

# **Microsoft Certified: Power Platform Developer Associate – Skills Measured**

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

## **Exam PL-400: Microsoft Power Platform Developer**

### **Skills Measured**

#### **Create a technical design (10-15%)**

##### **Validate requirements and design technical architecture**

- design and validate the technical architecture for a solution
- design authentication and authorization strategy
- determine whether you can meet requirements with out-of-the-box functionality
- determine when to use Logic Apps versus Power Automate flows
- determine when to use serverless computing, plug-ins, or Power Automate
- determine when to build a virtual entity data source provider and when to use connectors

##### **Design solution components**

- design a data model
- design Power Apps reusable components
- design custom connectors
- design server-side components

##### **Describe Power Platform extensibility points**

- describe Power Virtual Agents extensibility points including Bot Framework skills and Power Automate flows
- describe Power BI extensibility points including Power BI APIs, custom visuals, and embedding Power BI apps in websites and other applications
- describe Power Apps portal extensibility points including CRUD APIs and custom styling

#### **Configure Common Data Service (15-20%)**

## **Configure security to support development**

- troubleshoot operational security issues
- create or update security roles and field-level security profiles
- configure business units and teams

## **Implement entities and fields**

- configure entity and entity options
- configure fields
- configure relationships and types of behaviors

## **Implement application lifecycle management (ALM)**

- create solutions and manage solution components
- import and export solutions
- manage solution dependencies
- create a package for deployment
- automate deployments
- implement source control for projects including solutions and code assets

## **Create and configure Power Apps (15-20%)**

### **Create model-driven apps**

- configure a model-driven app
- configure forms
- configure views
- configure visualizations

### **Create canvas apps**

- create and configure a canvas app
- implement complex formulas to manage control events and properties
- analyze app usage by using App Insights
- build reusable component libraries

### **Manage and troubleshoot apps**

- troubleshoot app issues by using Monitor and other browser-based debugging tools
- interpret results from App Checker and Solution Checker
- identify and resolve connector and API errors
- optimize app performance including pre-loading data and query delegation

## **Configure business process automation (5-10%)**

### **Configure Power Automate**

- create and configure a flow
- configure steps to use Common Data Service connector actions and triggers
- implement complex expressions in flow steps
- implement error handling
- troubleshoot flows by analyzing JSON responses from connectors

### **Implement processes**

- create and configure business process flows
- create and configure business rules
- create, manage, and interact with business process flows by using server-side and client-side code
- troubleshoot processes

## **Extend the user experience (10-15%)**

### **Apply business logic using client scripting**

- create JavaScript or Typescript code that targets the XRM API
- register an event handler
- create client-side scripts that target the Common Data Service Web API

### **Create a Power Apps Component Framework (PCF) component**

- describe the PCF component lifecycle
- initialize a new PCF component
- configure a PCF component manifest
- implement the component interfaces
- package, deploy, and consume the component
- configure and use PCF Device, Utility, and WebAPI features
- test and debug PCF components by using the local test harness

### **Create a command button function**

- create the command function
- design command button rules and actions
- edit the command bar by using the Ribbon Workbench
- manage dependencies between JavaScript libraries

## **Extend the platform (15-20%)**

### **Create a plug-in**

- describe the plug-in execution pipeline
- design and develop a plug-in
- debug and troubleshoot a plug-in

- implement business logic by using pre and post images
- perform operations on data by using the Organization service API
- optimize plug-in performance
- register custom assemblies by using the Plug-in Registration Tool
- develop a plug-in that targets a custom action message

### **Create custom connectors**

- create a definition for the API
- configure API security
- use policy templates to modify connector behavior at runtime
- expose Azure Functions as custom connectors
- create custom connectors for public APIs by using Postman

### **Use platform APIs**

- interact with data and processes by using the Common Data Service Web API or the Organization Service
- implement API limit retry policies
- optimize for performance, concurrency, transactions, and batching
- query the Discovery service to discover the URL and other information for an organization
- perform entity metadata operations with the Web API
- perform authentication by using OAuth

### **Process workloads**

- process long-running operations by using Azure Functions
- configure scheduled and event-driven function triggers in Azure Functions
- authenticate to the Power Platform by using managed identities

## **Develop Integrations (5-10%)**

### **Publish and consume events**

- publish an event by using the API
- publish an event by using the Plug-in Registration Tool
- register service endpoints including webhooks, Azure Service Bus, and Azure Event Hub
- implement a Common Data Service listener for an Azure solution
- create an Azure Function that interacts with Power Platform

### **Implement data synchronization**

- configure entity change tracking
- read entity change records by using platform APIs
- create and use alternate keys