



Free Questions for MB-820 by dumpshq

Shared by Jimenez on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: DragDrop

A company plans to deploy Business Central.

The company has the following deployment requirements:

- * Use the company hardware architecture to run the deployment.
- * Use sandbox environments to develop extensions.
- * Allow tenants to connect to Shopify with the standard connector.
- * Use Microsoft Power Automate to create a workflow that calls a business event.

You need to identify the deployment type for each requirement.

Which deployment types should you use? To answer, move the appropriate deployment types to the correct requirements. You may use each deployment type once, more than once, or not at all.

Answer:

Question 2

Question Type: MultipleChoice

A company is deploying Business Central on-premises.

The company plans to use a single-tenant deployment architecture.

You need to describe how the data is stored and how the Business Central Server is configured.

In which two ways should you describe the single-tenant architecture? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

Options:

A- Each customer has their own Business Central Server.

- B-** The application and the business data are stored in the same database.
- C-** Multiple customers share a single Business Central Server.
- D-** The application and business data are stored in separate databases.
- E-** Multiple customers share multiple Business Central Server instances.

Answer:

B, D

Explanation:

In a single-tenant deployment architecture of Business Central on-premises, the following characteristics describe how the data is stored and how the Business Central Server is configured:

The application and the business data are stored in the same database (B): In a single-tenant architecture, each tenant (which typically corresponds to a single customer) has its own dedicated database. This database contains both the application objects (such as pages, reports, codeunits, etc.) and the business data (such as customer, vendor, and transaction records). This setup ensures that each tenant's data is isolated and can be managed independently.

The application and business data are stored in separate databases (D): While (B) is a characteristic of a single-tenant deployment, it's important to clarify that in some configurations, the application objects can be stored in a separate database from the business data. This approach can be used for easier maintenance and upgrades of the application code without affecting the business data. However, each tenant still has its own set of databases, maintaining the single-tenancy model.

The other options provided do not accurately describe a single-tenant architecture:

Each customer has their own Business Central Server (A): This statement might be misleading. In a single-tenant deployment, while each customer has their own database, they do not necessarily have their own Business Central Server instance. Multiple databases (tenants) can be hosted on a single server instance, although they are not shared across customers.

Multiple customers share a single Business Central Server (C) and Multiple customers share multiple Business Central Server instances (E): These options describe a multi-tenant architecture rather than a single-tenant one. In a multi-tenant setup, multiple customers (tenants) can share the same server instance and even the same application database, with data isolation ensured at the application level.

Question 3

Question Type: MultipleChoice

You are developing an app.

You plan to publish the app to Microsoft AppSource.

You need to assign an object range for the app.

Which object range should you use?

Options:

- A- custom object within the range 50000 to 59999
- B- custom object within the range 50000 to 99999
- C- divided by countries and use specific a country within the range 100000 to 999999
- D- an object range within the range of 7000000 to 74999999 that is requested from Microsoft
- E- free object within the standard range 1 to 49999

Answer:

D

Explanation:

When developing an app for Microsoft AppSource, it is essential to use an object range that is specifically designated by Microsoft to avoid conflicts with other apps and the base application. The correct object range to use is:

An object range within the range of 70000000 to 74999999 that is requested from Microsoft (D): This range is reserved for AppSource apps. Developers need to request this range from Microsoft to ensure that the objects used in their extension do not conflict with those used by other extensions or by the base application. Using this reserved range helps maintain the integrity and compatibility of extensions published on AppSource.

It's important to note that the other ranges mentioned (A, B, C, and E) are not suitable for apps intended for AppSource. Ranges 50000 to 59999 and 50000 to 99999 are typically reserved for per-tenant customizations or partner solutions, not for distribution on AppSource.

The standard range 1 to 49999 is reserved for the base application objects, and using an object range divided by countries (C) is not a standard practice for AppSource apps.

Question 4

Question Type: MultipleChoice

You are developing an app that will be published to Microsoft AppSource.

The app requires code analyzers to enforce some rules. You plan to add the analyzers to the settings.json file.

You need to activate the analyzers for the project.

Which three code analyzers should you activate to develop the app for AppSource? Each correct answer presents part of the solution

NOTE: Each correct selection is worth one point.

Options:

A- CodeCop

B- UICop

C- a custom rule set

D- PerTenantExtensionCop

E- AppSourceCop

Answer:

A, D, E

Explanation:

When developing an app for Microsoft AppSource, it is crucial to adhere to specific guidelines and standards to ensure compatibility and compliance. The three code analyzers you should activate are:

CodeCop (A): This is the default analyzer for AL language extensions. It enforces the AL Coding Guidelines, ensuring that the code follows best practices for readability, maintainability, and performance. It checks for a wide range of issues, from syntax errors to best practice violations, making it essential for any AL development.

PerTenantExtensionCop (D): This analyzer is specifically designed for extensions that are intended to be installed for individual tenants. It includes rules that ensure the extension does not interfere with the per-tenant customizations and adheres to the guidelines for extensions that can be safely installed and uninstalled without affecting the underlying application.

AppSourceCop (E): This analyzer is tailored for extensions that are intended for publication on Microsoft AppSource. It enforces additional rules that are specific to AppSource submissions, such as checking for the use of reserved object ranges and ensuring that all prerequisite dependencies are correctly declared. This is crucial for ensuring that your app meets all the requirements for listing on AppSource.

By activating these three analyzers, developers can ensure their app adheres to the standards required for AppSource, as well as maintain high code quality and compatibility with Business Central.

Question 5

Question Type: 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.

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 ' +  
    Base64Convert.FromBase64(Username + ':' + Password)  
    Base64Convert.ToBase64(Username + ':' + Password)  
    Base64Convert.ToBase64(Username) + Base64Convert.ToBase64>Password)  
    Username + ':' + Password );
```

```
  httpClient.GetHeaders(ContentHeaders);
```

```
  ContentHeaders.Remove('Content-Type');
```

```
  ContentHeaders.Add('Content-Type', 'application/json');
```

```
  if  
    httpClient.Post(Url, httpContent)  
    httpClient.Post(Url, httpContent, Response)  
    httpClient.Post(Url, httpContent, ResponseMessage)  
    httpClient.Send(Url, httpContent, ResponseMessage)  
  then  
    httpContent := Body  
    httpContent.Clear()  
    httpContent.WriteFrom(Body)
```

Question 6

Question Type: MultipleChoice

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

What is the cause of the problem?

Options:

- 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'.

Answer:

A

Explanation:

In Microsoft Dynamics 365 Business Central, when configuring snapshot debugging, it is crucial that the parameters in the configuration file are correctly set. From the options provided, the issue with the debugger not starting correctly is most likely due to an incorrect 'userId' parameter.

Option A is the cause of the problem. The 'userId' parameter must be the GUID of the user, not the username. The snapshot debugger needs the exact GUID to attach to the right session for debugging.

Option B is incorrect because 'breakOnNext' set to 'WebClient' is a valid setting. This tells the debugger to break on the next client action in the web client, which is a typical scenario.

Option C is not the cause of the problem. The 'userId' parameter is meant to specify which user session to debug, and this works in conjunction with the 'breakOnNext' parameter.

Option D is incorrect as the 'executionContext' parameter does not need to be set to 'Debug' for snapshot debugging to work. 'DebugAndProfile' is a valid value for the 'executionContext' parameter, as it allows for debugging and collecting performance information.

Therefore, the reason why the debugger does not start correctly is due to Option A: The 'userId' parameter must have the GUID of the user specified, not the username.

Question 7

Question Type: Hotspot

You need to create the API page according to the requirements.

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.

Code segment for API page

```
    page 50100 ItemAPI
    {
        PageType = API;
        Caption = 'ItemApi';
        APIPublisher = 'contoso';
        APIGroup = 'sales';
        APIVersion = 'v1.0';
        EntityName = 'item';
        EntitySetName = 'items';
        SourceTable = Item;
        ODataKeyFields = SystemId;
```

▼
InsertAllowed = false; DelayedInsert = true; DataAccessIntent = ReadOnly;
▼
ModifyAllowed = false; Editable = false; UsageCategory = Lists;

```
    layout
    {
        area(Content)
        {
            repeater(Groupname)
            {
                field(no; Rec."No.")
                {
                }
                field(description; Rec.Description)
                {
                }
                field(salesTotal; Rec."Sales (LCY)")
                {
                }
            }
        }
    }
}
```

Answer:

Question 8

Question Type: OrderList

You need to configure telemetry for the SaaS tenant and test whether the ingested signals are displayed.

Which three actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Actions

Select the Application Insights instance, select Logs and then inspect the Traces table.

Select the environment in the Admin Center and place the connection string in the Application Insights Connection String field.

Create an Azure Application Insights instance by using the Azure Portal in the Partner's subscription.

Create an Azure Application Insights instance by using the Azure Portal in the Customer's subscription.

Select the Application Insights instance, select Events, and then inspect the Traces table.

Select the Sessions menu and then select Restart Environment.



Steps to configure telemetry



Answer:

To Get Premium Files for MB-820 Visit

<https://www.p2pexams.com/products/mb-820>

For More Free Questions Visit

<https://www.p2pexams.com/microsoft/pdf/mb-820>

