

- **Vendor: Microsoft**
- **Exam Code: PL-400**
- **Exam Name: Microsoft Power Platform Developer**
- **New Updated Questions from [Braindump2go](#)**
- **(Updated in [December/2021](#))**

Visit Braindump2go and Download Full Version PL-400 Exam Dumps

QUESTION 176

You are creating a new page for a Power Apps portal.
You need to display data from Microsoft Dataverse on the page.
What should you use?

- A. Liquid
- B. CSS
- C. iFrame
- D. Bootstrap

Answer: A

Explanation:

Liquid is an open-source template language that is integrated natively into Microsoft Power Apps portals. It acts as a bridge between Dataverse and the HTML or text output that is sent to the browser. Liquid can be used to add dynamic content to pages and to create a variety of custom templates. Additionally, Liquid provides access only to the data and operations that are explicitly allowed by the portals.

Reference:

<https://docs.microsoft.com/en-us/learn/modules/liquid-template-language/>

QUESTION 177

Drag and Drop Question

You are creating a canvas app for a company. A security role has been created for sales representatives and a second security role has been created for sales managers.

The canvas app has the following requirements:

- Sales managers must be able to view the records of the salespeople in their business unit.
- Sales managers must be the only people who can view sales probability data in opportunity records.
- Sales representatives and new hires assigned to the same territory share access to sales records.

You need to assign permissions for the app.

Which security options should you use? To answer, drag the appropriate security options to the correct scenarios. Each security option may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

[PL-400 Exam Dumps](#) [PL-400 Exam Questions](#) [PL-400 PDF Dumps](#) [PL-400 VCE Dumps](#)

<https://www.braindump2go.com/pl-400.html>

Security options

Answer Area
Scenario

Sales managers must be able to view the records of the salespeople in their business unit.

Sales managers must be the only people who can view sales probability data in opportunity records.

Sales representatives and new hires assigned to the same territory share access to sales records.

Security option

Answer:

Security options
Answer Area
Scenario

Sales managers must be able to view the records of the salespeople in their business unit.

Sales managers must be the only people who can view sales probability data in opportunity records.

Sales representatives and new hires assigned to the same territory share access to sales records.

Security option

Explanation:

Box 1: Role-based security

Dataverse uses role-based security to group together a collection of privileges. These security roles can be associated directly to users, or they can be associated with Dataverse teams and business units.

Box 2: Field-level security

Sometimes record-level control of access is not adequate for some business scenarios. Dataverse has a field-level security feature to allow more granular control of security at the field level. Field-level security can be enabled on all custom fields and most system fields.

Box 3: Record-level security

Reference:

<https://docs.microsoft.com/en-us/power-platform/admin/wp-security-cds>

QUESTION 178

Drag and Drop Question

Technicians for a company use a model-driven app on their phones to record information about service visits. Users do not have permissions to the Power Apps maker portal to create or update apps.

Technicians report issues with the model-driven app. You are unable to reproduce the issues in a development environment.

You need to provide instructions to the technicians to gather more details about the errors.

Which four actions should you recommend be performed 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

Perform the steps to generate the errors and download the results from Monitor.

Open the app in a browser on the phone.

Open the application in a browser on a laptop computer when they return to the office.

Perform the steps to generate the errors while you monitor the technician's monitor debug session.

Add the following text to the end of the URL for the app: "&monitor=true"

Open the app on a phone by using Power Apps mobile.

Answer area



Answer:

Actions

Perform the steps to generate the errors while you monitor the technician's monitor debug session.

Open the app on a phone by using Power Apps mobile.

Answer area



Explanation:

Step 1: Open the app in a browser on the phone.

Step 2: Add the following text to end of the URL for the app: "&monitor=true"

You can start a Monitor session from a model-driven app. To do this, append &monitor=true to the end of the URL in the browser. This displays the Monitor command on the model-driven app global command bar. Select Monitor to open a monitoring session in a new tab.

Step 3: Perform the steps to generate the errors and download the results from Monitor.

Step 4: Open the application in a browser on a laptop computer when they return to the office

References:

<https://powerapps.microsoft.com/en-us/blog/monitor-now-supports-model-driven-apps/>

QUESTION 179

You deploy a Power Platform plug-in to a production environment. The plug-in code contains detailed tracing information. You are a member of the Environment Maker security role for the environment.

Users report unexpected results when they interact with confidential data by using the plug-in. You confirm that the plug-in works without errors in a development environment.

You need to investigate the root cause of the plug-in errors.

What should you do?

- A. Send a PUT request to enable plug-in tracing for the production environment.
- B. Send a GET request to retrieve the plugintracelogs records.
- C. Install Plug-in profiler in the production environment by using the Plug-in Registration tool.
- D. Reproduce and capture the errors, then debug from Visual Studio.

Answer: C

[PL-400 Exam Dumps](#) [PL-400 Exam Questions](#) [PL-400 PDF Dumps](#) [PL-400 VCE Dumps](#)

<https://www.braindump2go.com/pl-400.html>

Explanation:

Plug-in profiler is a solution that you can install on your environment that enables you to capture the execution context of a plug-in and then use that data to re-play the event within Visual Studio while debugging.

There are two tools available from which to run the Plug-in Profiler:

the Plug-in Registration Tool and Power Platform Tools for Visual Studio.

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/data-platform/tutorial-debug-plug-in>

QUESTION 180

Drag and Drop Question

You are creating a model-driven app.

Users need to see only the entities in the app navigation that are relevant to their role and their method of accessing the app.

You need to restrict entities on the sub-areas in the SiteMap.

Which properties should you use? To answer, drag the appropriate properties to the correct requirements. Each property may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.

NOTE: Each correct selection is worth one point.

Answer Area

Properties	Requirement	Property
Client	Ensure that the entity is visible only if the user can create records.	
Offline Availability	Ensure that the entity is visible only if the user is accessing the app with a web browser.	
Privileges		
SKUs		

Answer:

Answer Area

Properties	Requirement	Property
Client	Ensure that the entity is visible only if the user can create records.	Privileges
Offline Availability	Ensure that the entity is visible only if the user is accessing the app with a web browser.	SKUs

Explanation:

Box 1: Privileges

Privileges: This defines whether a subarea is displayed based on privileges available in any security roles that are assigned to the user.

Box 2: SKU

SKUs: Select the versions of Dynamics 365 that display this subarea.

Incorrect Answers:

Offline Availability: Select this check box to make this subarea available to users when they are offline in Dynamics 365 for Outlook.

Reference:

<https://docs.microsoft.com/en-us/powerapps/maker/model-driven-apps/create-site-map-app>

[PL-400 Exam Dumps](#) [PL-400 Exam Questions](#) [PL-400 PDF Dumps](#) [PL-400 VCE Dumps](#)

<https://www.braindump2go.com/pl-400.html>

QUESTION 181

Hotspot Question

You are training a group of makers to use Power Automate.

You have the following expressions:

Name	Expression
1	outputs('Get_Item').statusCode
2	"from": "@result('MyScope)'"

You need to identify what each expression is doing.

What does each expression do? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

Answer Area

Expression	Action
1	<div> <input type="text"/> </div> <div> <input type="text"/> Return the statuscode at runtime. <input type="text"/> Return the output to the statuscode at runtime. <input type="text"/> Return the Get_Item at runtime. </div>
2	<div> <input type="text"/> </div> <div> <input type="text"/> Return MyScope as all the action items. <input type="text"/> Return all the variables from all actions from MyScope. <input type="text"/> Return all the results from all actions from MyScope. </div>

Answer:

Answer Area

Expression	Action
1	<div> <input type="text"/> </div> <div> <input checked="" type="text"/> Return the statuscode at runtime. <input type="text"/> Return the output to the statuscode at runtime. <input type="text"/> Return the Get_Item at runtime. </div>
2	<div> <input type="text"/> </div> <div> <input type="text"/> Return MyScope as all the action items. <input type="text"/> Return all the variables from all actions from MyScope. <input checked="" type="text"/> Return all the results from all actions from MyScope. </div>

Explanation:

Box 1: Return the statuscode at runtime.

You could try the following method to get the status code.

Configure Compose action under the specified action to get the status code.

outputs('ActionName')['statusCode']

Box 2: Return all the results from all actions from MyScope

The @result() expression accepts the name of a Scope as a parameter and returns a JSON array of objects that

[PL-400 Exam Dumps](#) [PL-400 Exam Questions](#) [PL-400 PDF Dumps](#) [PL-400 VCE Dumps](#)

<https://www.braindump2go.com/pl-400.html>

represent the results of the execution of each action within the Scope.

Reference:

<https://powerusers.microsoft.com/t5/Building-Flows/How-to-determine-status-code-for-a-condition/td-p/355653>

<https://blogs.msmvps.com/windsor/2019/04/25/microsoft-flow-error-handling/>

QUESTION 182

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 once 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 in the review screen.

You are developing a model-driven app for a company.

When you create a new Account record, you must automatically display a form to collect data that is needed to create a Contact record. The form must switch to the appropriate form layout based on the contact type.

You open the Contact form by using JavaScript. You pass the contact type information to the form by using the `Xrm.Navigation.openForm` function. An `OnLoad` event handler in the Contact form processes the data and shows only the appropriate sections of the form for the given contact type.

You need to configure the receiving form to accept the data parameter.

Solution: In the form editor, add an event handler for the data parameter.

Does the solution meet the goal?

A. Yes

B. No

Answer: B

Explanation:

By default, model-driven apps allows a specified set of query string parameters to be passed to a form. You use these parameters to set default values when you create a new record in the application. Each parameter must use a standard naming convention that includes a reference to the column logical name.

There are two ways to specify which query string parameters will be accepted by the form:

Edit form properties

Edit form XML

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/model-driven-apps/configure-form-accept-custom-querystring-parameters>

QUESTION 183

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 once 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 in the review screen.

You are developing a model-driven app for a company.

When you create a new Account record, you must automatically display a form to collect data that is needed to create a Contact record. The form must switch to the appropriate form layout based on the contact type.

You open the Contact form by using JavaScript. You pass the contact type information to the form by using the `Xrm.Navigation.openForm` function. An `OnLoad` event handler in the Contact form processes the data and shows only the appropriate sections of the form for the given contact type.

You need to configure the receiving form to accept the data parameter.

Solution: In the form editor, add a query string parameter for the data parameter.

Does the solution meet the goal?

A. Yes

B. No

Answer: A

Explanation:

[PL-400 Exam Dumps](#) [PL-400 Exam Questions](#) [PL-400 PDF Dumps](#) [PL-400 VCE Dumps](#)

<https://www.braindump2go.com/pl-400.html>

By default, model-driven apps allows a specified set of query string parameters to be passed to a form. You use these parameters to set default values when you create a new record in the application. Each parameter must use a standard naming convention that includes a reference to the column logical name.

There are two ways to specify which query string parameters will be accepted by the form:

Edit form properties

Edit form XML

When you edit a form, on the Home tab in the Form group, select Form Properties. In the Form Properties dialog box, select the Parameters tab.

Use this tab to modify the names and data types that the form allows.

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/model-driven-apps/configure-form-accept-custom-querystring-parameters>

QUESTION 184

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 in the review screen.

You are developing a model-driven app for a company.

When you create a new Account record, you must automatically display a form to collect data that is needed to create a Contact record. The form must switch to the appropriate form layout based on the contact type.

You open the Contact form by using JavaScript. You pass the contact type information to the form by using the `Xrm.Navigation.openForm` function. An `OnLoad` event handler in the Contact form processes the data and shows only the appropriate sections of the form for the given contact type.

You need to configure the receiving form to accept the data parameter.

Solution: Export the solution, edit the customizations.xml, and add a `querystringparameter` element to the XML.

Does the solution meet the goal?

A. Yes

B. No

Answer: B

Explanation:

Within the exported solution customizations.xml file, immediately following the footer element, you can add a `<formparameters>` element. In the `<formparameters>` element, add `<querystringparameter>` elements to specify which parameters will be allowed.

Note: By default, model-driven apps allows a specified set of query string parameters to be passed to a form. You use these parameters to set default values when you create a new record in the application. Each parameter must use a standard naming convention that includes a reference to the column logical name.

There are two ways to specify which query string parameters will be accepted by the form:

Edit form properties

Edit form XML

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/model-driven-apps/configure-form-accept-custom-querystring-parameters>

QUESTION 185

Drag and Drop Question

You are developing a new Power Apps Component Framework (PCF) control.

The control must be deployed to a development environment by using the Power Apps CLI and a new solution.

You need to deploy the PCF control.

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

Actions**Answer area****Answer:****Actions****Explanation:**

Step 1: Create a solution

Create a new solutions project using the following command. The solution project is used for bundling the code component into a solution zip file that is used for importing into Dataverse.

```
pac solution init --publisher-name developer --publisher-prefix dev
```

Step 2: Add a solution reference to the project

Once the new solution project is created, refer the Solutions folder to the location where the created sample component is located. You can add the reference using the command shown below. This reference informs the solution project about which code components should be added during the build. You can add references to multiple components in a single solution project.

```
pac solution add-reference --path c:\downloads\mysamplecomponent
```

Step 3: Build the project and solution

To generate a zip file from the solution project, go into your solution project directory and build the project using the following command.

```
msbuild /t:build /restore
```

Step 4: Deploy the solution

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/component-framework/import-custom-controls>

QUESTION 186

Drag and Drop Question

You create solutions in a development environment and export the solution for testing by various departments in your organization. Power users in each department control the testing environments.

You must display department-specific wording at the beginning of any custom notifications that are displayed in testing environments.

You need to package solutions to ensure that the power users can customize the notification content.

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

Actions

Create an empty environment variable named **Custom Text Placeholder**.

Create a configuration page in the classic solution by using a text field named **Custom Text Placeholder** that uses the HTML file format.

Set the default value of the text field Custom Text Placeholder to **Enter custom text**.

Create a function to retrieve the value from the custom text placeholder and display the notification.

Export the solution.

Create a solution component configuration named **Custom Text Placeholder** that uses the JSON file format.

Answer area

⏪ ⏩

⏴ ⏵

Answer:

Actions

Create a configuration page in the classic solution by using a text field named **Custom Text Placeholder** that uses the HTML file format.

Set the default value of the text field Custom Text Placeholder to **Enter custom text**.

Create a solution component configuration named **Custom Text Placeholder** that uses the JSON file format.

Answer area

Create an empty environment variable named **Custom Text Placeholder**.

Create a function to retrieve the value from the custom text placeholder and display the notification.

Export the solution.

⏪ ⏩

⏴ ⏵

Explanation:

Step 1: Create an empty environment variable named Custom Text Placeholder.

Applications often require different configuration settings or input parameters when deployed to different environments. Environment variables store the parameter keys and values, which then serve as input to various other application objects. Separating the parameters from the consuming objects allows you to change the values within the same environment or when you migrate solutions to other environments.

Step 2: Create a function to retrieve the value from the custom text placeholder and display the notification.

Step 3: Export the solution

You can manually export solutions. Microsoft recommends that you create an unmanaged solution to use for exporting your customizations, and then export your customizations periodically so that you have a backup in case anything happens.

Reference:

<https://docs.microsoft.com/en-us/powerapps/maker/data-platform/environmentvariables>

<https://docs.microsoft.com/en-us/powerapps/maker/data-platform/export-solutions>