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.geekcert.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

You are exporting data from Business Central.

You must export the data in a non-fixed length and width in CSV format.

You need to generate an XMLport to export the data in the required format

Which Format property value should you use?

A. XML

B. VariableText

C. FixedText

Correct Answer: B

When exporting data from Business Central and the requirement is for the data to be in a non-fixed length and width CSV format, the Format property of the XMLport should be set to VariableText (B). The VariableText format is designed for handling data exports where the fields are separated by a delimiter, such as a comma or tab, which is typical of CSV (Comma-Separated Values) files. This format allows for the flexibility needed when dealing with varying field lengths, as it does not enforce a fixed width for each field, making it ideal for CSV data exports. Setting the Format property to FixedText (C) would enforce a fixed width for each field, which is not suitable for CSV files, while the XML format (A) is used for exporting data in an XML structure, which is different from the CSV format requirements.

QUESTION 2

HOTSPOT

You plan to create a table to hold client data.

You have the following data integrity requirements:

1.

Lookups into other records must be established.

2.

Validate if a record exists in a destination record.

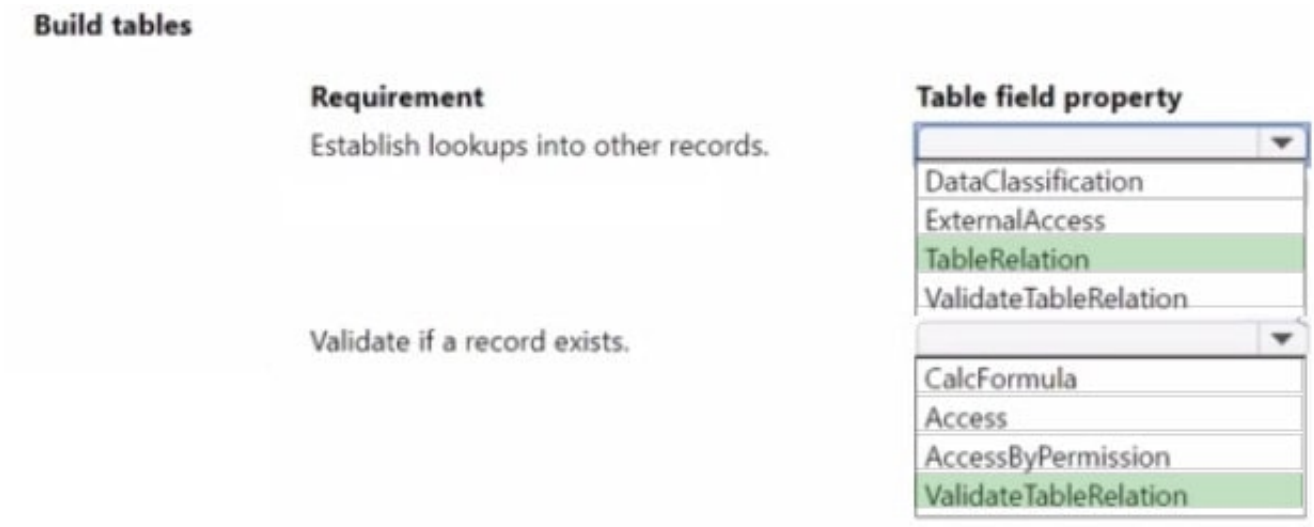
You need to select the table field property to use for each requirement.

Which table field property should you use? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Hot Area:

Correct Answer:



For the data integrity requirements, the table field properties to use are:

To establish lookups into other records, use the TableRelation property. To validate if a record exists in a destination record, use the ValidateTableRelation property.

In Business Central, when creating tables to hold data, maintaining data integrity is crucial:

TableRelation Property:This property is used to create a relationship between the field in one table and a field in another table, which is typically used for lookups. When you set the TableRelation property on a field, it allows users to select

from a list of values that exist in the related table.

ValidateTableRelation Property:This property is used to ensure that the value entered in a field matches one of the values in a related table. If a user tries to enter a value that doesn\\'t exist in the related table, an error will occur.

QUESTION 3

You create a page with the PageType property set to RoleCenter.

You navigate through the different sections of the page.

You need to add functionalities to the page.

What should you do?

A. Define actions in the area (reporting) before actions in the area (creation).

B. Define the navigation menu in the area(processing).

C. Define the navigation bar in the area (embedding).

D. Add a source table on the Role Center page.

Correct Answer: A

When creating a page with the PageType property set to RoleCenter in Microsoft Dynamics 365 Business Central, it\\'s essential to organize the functionalities and actions in a manner that enhances user experience and efficiency. The best practice is to define actions in the area (reporting) before actions in the area (creation) (A). This organization allows users to access reporting and analytical features quickly, which are commonly used in Role Centers for overview and insight purposes, before moving on to creation or transactional tasks. This logical flow aligns with typical user workflows, where analysis and review precede the creation of new records or transactions. The other options, such as defining the navigation menu in the area(processing) (B), defining the navigation bar in the area (embedding) (C), or adding a source table on the Role Center page (D), do not directly address the need to add functionalities to the Role Center page in a user- friendly manner.

QUESTION 4

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

QUESTION 5

You need to determine why the debugger does not start correctly.

What is the cause of the problem?

A. The "userId" parameter must have the GUID of the user specified, not the username.

B. The "breakOnNext" parameter is not set to "WebServiceClient".

C. The "userId" parameter is specified, and the next user session that is specified in the "breakOnNext" parameter is snapshot debugged.

D. The "executionContext" parameter is not set to "Debug".

Correct Answer: A

Initialize a snapshot debugging session

You can start a snapshot by creating a snapshot configuration file in Visual Studio Code.

Choose whether to run the session on a cloud service or locally. The configuration file contains the following information.

*

userId The GUID of the user who initiated the process to start snapshot debugging. For on-premises, this can also be the user name in user password authentication scenarios. The user must be able to start, or have a session type

opened that is specified in the breakOnNext parameter.

*

Etc.

Scenario: Debugging problems A user of the ISSUE BASE extension in Business Central reports a problem.

You discover that the debugging is not triggering.

Incorrect:

Not B: Example (attach to a web client session)

The following example illustrates a configuration for a local server, where you want to debug a web client session.

...

{

"name": "My attach to local server",

"type": "al",

"request": "attach",

"server": "https://localhost",

"serverInstance": "BC200",

"authentication": "Windows",

"breakOnError": true,

"breakOnRecordWrite": false,

"enableSqlInformationDebugger": true,

"enableLongRunningSqlStatements": true,

"longRunningSqlStatementsThreshold": 500,

"numberOfSqlStatements": 10,

"breakOnNext": "WebClient"

} ...

Reference: https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-attach-debug-next https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-snapshot-debugging