

## Microsoft

*PL-400*

*Microsoft Power Platform Developer*

- **Up to Date products, reliable and verified.**
- **Questions and Answers in PDF Format.**

### **Full Version Features:**

- **90 Days Free Updates**
- **30 Days Money Back Guarantee**
- **Instant Download Once Purchased**
- **24 Hours Live Chat Support**

### **For More Information:**

**<https://www.testsexpert.com/>**

- **Product Version**

# Latest Version: 18.2

## Question: 1

DRAG DROP

You need to select connectors for the app.

Which types of connectors should you use? To answer, drag the appropriate connectors to the correct requirements. Each connector 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.

Connectors	Requirement	Connectors
Create a custom connector.	View full registration records.	
Use an AppSource connector.	View customer names.	
Use a native application function.	View daily registrations.	
Create a connector with a Postman collection.		

**Answer:**

Requirement	Connectors
View full registration records.	Create a custom connector.
View customer names.	Use an AppSource connector.
View daily registrations.	Use a native application function.

Explanation:

Box 1: Create a custom connector

A custom connector is a wrapper around a REST API (Logic Apps also supports SOAP APIs) that allows Logic Apps, Power Automate, or Power Apps to communicate with that REST or SOAP API.

Box 2: Use an AppSource connector

You can only retrieve the Customer, UnifiedActivity, and Segments entities through the Power Apps connector. Other entities are shown because the underlying connector supports them through triggers in Power Automate.

Scenario: Customer information is stored in the Accounts entity.

Box 3: Use a native application function

You must produce a report that details the number of registrations for a day and send the report as a PDF to the management team.

Reference:

<https://docs.microsoft.com/en-us/connectors/custom-connectors/>

<https://docs.microsoft.com/en-us/dynamics365/customer-insights/audience-insights/export-powerapps>

## Question: 2

You need to determine the primary cause of the issue reported by interns when they use the app. What is the primary cause?

- A. Interns have the System Customizer security role but need the Environment Maker security role.
- B. Interns have the Common Data Service User security role but need the Environment Maker security role.
- C. Interns have the Environment Maker security role but need the Common Data Service User security role.
- D. Interns have the Environment Maker security role but need the System Customizer security role.
- E. Interns have the Environment Maker security role but need the Delegate security role.

**Answer: D**

Explanation:

Scenario: Interns can create apps but cannot interact with their own data.

Environment Maker role: Can create new resources associated with an environment, including apps, connections, custom APIs, gateways, and flows using Microsoft Power Automate. However, this role doesn't have any privileges to access data within an environment.

System Customizer role: full permission to customize the environment. However, users with this role can only view records for environment entities that they create.

Reference:

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

## Question: 3

You need to handle errors in UpdateRecord.js. Which code segment should you add at line UR06?

- A. 

```
catch(error) {  
  alert("Caught error: " + error.message);  
}
```
- B. 

```
Exception exception = Server.GetLastError();  
if(exception != null){
```
- C. 

```
catch(exception e){  
  console.writeline(e)}
```
- D. 

```
function (error){  
  console.log(error.message)}
```

**Answer: A**

Explanation:

<https://docs.microsoft.com/en-us/powerapps/developer/model-driven-apps/clientapi/reference/xrmwebapi/updaterecord>

## Question: 4

DRAG DROP

You need to address the user interface issues.

What should you do? To answer, drag the appropriate actions to the correct issues. Each action 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.

### Actions

Add `&ribbondebug=true` to the end of the application URL.

Export the XML file.

Modify the RibbonWSS.xsd file.

Use Ribbon Workbench.

### Requirement

Resolve rendering issue for New and Save buttons.

Add email button for registration form.

### Action

**Answer:**

### Requirement

Resolve rendering issue for New and Save buttons.

Add email button for registration form.

### Action

Add `&ribbondebug=true` to the end of the application URL.

Use Ribbon Workbench.

Explanation:

Box 1: Add `&ribbondebug=true` to the end of the application URL.

---

Scenario: The captions for the New and Save buttons do not render properly on the form. You can use the an in-app tool called the Command Checker to inspect the ribbon component definitions

to help us determine why the button is not rendered correctly.

To enable the Command Checker, you must append a parameter `&ribbondebug=true` to your D365 application URL. For example:

<https://yourorgname.crm.dynamics.com/main.aspx?appid=9ab590fcd25e-ea11-a81d-000d3ac2b3e6&ribbondebug=true>

Box 2: Use the Ribbon Workbench

Adding Buttons to Ribbons

Download and install Ribbon Workbench.

Select a suitable ICON for your button.

Create a solution.

Edit the button in Ribbon Workbench.

Publish and test.

Reference:

<https://support.microsoft.com/en-us/help/4552163/ribbon-troubleshooting-guide>

<https://neilparkhurst.com/2015/10/19/adding-buttons-to-ribbons/>

## Question: 5

You need to add the script for the registration form event handling. Which code segment should you use?

- A. `formContext.data.entity.addOnSave(myFunction)`
- B. `formContext.data.addOnLoad(myFunction)`
- C. `formContext.data.removeOnLoad(myFunction)`
- D. `addOnPreProcessStatusChange`
- E. `formContext.data.isValid()`

**Answer: B**

Explanation:

Scenario: Information about upcoming tournaments must be pre-located into the registration form when the registration form loads.

`addOnLoad` adds event handlers to the Subgrid OnLoad event event.

Reference:

<https://docs.microsoft.com/en-us/powerapps/developer/model-drivenapps/clientapi/reference/grids/gridcontrol/addonload>

For More Information – Visit link below:  
<https://www.testsexpert.com/>

16\$ Discount Coupon: **9M2GK4NW**

## Features:

■ Money Back Guarantee.....



■ 100% Course Coverage.....



■ 90 Days Free Updates.....



■ Instant Email Delivery after Order.....

