. The issues reported by the users suggest that they need more than just read and insert permissions on MyTable. Since users are unable to post sales orders, they likely need Modify, Delete, or Execute permissions on certain tables or objects related to the sales order process. Additionally, the inability to access the list page created in MyTable could be due to lacking Read permissions on other related objects or pages. Therefore, merely setting InherentPermissions = RL on MyTable does not comprehensively address the users\\' permission issues, especially when considering the principle of least privilege. A more tailored approach to permissions, potentially involving adjustments to the app\\'s code or configuration to ensure proper permissions are applied where necessary, would be needed.

QUESTION 2

HOTSPOT

You have the following XML file sample for the Items list:

```
<Items>
  <Item No="1000">
    <Description>Table</Description>
  </Item>
  <Item No="1001">
    <Description>Chair</Description>
  </Item>
  <Item No="1002">
    <Description>Sofa</Description>
  </Item>
</Items>
```

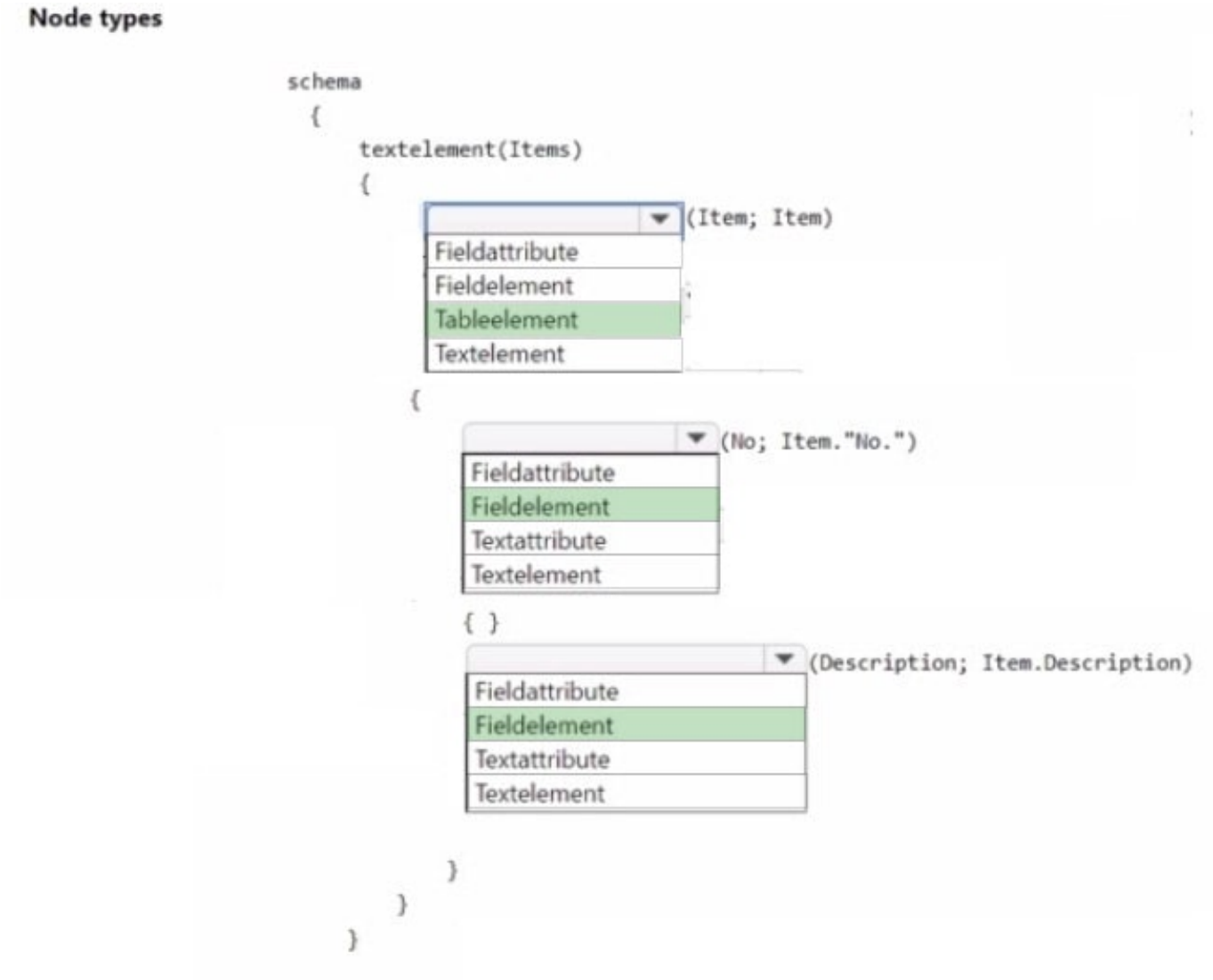
You plan to create the next XML file by using an XMLport object. You need to complete the code segment to export the file in the required format How should you complete the code segment? To answer, select the appropriate options in the answer area.

Hot Area:

Node types

```
schema
  {
      textelement(Items)
      {
          [Fieldattribute / Fieldelement / Tableelement / Textelement](Item; Item)
          {
              [Fieldattribute / Fieldelement / Textattribute / Textelement](No; Item."No.")
              { }
              [Fieldattribute / Fieldelement / Textattribute / Textelement](Description; Item.Description)
          }
      }
  }
```

Correct Answer:



QUESTION 3

You are creating an entitlement object in Business Central to enable transactability for AppSource apps.

You must map the entitlement object to a plan in Partner Center.

You need to select the value of the Type property to use in the entitlement object.

Which value should you use?

A. PerUserServicePlan

B. Implicit

C. Unlicensed

D. Role

Correct Answer: A

In Business Central, when creating an entitlement object to enable transactability for AppSource apps and mapping it to a plan in Partner Center, the Type property of the entitlement object should be set to PerUserServicePlan (A). The PerUserServicePlan type is used to define an entitlement that is based on a service plan, which is typically how transactability features are managed for apps distributed through AppSource. This type of entitlement allows for the mapping of specific features or capabilities of the app to a service plan in Partner Center, enabling granular control over what users are entitled to use based on their subscription. The other values, such as Implicit (B), Unlicensed (C), and Role (D), are used in different contexts and do not apply to the scenario of mapping an entitlement object to a plan for AppSource apps.

QUESTION 4

HOTSPOT

A company plans to customize its per tenant extension reports. The company has the following requirements for the customization:

1.

Child data items must not be displayed on the request page for some master detail reports.

2.

Selecting key filter fields takes users too much time. The customization must decrease the amount of time to select the fields.

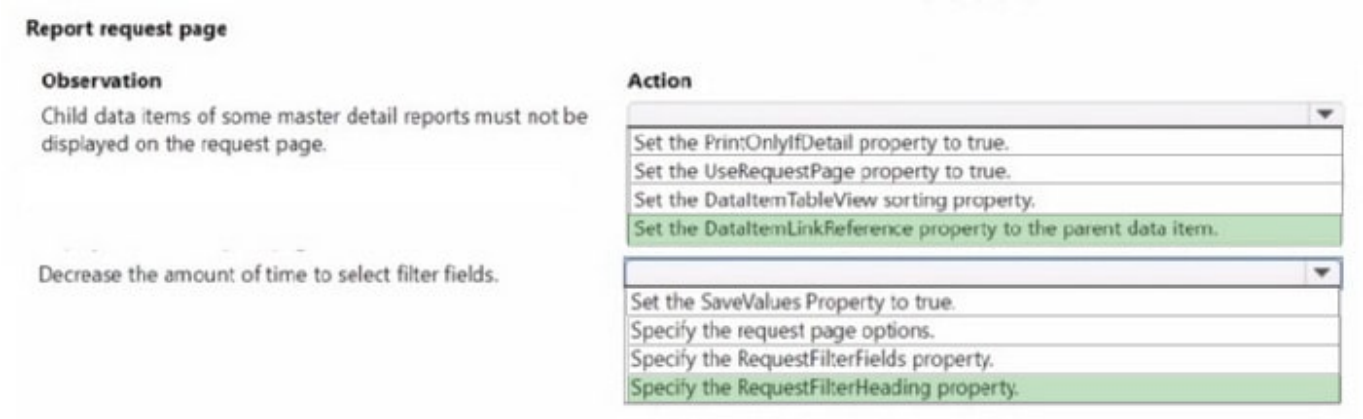
You need to optimize the report request page.

Which actions should you configure? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Correct Answer:



For the given requirements, you should configure the following actions:

For child data items not to be displayed on the request page for some master- detail reports, set the DataItemLinkReference property to the parent data item. To decrease the amount of time to select key filter fields, specify the

RequestFilterHeading property.

In Dynamics 365 Business Central, when customizing report request pages, certain properties can be set to control the behavior and display of the report options:

Hiding Child Data Items:The DataItemLinkReference property is used to link a child data item to a parent data item in the data model of a report. Setting this property correctly will ensure that the child data items are related to the correct

parent data item and will be displayed or hidden accordingly on the request page. If the goal is to prevent child data items from being displayed, you need to make sure they are correctly linked and configured to not appear. Optimizing Filter

Field Selection:The RequestFilterHeading property is used to group filter fields on the request page. By specifying this property, you can create a more organized and user-friendly interface, which can significantly speed up the process of

selecting filters. This property allows you to categorize filters into headings, making it quicker and easier for users to find and set the necessary filters for the report.

By adjusting these properties on the report request page as part of the per tenant extension customization, you will address the company\\'s requirements to optimize the user experience when running reports.

QUESTION 5

HOTSPOT

You need to write the code to call the subcontractor\\'s REST API.

How should you complete the code segment? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

REST services

```
procedure CallSubcontractorAPI(Url: Text[2048]; Username: Text[100]; Password:
Text[100]; Body: Text)
var
    httpClient: HttpClient;
    ResponseMessage: HttpResponseMessage;
    RequestHeaders, ContentHeaders: HttpHeaders;
    httpContent: HttpContent;
    Base64Convert: Codeunit "Base64 Convert";
    Response: Text;
begin
    RequestHeaders := httpClient.DefaultRequestHeaders();
    RequestHeaders.Add( [ ▼ ] , 'Basic '+ [ ▼ ] );
        'Authentication'
        'Authorization'
        Authorization
        Authentication
        'Credentials'
        Base64Convert.FromBase64(Username + ':' + Password)
        Base64Convert.ToBase64(Username + ':' + Password)
        Base64Convert.ToBase64(Username) + Base64Convert.ToBase64(Password)
        Username + ':' + Password
    httpContent.GetHeaders(ContentHeaders);
    ContentHeaders.Remove('Content-Type');
    ContentHeaders.Add('Content-Type', 'application/json'); [ ▼ ] ;
        httpContent := Body
        httpContent.Clear()
        httpContent.WriteFrom(Body)
    if [ ▼ ] then
        httpClient.Post(Url, httpContent)
        httpClient.Post(Url, httpContent, Response)
        httpClient.Post(Url, httpContent, ResponseMessage)
        httpClient.Send(Url, httpContent, ResponseMessage)
        ResponseMessage.Content.ReadAs(Response);
end;
```

Correct Answer:

REST services

```
procedure CallSubcontractorAPI(Url: Text[2048]; Username: Text[100]; Password:
Text[100]; Body: Text)
var
    httpClient: HttpClient;
    ResponseMessage: HttpResponseMessage;
    RequestHeaders, ContentHeaders: HttpHeaders;
    httpContent: HttpContent;
    Base64Convert: Codeunit "Base64 Convert";
    Response: Text;
begin
    RequestHeaders := httpClient.DefaultRequestHeaders();
    RequestHeaders.Add( [ 'Authentication' | 'Authorization' | Authorization | Authentication | 'Credentials' ] , 'Basic '+ [ Base64Convert.FromBase64(Username + ':' + Password) | Base64Convert.ToBase64(Username + ':' + Password) | Base64Convert.ToBase64(Username) + Base64Convert.ToBase64(Password) | Username + ':' + Password ] );
    httpContent.GetHeaders(ContentHeaders);
    ContentHeaders.Remove('Content-Type');
    ContentHeaders.Add('Content-Type', 'application/json'); [ httpContent := Body | httpContent.Clear() | httpContent.WriteFrom(Body) ] ;
    if [ httpClient.Post(Url, httpContent) | httpClient.Post(Url, httpContent, Response) | httpClient.Post(Url, httpContent, ResponseMessage) | httpClient.Send(Url, httpContent, ResponseMessage) ] then
        ResponseMessage.Content.ReadAs(Response);
end;
```

Box 1: \\'Authorization\\'

Business Central and the AL language have made web service code much easier with the HttpClient and Json types available. Handling the HTTP Authorization header is easier too with the TempBlob table, which can now encode the basic

authentication string using base64.

See below for an example of how to add a basic authorisation header to the AL HttpClient:

procedure AddHttpBasicAuthHeader(UserName: Text[50]; Password: Text[50], var HttpClient : HttpClient);

var

AuthString: Text;

TypeHelper: "Type Helper";

begin

AuthString := STRSUBSTNO(\\'%1:%2, UserName, Password);

AuthString := TypeHelper.ConvertValueToBase64(AuthString);

AuthString := STRSUBSTNO(\\'Basic %1\\', AuthString);

HttpClient.DefaultRequestHeaders().Add(\\'Authorization\\', AuthString);

end;

Box 2: Base64Convert.ToBase64(Username + \\':\\' + Passsword)

How To Connect To External APIs From Business Central (HTTP Request)

Authentication

There are several methods of authentication, basic, OAuth, etc. In the next example, you can see a way to use basic authentication.

We introduce 2 new types, HttpRequestMessage and HttpHeaders. They become fundamental when creating more complex API calls.

The way to build the call is quite simple:

Set the information for the message

Set the authorization in the header of the message

Send the request

Read and handle the response

Example:

Box 3: httpContent.WriteFrom(Body)

Example

The following example illustrates how to use the HttpContent type to send a simple POST request containing JSON data.

// Add the payload to the content content.WriteFrom(payload);

// Replace the default content type header with a header associated with this request content.GetHeaders(contentHeaders); contentHeaders.Clear(); contentHeaders.Add(\\'Content-Type\\', \\'application/json\\');

// Assigning content to request.Content will actually create a copy of the content and assign it.

// After this line, modifying the content variable or its associated headers will not reflect in

// the content associated with the request message

request.Content := content;

request.SetRequestUri(uri);

request.Method := \\'POST\\';

Box 4: httpClient.Post(Url,httpContent,ResponseMessage) POST as required, and include ResponseMessage.

Scenario: The API for sending subcontracting orders must be called by sending an authenticated POST request to the given endpoint.

Scenario, full:

An external business partner of Contoso, Ltd. exposed a REST API for receiving details about new subcontracting orders and for sending the planned release date of each subcontracting order received.

Integration with business partner for subcontracting

Contoso, Ltd. must connect Business Central to the external API provided by the business partner. This will be used for the partner to send the details of new subcontracting orders to fulfill the sales demand, and for receiving the planned

release date of each order sent. The integration requirements are as follows:

The business partner will provide a REST API secured with basic authentication. Credentials to access the API will be shared with Contoso, Ltd.

The API for sending subcontracting orders must be called by sending an authenticated POST request to the given endpoint.

The API for retrieving the order no. and planned release date of each subcontracting order responds with the following JSON:

Reference:

https://dankinsella.blog/http-basic-authentication-with-the-al-httpclient/ https://businesscentralgeek.com/how-to-connect-to-external-apis-from-business-central-http-request https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/methods-auto/httpcontent/httpcontent-data-type