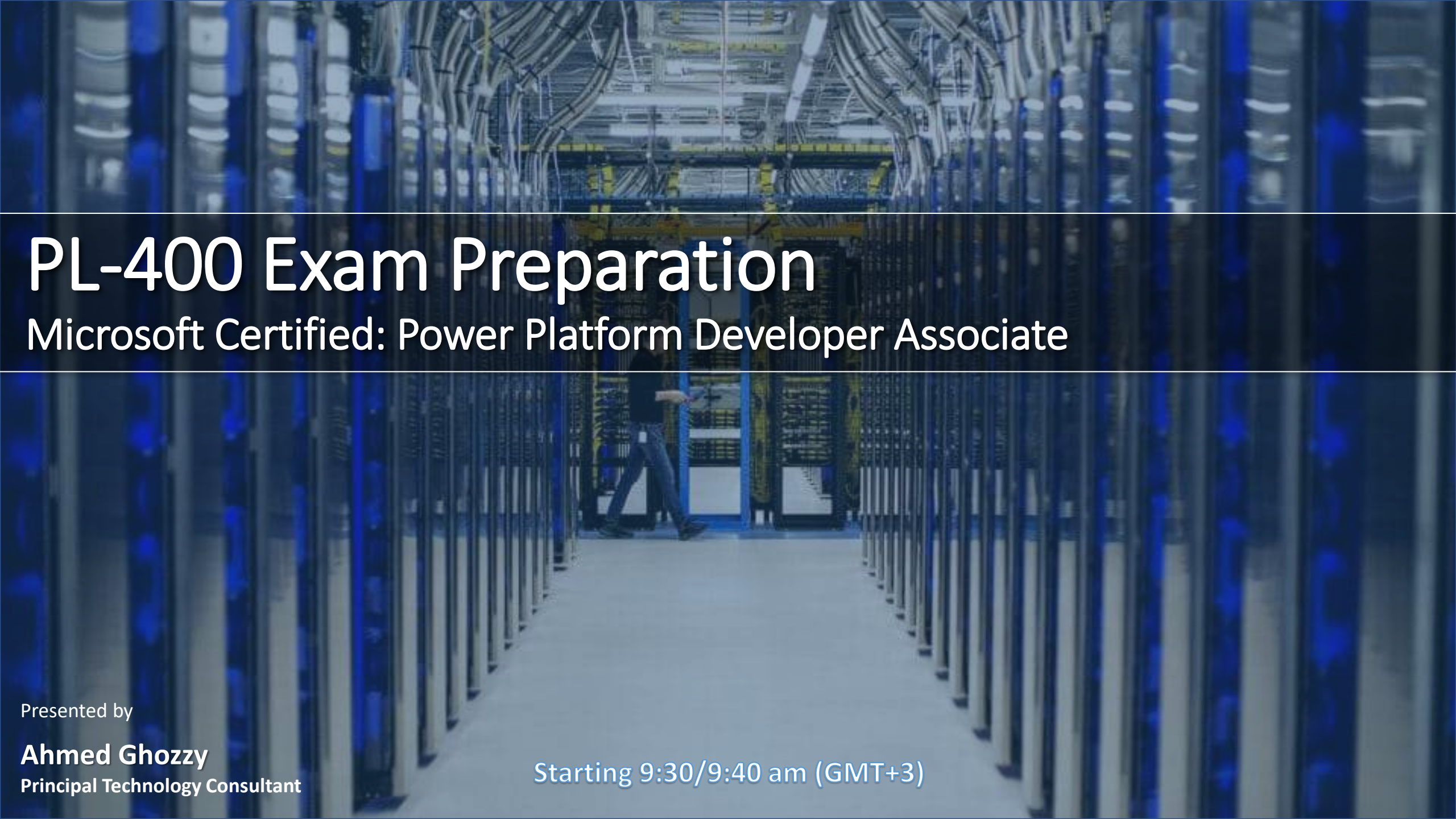


PL-400 Exam Preparation

Microsoft Certified: Power Platform Developer Associate

Presented by

Ahmed Ghozzy

Principal Technology Consultant

Starting 9:30/9:40 am (GMT+3)

Agenda

- Understanding the Exam Guide
- Setting up Personal Preparation Plan
- Key Topics/Questions Points on Technical Designs & Dataverse
- Key Topics/Questions Points on Creating & Configuring Power Apps
- Key Topics/Questions Points on Configuring Power Automate Flows
- Key Topics/Questions Points on Extending User Experience
- Key Topics/Questions Points on Extending The Platform
- Key Topics/Questions Points on Integration Development
- Briefing on Official Practice Tests
- Useful Resources
- Q&A (Kindly keep away un-official exam dumps questions & discussions)

Understanding the Exam Guide

Exam PL-400: Microsoft Power Platform Developer

- **Skills measured**
  - Create a technical design (10—15%)
  - Configure Microsoft Dataverse (10—15%)
  - Create and configure Power Apps (5—10%)
  - Configure business process automation (5—10%)
  - Extend the user experience (15—20%)
  - Extend the platform (20—25%)
  - Develop integrations (5—10%)

Exam PL-400: Microsoft Power Platform Developer - Certifications | Microsoft Learn

Setting up Personal Preparation Plan

Plan & Prepare

- Assess the exam objectives & outlines, Highlight in different color:
  - Topics you know very well.
  - Topics you partially know but needs more learning.
  - Topics totally new to you.
- Outcome -> Gap Understood & Extra Learning Efforts Assessed.
- Learn more from MOC PL400 and Microsoft Docs.
- If Budget allows, Take Official Practice Test.
- Do NOT use ~~unofficial exam dumps~~, do NOT trust answers you find there.

Key Points on
Technical Designs & Dataverse

Identify Different Azure AD Authentication Methods

- Azure AD
  - Uses Azure AD Identity
  - Intended for Internal users / employees
- Azure AD B2B [B2B collaboration overview - Azure AD - Microsoft Entra | Microsoft Docs]
  - Uses Partner Tenant Azure AD or 3rd Party Identity Providers
  - Intended for Business Partners to be “Invited as Guests” and share access to the tenant.
- Azure AD B2C [What is Azure Active Directory B2C? | Microsoft Docs]
  - Uses Customer Tenant Azure AD or 3rd Party Identity Providers
  - Intended for Customers/Consumers to be authenticated and access application resources (i.e. PowerApps Portals)

Internal / External Access

- Model Driven & Canvas Apps are typically designed for internal user access, and a license is required.
- Portal Apps can be used for internal & external access, users/customers accessing the portal are not necessarily holding license and can also access the portal anonymously or authenticated.
- Canvas Apps can allow external access to “Guest Users” using Azure B2B
  - If the App is connecting to Dataverse, Power Apps License is required.

Share a canvas app with guest users (contains video) - Power Apps | Microsoft Docs

Overview of authentication in Power Apps portals - Power Apps | Microsoft Docs

Access Microsoft Graph API

- Microsoft Graph is a RESTful web API that enables you to access Microsoft Cloud service resources.
- Microsoft Graph API offers access to centralized wide range of Data from Microsoft 365 and its services.
- For Apps to Access the API, they need to be Registered in AAD, and permissions granted.
- Registering an App in AAD involve creating a unique App ID and a Secret (or Cert) for App authentication vs AAD
- Permissions will let admins to grant access to Graph API to the app.
- When Users access the app (which subsequently authenticate to AAD to access Graph API), users may need to “consent” for the App permission.
- Admins can “Consent” on behalf of all organization users.
- Connecting from Power Platform to MS Graph API requires a **“Custom Connector”** (Note: new Microsoft Graph Security Connector is in “Preview”)

Use the Microsoft Graph API - Microsoft Graph | Microsoft Learn

Register your app with the Azure AD v2.0 endpoint - Microsoft Graph | Microsoft Learn

Creating Microsoft Graph custom connector in Power Apps - Microsoft Tech Community

Using Azure Functions

- Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs.
- **Use your preferred language**: Write functions in C#, Java, JavaScript, PowerShell, or Python, or use a custom handler to use virtually any other language.
- **Automate deployment**: From a tools-based approach to using external pipelines, there's a myriad of deployment options available.
- **Troubleshoot a function**: Use monitoring tools and testing strategies to gain insights into your apps. Azure Functions can run/test locally.
- **Flexible pricing options**: With the Consumption plan, you only pay while your functions are running, while the Premium and App Service plans offer features for specialized needs.

Azure Functions Overview | Microsoft Learn

Deciding an Automation Option

- **Business Rules**
  - Good for simple validation or setting of values
  - Optimized to run as part of the transaction for modifications that occur on the rows
  - No ability to use related rows or connectors
  - Can also be configured to run in model-driven apps for basic UX operations like hide/show
    Available scope- single form, all forms, table
- **Classic Workflows**
  - Primary scenario is when real-time processing is required
  - Power Automate should be the first choice for background operations
- **Power Automate**
  - Primary choice for non-real time automation
  - Supports near real-time triggering from Microsoft Dataverse
  - Support for connector and UI automation and Business Process Flows
- **Plug-in**
  - Custom logic is an extension of the Microsoft Dataverse operation
  - Ability to modify the request and response on the fly
  - Able to handle complex logic
  - Requires code developer skills
  - Can be either synchronous or asynchronous

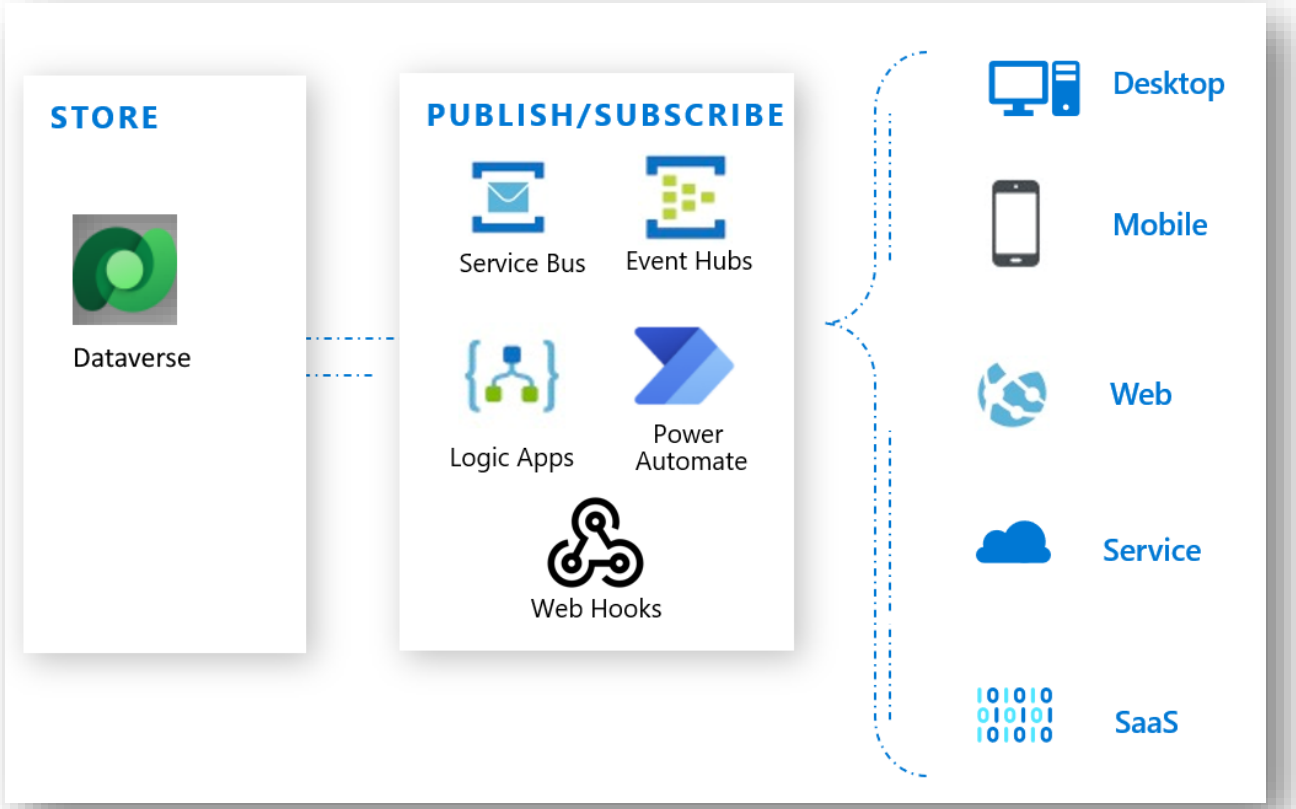
Create model-driven app business rules and recommendations - Power Apps | Microsoft Learn

Enable Power Automate integration to automate processes - Power Platform | Microsoft Learn

Use plug-ins to extend business processes (Microsoft Dataverse) - Power Apps | Microsoft Learn

Integrating with apps by using events

- Microsoft Dataverse provides multiple ways to integrate in any type of app (mobile, web, desktop), device, system, or service.
- A common approach of app integration is through the use of events. For example, an event such as adding a new row occurs in Dataverse, and this should be communicated to an associated system so that an action can be taken.
- **Webhooks** let you send data about events that occur on Dataverse to a web app by using a lightweight HTTP pattern for connecting web APIs and services with a publish-and-subscribe model.
  - Webhooks can only scale to the point at which your hosted web service can handle the messages.
  - Webhooks enable synchronous and asynchronous steps.
  - Webhooks send POST requests with the JSON payload and can be consumed by any programming language or web app hosted anywhere.



[Work with any type of app - Power Apps | Microsoft Learn]

[Use Webhooks to create external handlers for server events (Microsoft Dataverse) - Power Apps | Microsoft Learn]

Portals Elements Comparison

- **Table Permission vs Page Permission**

  Configure table permissions using portals Studio - Power Apps | Microsoft Docs
  Manage page permissions - Power Apps | Microsoft Docs

  - Table Permissions control access to individual records
    - CRUD
    - can be scoped Global (all records), Contact (only his records), Account (only his company records) etc.
  - Page Permissions control access to web pages
    - Allow/Disallow Anonymous access
    - Allow/Disallow certain roles access
- **Security Roles vs Web Roles**

  Create web roles for portals - Power Apps | Microsoft Docs

  - Security Roles control access for “internal organization users” using Model Driven Apps or Canvas Apps.
  - Web Roles control access for Portal Users and typically associated with Table Permission and Page Permissions.
- **Web Templates vs Page Templates**

  Create and manage page templates - Power Apps | Microsoft Docs
  Store source content by using web templates - Power Apps | Microsoft Docs

  - Web Template define the “Layout” to be used and typically include HTML/Liquid to define columns layout, navigation, breadcrumbs, etc.
  - Page Templates are “starting point” to create Web Pages, they drive how the web page is rendered, and typically bridge that to either a “Web Template” (with Liquid) or “ASPX” page.
- **Basic Forms vs Advanced Forms**

  About basic forms - Power Apps | Microsoft Docs
  Define advanced form properties for portals - Power Apps | Microsoft Docs

  - Basic Forms facilitate to a user to create or update a record
  - Advanced Forms behave like multi-stage data entry, more like a wizard.

Microsoft Bot Framework Skills

- A skill is a bot that can perform a set of tasks for another bot.
  - A skill's interface is described by a manifest. Developers who don't have access to the skill's source code can use the information in the manifest to design their skill consumer (potentially another Interactive Bot).
  - A skill can use claims validation to manage which bots or users can access it (AllowedCallers Array).
- For Bots to be able to call your Microsoft Bot Framework Skill
  - Skill needs to allow your bot to call it in AllowedCallers, You Bot ID is required.
  - Your Bot need to understand the skill and how to call it by pointing it to the skill manifest URL

Configure Bot Framework skills - Power Virtual Agents | Microsoft Learn

Use Microsoft Bot Framework skills - Power Virtual Agents | Microsoft Learn

Using Dataverse Custom APIs

- Use Custom APIs to create your own APIs in Dataverse. You can consolidate one or more operations into a Custom API that you and other developers can call in their code or from Power Automate.
- A Custom API may include logic implemented with a plug-in.
- Invoking Custom APIs from the Dataverse Web API using:
  - Unbound/Bound Actions – Actions that are bound/unbound to Dataverse table
    Can be called using POST if they accept input parameters
  - Unbound/Bound Functions – Functions to retrieve data without causing an effect (unlike actions)
    Called with GET requests.

Create and use Custom APIs (Microsoft Dataverse) - Power Apps | Microsoft Learn

Connecting to External Data

- Power Platform Components (i.e. Apps or Flows) can connect to external data using verity of options depending on the component type.
- Two Important Approaches
  - Virtual Tables
    - Enable the integration of data residing in external systems by seamlessly representing that data as tables in Microsoft Dataverse, without replication of data and often without custom coding.
    - Virtual Tables uses Data Providers (Dataverse include ODATA and Azure Cosmos DB)
    - Custom Data Providers can be developed & used
    - All tables in the external data source must have an associated GUID primary key.
      - Some other Limitation exist.
  - Custom Connectors
    - A custom connector is a wrapper around a REST API (also SOAP if used in Logic Apps)
    - Requires Authentication Type (None, Basic, API Key, Oauth), OpenAPI Definition, & Acceptable HTTP Verbs
    - API can be Publicly Accessible – Hosted as Azure Function, API App, or Web App
    - API can be Privately Accessible – using On-Premises Data Gateway
  - Canvas Apps
    - Canvas App can access wide variety of sources using Power Platform Connectors
    - They can be “embedded” in Model-Driven Apps, Portal Apps, and Microsoft Teams

Get started with virtual tables (entities) (Microsoft Dataverse) - Power Apps | Microsoft Learn

Custom connectors overview | Microsoft Learn

Embed a canvas app on a model-driven form - Power Apps | Microsoft Learn

Canvas Apps & Components

- Components “in” Canvas Apps
  - Suitable for reuse only within the same app.
  - If a need to reuse with another app, it must be exported and imported, with lack of control on future version updates.
- Components Library
  - Great for reusability between different Canvas Apps, with better change/update notifications to Apps owner to update to a new version.
- PCF Components
  - Great option but scope it to complex and advanced needs.
  - Requires code development in Typescript, HTML, and CSS.

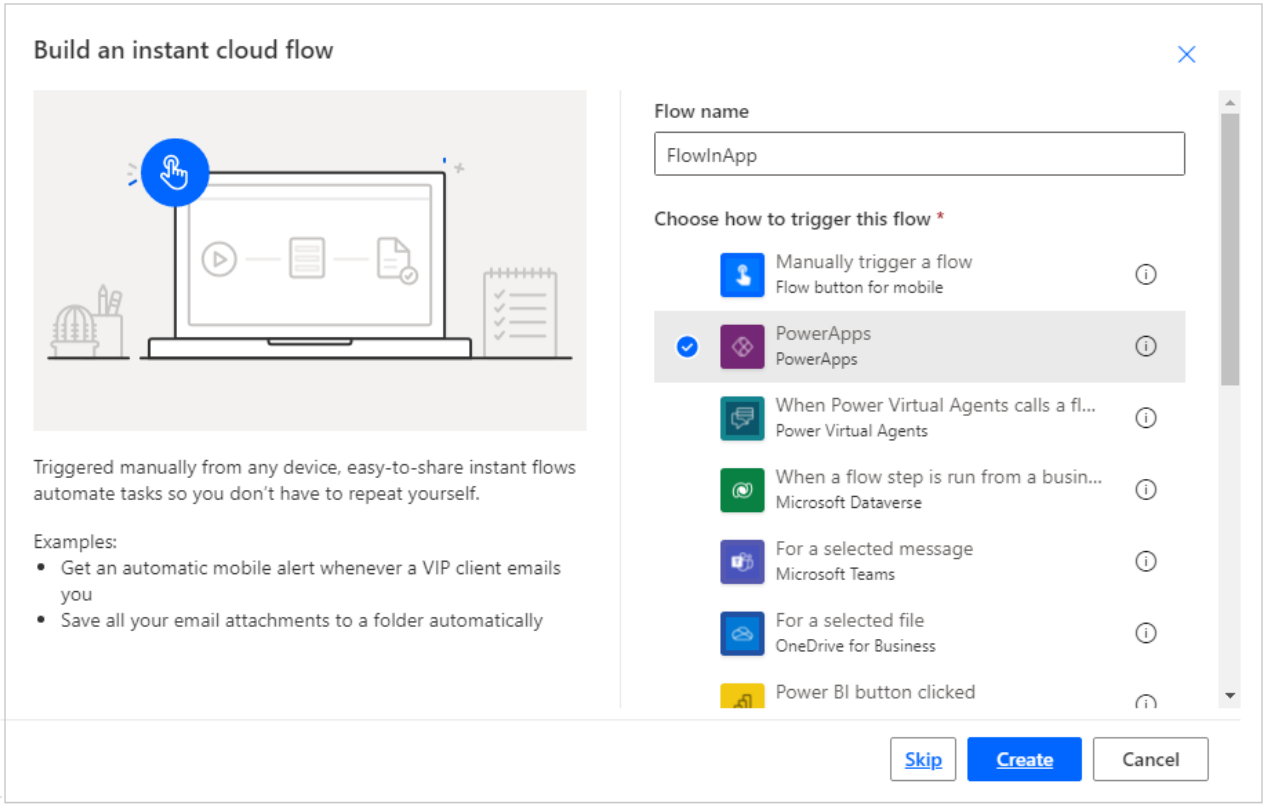
Canvas component overview (contains video) - Power Apps | Microsoft Docs

Component library - Power Apps | Microsoft Docs

Canvas Apps to Call Power Automate Flows

- Flow needs to be Instant Flow with Trigger “Power Apps”
- Flow may include Parameters (Ask in Power App) to be passed by the app
- Flow may end with action “Respond to Power App” to pass back output
- Canvas App Call, Pass input, collect output like below:

```
=  fx  Set(SalesInfoVariable,GetSalesInfo.Run())
```



[Start a flow in a canvas app - Power Apps | Microsoft Docs]

Relationship Behaviors

Behavior	Description
Cascade Active	Perform the action on all active related table rows.
Cascade All	Perform the action on all related table rows.
Cascade None	Do nothing.
Remove Link	Remove the lookup value for all related rows.
Restrict	Prevent the primary table row from being deleted when related table rows exist.
Cascade User Owned	Perform the action on all related table rows owned by the same user as the primary table row.

Column	Options
Assign	Cascade All Cascade Active Cascade User-owned Cascade None
Reparent	Cascade All Cascade Active Cascade User-owned Cascade None
Share	Cascade All Cascade Active Cascade User-owned Cascade None
Delete	Cascade All Remove Link Restrict
Unshare	Cascade All Cascade Active Cascade User-owned Cascade None

[About table relationships for Microsoft Dataverse - Power Apps | Microsoft Learn]

Dataverse Security Capabilities

- Security Roles

  Security roles and privileges - Power Platform | Microsoft Docs

  - Performs Row Level Security
  - Users/Teams can hold multiple roles, effectively combining all permissions on them.
- Column Security Profiles

  Field-level security - Power Platform | Microsoft Docs

  - Performs Column Level Security
  - Can NOT be linked/associated with Security Roles
  - Associated with a User or Team.
  - Don’t affect Rollups Fields based on Secure Columns
- Position Hierarchy
  - Allow users to read/write rows for their direct reports across Business Units (ignores Business Units structure)

  Hierarchy security - Power Platform | Microsoft Docs
- Manager Hierarchy
  - Similar to Position Hierarchy, but observes Business Unit structure (manager must be in same or parent business unit)

Environment Security Roles

- Security roles can be used to configure environment-wide access to all resources in the environment, or to configure access to specific apps and data in the environment.
- An environment can have zero or one Dataverse database, available roles can be different based on Dataverse presence.
- Key Roles:
  - Environment Admin
  - Environment Maker
  - System Administrator
  - System Customizer
  - Basic User

Configure user security in an environment - Power Platform | Microsoft Docs

Security roles and privileges

- Ensure you understand the difference between different privileges in security roles.
- Be aware of some know tricks to combine some privileges like:
  - To Allow Creating records, Create + Read is needed together.
  - To relate a child record (B) to a parent record (A) use:
    - Read + Write + “Append” on Table B (Write is needed to edit, Append is needed to relate)
    - Read + “Append To” on Table A (Read is needed to list, Append To is needed to relate “to”)

Security roles and privileges - Power Platform | Microsoft Docs

Model Driven Apps Forms

- **Main**
  - Provide the main user interface for viewing and interacting with table data.
- **Quick Create**
  - Provide a basic form optimized for creating new records.
- **Quick View**
  - Appear within the main form to display additional data for a parent row that is referenced by a lookup column in the form.
  - Users can view data from related tables without having to leave the form.
- **Card**
  - Used in views. Card forms are designed to present information in a compact format that is suitable for mobile devices.

- Forms can be secured by Security Roles, only eligible Users can access the Form

Type of model-driven app forms in Power Apps - Power Apps | Microsoft Docs

Control access to forms in Dynamics 365 Customer Engagement (on-premises) | Microsoft Learn

Automating ALM using GitHub Actions

- GitHub Actions enable developers to build automated software development lifecycle workflows.
- You can create workflows in your repository to build, test, package, release, and deploy apps; perform automation; and manage bots and other components built on Microsoft Power Platform.
- Connection to environments
  - **Username/password:** Configured as a generic service connection with username and password. Username/password authentication doesn't support multifactor authentication.
  - **Service principal and client secret:** This connection type uses service principal–based authentication and supports multifactor authentication. Service principal authentication
- Solution tasks
  - import-solution, export-solution, unpack-solution, pack-solution, publish-solution, more

Available GitHub Actions for Microsoft Power Platform development - Power Platform | Microsoft Learn

GitHub Actions for Microsoft Power Platform - Power Platform | Microsoft Learn

Understand Automation & Dev Tools

- **Power Platform VS Code Extension** Use the Visual Studio Code extension - Power Apps | Microsoft Docs
  adds the capability to configure portals using VS Code, and use the built-in Liquid language IntelliSense enabling help with code completion, assistance, and hinting while customizing portals interface using VS Code.
- **Power Platform Build Tools** Microsoft Power Platform Build Tools for Azure DevOps - Power Platform | Microsoft Docs
  For Azure DevOps to automate common build and deployment tasks related to apps built on Microsoft Power Platform
- **Power Platform CLI** Microsoft Power Platform CLI - Power Apps | Microsoft Docs
  Simple, one-stop developer CLI that empowers developers and ISVs to perform various operations in Microsoft Power Platform related to environment
- **Configuration Migration Tool** Configuration Migration tool - Power Platform | Microsoft Docs

  Used to move configuration data across environments. Configuration data is used to define custom functionality and is typically stored in custom tables. This tool is not designed to move business data.
- **Package Deployer** Package Deployer tool - Power Platform | Microsoft Docs

  Used to deploy packages on Dataverse environments. A package is an installable unit that includes solutions.
- **Solutions Packager** SolutionPackager tool (Microsoft Dataverse) - Power Platform | Microsoft Docs

  A tool that can reversibly decompose a Dataverse compressed solution file into multiple XML files and other files so that these files can be easily managed by a source control system.
- **XrmToolBox** Community tools for Microsoft Dataverse (Dataverse) - Power Apps | Microsoft Docs

  Windows application that connects to Dataverse, providing tools to ease customization, configuration and operation tasks. It is shipped with more than 30 plugins to make administration, customization or configuration tasks easier and less time consuming. “Document Template Mover” plugin can be used to move Word Templates between environments instead of recreating them.

Unmanaged vs Managed Solutions

- Use Unmanaged when:
  - Deploying to Dev or Test Environment
  - Reliable rollback is not needed (Removing Solution doesn’t remove customization)
  - You want to allow users in other environments to edit the solution
- Use Managed when:
  - Deploying to Production and Critical Environment
  - Reliable rollback is needed (Removing Solution removes customization & related data)
  - You want to limit customization of the solution.

Solution concepts - Power Platform | Microsoft Docs

Updating vs Upgrading Solutions when Importing

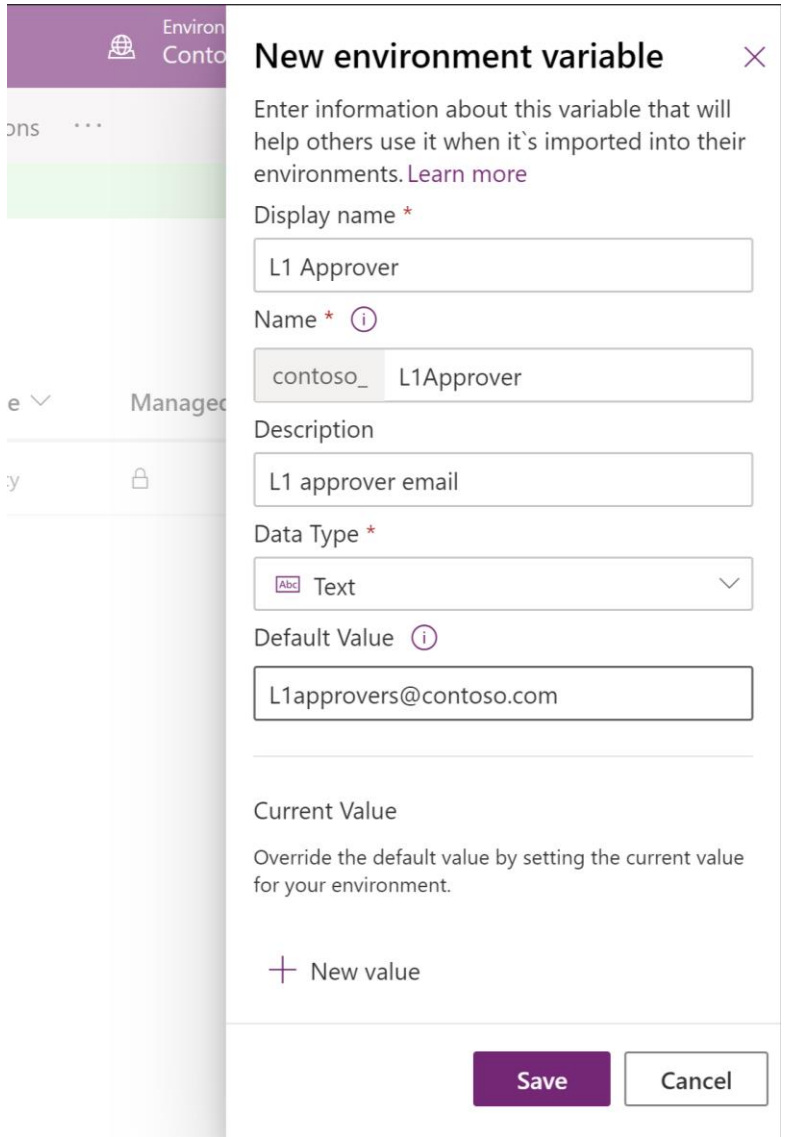
- **Upgrade** (Default) This is the default option and upgrades your solution to the latest version and rolls up all previous patches in one step. Any components associated to the previous solution version that are not in the newer solution version will be deleted.
- **Stage for Upgrade** This option upgrades your solution to the higher version, but defers the deletion of the previous version and any related patches until you apply a solution upgrade later.
- **Update** This option replaces your solution with this version. Components that are not in the newer solution won't be deleted and will remain in the system. (Not Recommended)
- Solution Version
  - Format is <major>.<minor>.<build>.<revision>
  - System automatically increments Build & Revision
  - You update Major & Minor to reflect the feature/fix/impact of the solution change

Update a solution - Power Apps | Microsoft Docs

Environment Variables to Configure Components

- Environment variables can be created and modified within the modern solution interface, automatically created when connecting to certain data sources in canvas apps, or by using code. They can also be imported to an environment via solutions.

[Use environment variables in solutions - Power Apps | Microsoft Docs]



Key Points on
Creating & Configuring Power Apps

Delegation

- *Delegation* is where the expressiveness of Power Apps formulas meets the need to minimize data moving over the network. In short, Power Apps will delegate the processing of data to the data source, rather than moving the data to the app for processing locally.
- Working with large data sets requires using data sources and formulas that can be delegated.
- Delegation is supported for certain tabular data sources only.
- Delegable Functions (yet depends on which data source it is)
  - Filter, Search, and LookUp can be delegated.
  - Sort and SortByColumns can be delegated.
  - Sum, Average, Min, and Max can be delegated.

Understand delegation in a canvas app - Power Apps | Microsoft Learn

Performance Optimization Tips

- Use Delegable Data Sources & Functions as possible.
- Use Local Collections for small data sets, will be also beneficial to do non-delegable operations.
- OnStart Code is invaluable for one-time calls to initialize app assets.
- Use “Concurrent” Function to run multiple data sources call in parallel instead of sequential.
- For SQL DBs, use Views to abstract tables traversing and join instead of directly handling the Tables from your app.
- Periodically republishing your apps to gain any new improvements of platform optimization released after your last publish.

Tips and best practices to improve performance of canvas apps - Power Apps | Microsoft Learn

Concurrent function in Power Apps - Power Platform | Microsoft Learn

Embedding Model-Driven & Canvas Apps in MS Teams

- You can embed Model-Driven & Canvas Apps in Teams in 2 ways
  - As a Tab: withing a Teams Channel
    - Adding is initiated from Microsoft Teams withing the desired Channel
  - As a Personal App: Added by Users to their Teams or Chats
    - Adding is initiated from Maker Portal, clicking the App and choose “Add to Teams”
    - Also, you can download the App and Upload it to Teams

Embed a canvas app as a personal app - Power Apps | Microsoft Learn

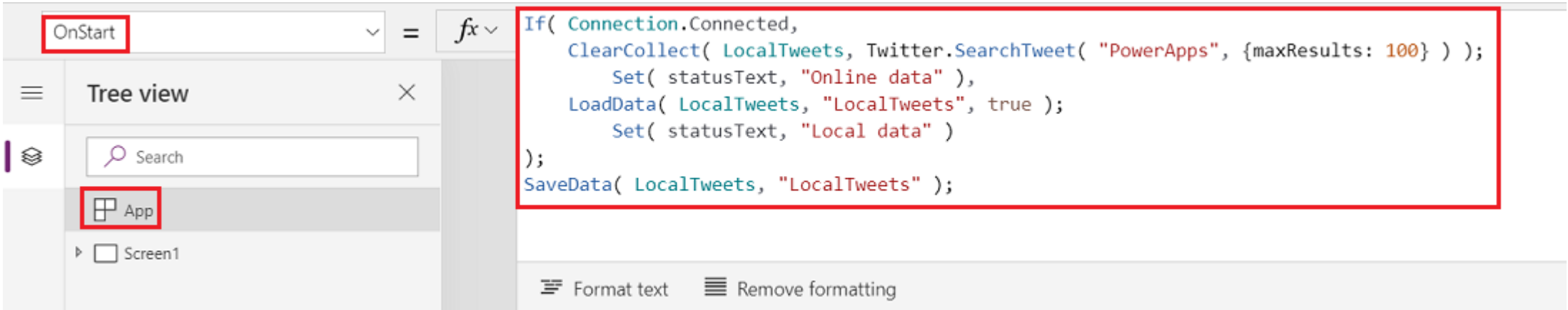
Embed a model-driven app as a personal app - Power Apps | Microsoft Learn

Embed a canvas app as tab app - Power Apps | Microsoft Learn

Embed a model-driven app as tab app - Power Apps | Microsoft Learn

Offline Capabilities

- Determine when an app is offline, online, or in a metered connection by using the Connection signal object.
- Use collections and leverage the LoadData and SaveData functions for basic data storage when offline.
- Offline capability for canvas apps is only available while running the apps using the native Power Apps Mobile players on iOS, Android, and Windows.



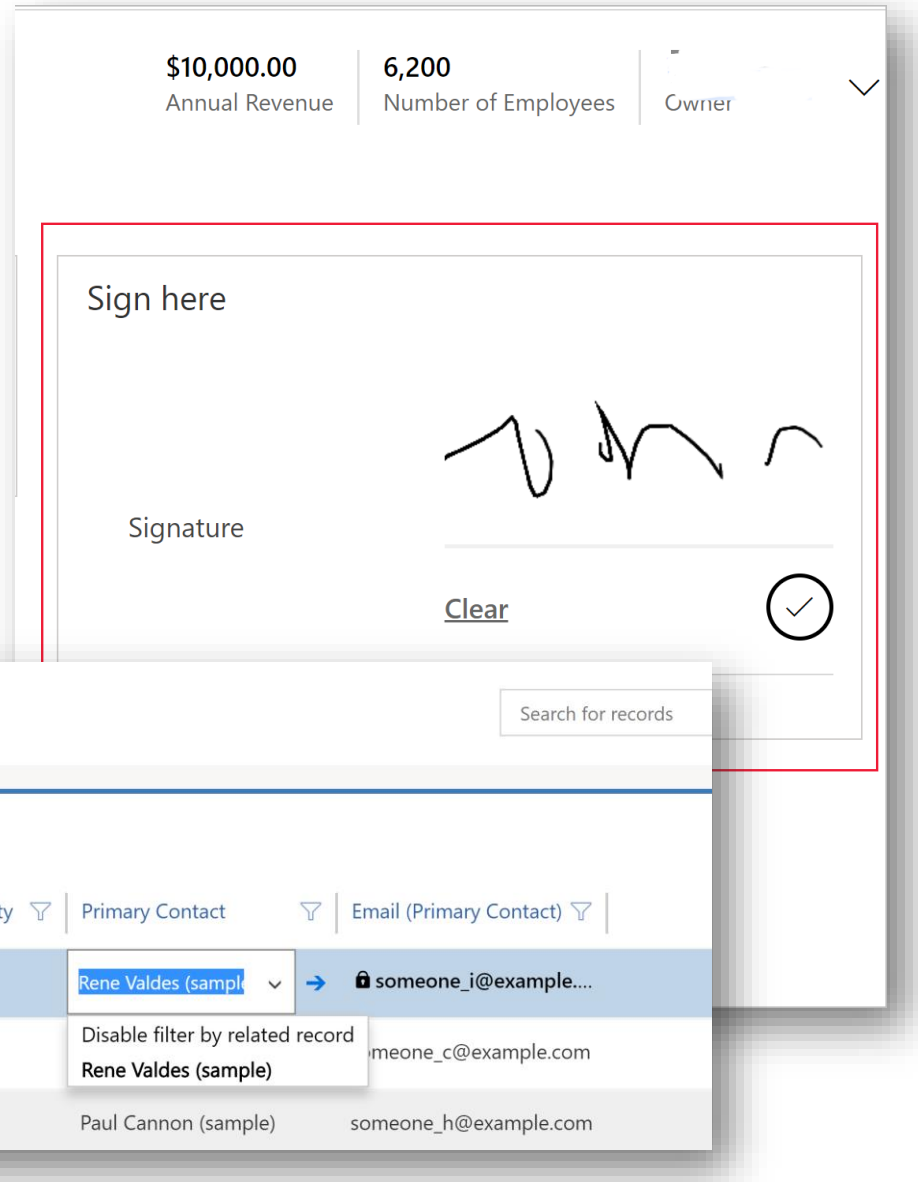
[SaveData, LoadData, and ClearData functions in Power Apps - Power Platform | Microsoft Learn]

Testing & Monitoring Model Driven & Canvas Apps

- Test Studio [Test Studio - Power Apps | Microsoft Docs]
  - Regression Test automation for Canvas Apps, can be integrated with Azure DevOps.
- Easy Repro [EasyRepro automated testing framework June Update is now available | Microsoft Power Apps]
  - Regression Test for Model Driven Apps, can be integrated with Azure DevOps.
- Monitor [Monitor overview - Power Apps | Microsoft Docs]
  - Debug & Troubleshoot a user's Session, view events, page navigations, formulas & command executions, form issues.
  - Real time
  - For Model Driven & Canvas Apps
- Power Platform Admin Analytics [Administrator analytics and reports for Microsoft Power Apps - Power Platform | Microsoft Learn]
  - Analytics for the environment admin is available at the Microsoft Power Platform admin center.
  - The admin reports provide a view into environment level usage, errors, and service performance
- Application Insights [Analyze telemetry of a canvas app using Application Insights - Power Apps | Microsoft Docs]
  - Collect usage and performance information about Model Driven & Canvas App
  - Provided by Azure, Generally can be used by any Application, not only Power Apps.
- Fiddler [Script web resource development using Fiddler AutoResponder (model-driven apps) - Power Apps | Microsoft Docs]
  - Client based debugging (browser), can be helpful to debug issues from browser standpoint like Javascript/TypeScripts used by PFC.

Model-Driven Apps UI Controls

- Controls exist to provide a more touch-friendly experience with model-driven apps.
- Grid controls
  - Subgrid
  - Editable grid
  - Power Apps grid control (preview)
- Display controls
  - Calendar
  - Canvas app
  - External website
  - HTML and image web resource
  - Knowledge search
  - Quick view
  - Timeline control
- Input controls
  - Checkbox
  - Number input
  - Option set
  - Pen input
  - Rich text editor
  - Star rating
- AI Builder Business card reader
- More



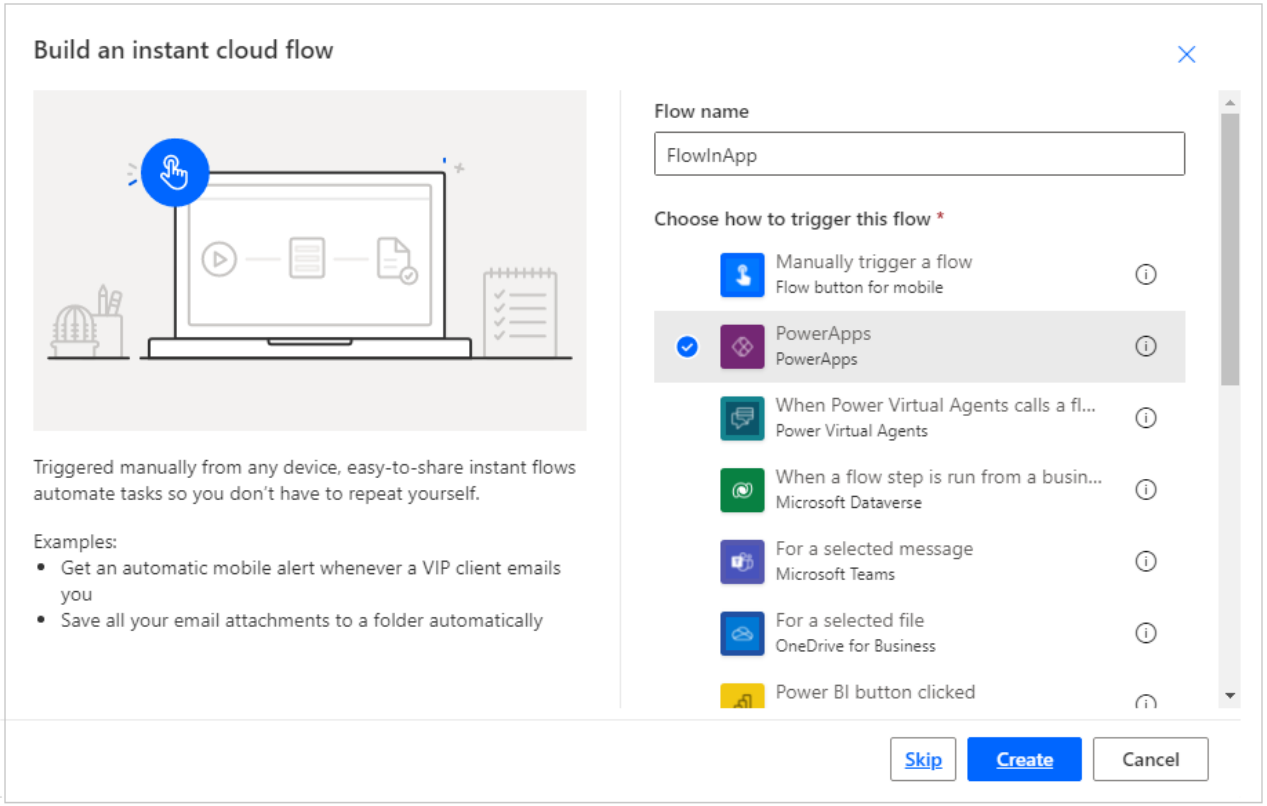
[Use controls for model-driven app data visualizations in Power Apps - Power Apps | Microsoft Learn]

[List of controls available for model-driven apps - Power Apps | Microsoft Learn]

Canvas Apps to Call Power Automate Flows

- Flow needs to be Instant Flow with Trigger “Power Apps”
- Flow may include Parameters (Ask in Power App) to be passed by the app
- Flow may end with action “Respond to Power App” to pass back output
- Canvas App Call, Pass input, collect output like below:

```
=  fx  Set(SalesInfoVariable,GetSalesInfo.Run())
```



[Start a flow in a canvas app - Power Apps | Microsoft Docs]

Canvas App Variable & Collections Scope

- Variables Scopes
  - Context Variables: In a Single Screen , or passed from one screen to another during navigation (Use “UpdateContext” or “Navigate” functions).
  - Global Variable: store a value or a record and accessible from any screen in the app (Use “Set” function)
- Collections Scope
  - Store a table of records
  - Can be accessed from anywhere in the app
    (Use “Collect”, “Clear”, “ClearCollect” functions)

Understand variables in canvas apps - Power Apps | Microsoft Docs

Create and update a collection in a canvas app (contains video) - Power Apps | Microsoft Docs

Update vs Patch

- **Update** function to replace an entire record in a data source.
- **UpdateIf** and the **Patch** functions modify one or more values in a record, leaving the other values alone.

```
Update( IceCream,
First( Filter( IceCream, Flavor="Chocolate" ) ),
{ ID: 1, Flavor: "Mint Chocolate", Quantity:150 }
)
```

```
Patch( IceCream,
Lookup( IceCream, Flavor =
"Chocolate" ),
{ Quantity: 400 } )
```

```
UpdateIf( IceCream, Quantity > 175,
{ Quantity: Quantity + 10 } )
```

```
Patch( IceCream,
Defaults( IceCream ),
{ Flavor: "Strawberry" } )
```

Patch function in Power Apps (contains video) - Power Platform | Microsoft Learn

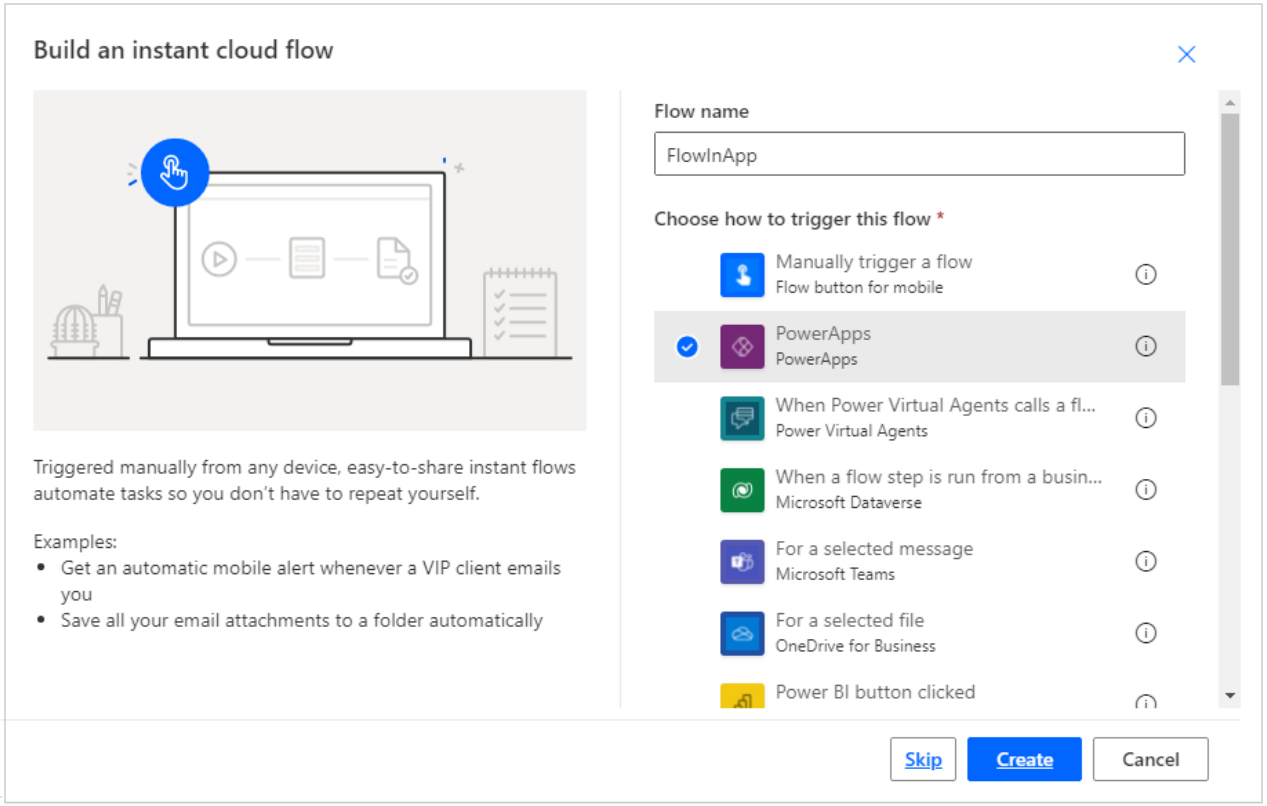
Update and UpdateIf functions in Power Apps - Power Platform | Microsoft Learn

Key Points on Configuring Power Automate Flows

Canvas Apps to Call Power Automate Flows

- Flow needs to be Instant Flow with Trigger “Power Apps”
- Flow may include Parameters (Ask in Power App) to be passed by the app
- Flow may end with action “Respond to Power App” to pass back output
- Canvas App Call, Pass input, collect output like below:

```
=  fx  Set(SalesInfoVariable,GetSalesInfo.Run())
```



[Start a flow in a canvas app - Power Apps | Microsoft Docs]

Dataverse Connector vs Dataverse Connector (Legacy)

- Dataverse Connector (Legacy)
  - Support Logic Apps & Power Automate
  - Cross-Environment
  - Avoid if packaging Flow in a solution
- Dataverse Connector
  - Currently Power Automate only.
  - Current Environment
  - Suitable for packaging Flows in solutions
- Not Same Actions/Triggers available
  - i.e. Manual Trigger “When a row is selected” is only available in Legacy

Microsoft Dataverse - Connectors | Microsoft Docs

Microsoft Dataverse (legacy) - Connectors | Microsoft Docs

Starting Flows from Business Process Flows

- Power Automate Flow must be of type “Instant”
- Connector to use is “Dataverse Connector (Legacy)”, it has “When row is selected trigger”
- The flow associate with the BPF table (same name as BPF)
- The flow must be “in a solution”, flows in Power Automate Portal can not be added to BPF

Preview Instant Flow steps in Business Process Flows | Power Automate Blog (microsoft.com)

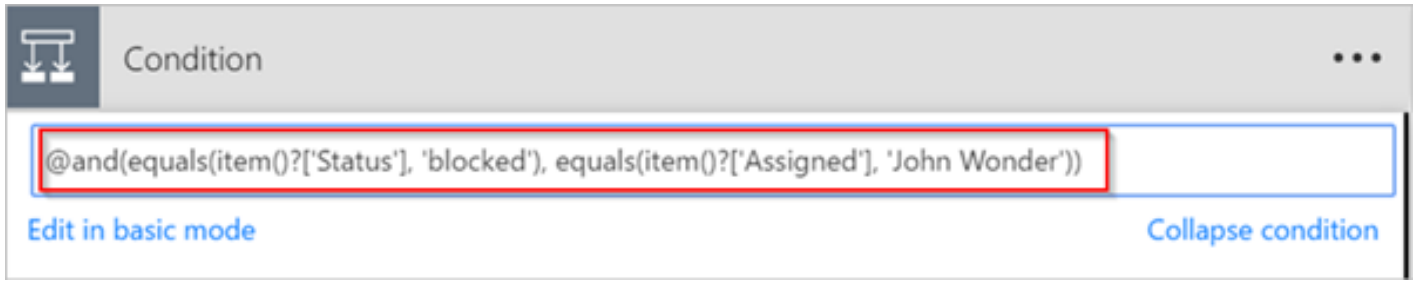
Working With BPF with Code

- Business process flow definition is stored in the “workflow” table, and the stage information for the business process flow is stored in the “processstage” table.
- BPF also have their own custom table created when activated, it store the different BPF instances associated with a particular process. The table name is driven from BPF name.
- For users to be able to see and access the business process flow, they need permissions on
  - “Process” Table
  - Custom BPF Table
- Via Dataverse API, your code could trigger many actions on a BPF instance changing state (active, finish, abandon) or stage navigation

[Work with business process flows using code - Power Automate | Microsoft Learn]

Power Automate Expressions

- Condition action can have basic comparison options if we use the UI approach to build the condition
- We can use Expressions to compose advanced Conditions

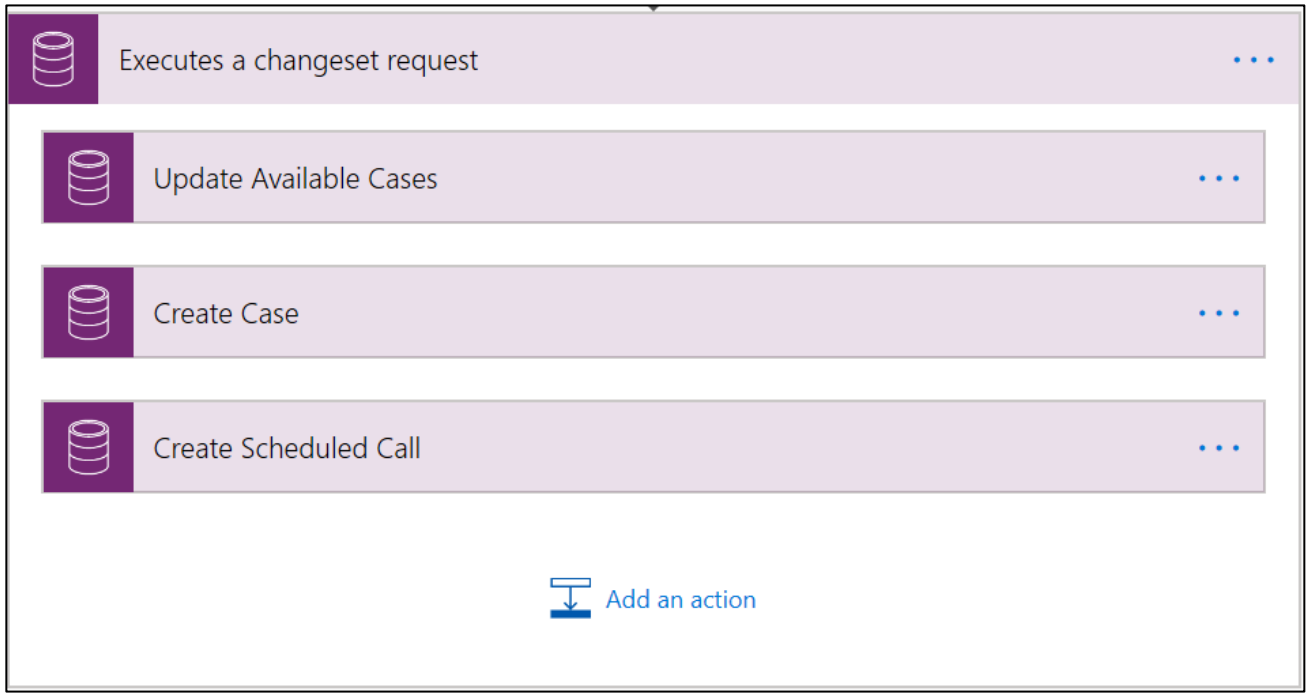


[Use expressions in conditions in Power Automate - Power Automate | Microsoft Learn]

Expression	Example
and	This expression returns false: and(greater(1,10),equals(0,0))
or	This expression returns true: or(greater(1,10),equals(0,0))
equals	For example, if parameter1 is someValue, this expression returns true: equals(parameters('parameter1'), 'someValue')
less	This expression returns true: less(10,100)
lessOrEquals	This expression returns true: lessOrEquals(10,10)
greater	This expression returns false: greater(10,10)
greaterOrEquals	This expression returns false: greaterOrEquals(10,100)
empty	This expression returns true: empty('')
not	This expression returns true: not(contains('200 Success','Fail'))
if	This expression returns "yes": if(equals(1, 1), 'yes', 'no')

Using Change sets

- Operations run in a transaction
- Microsoft Dataverse Connector actions only, Create Update Delete

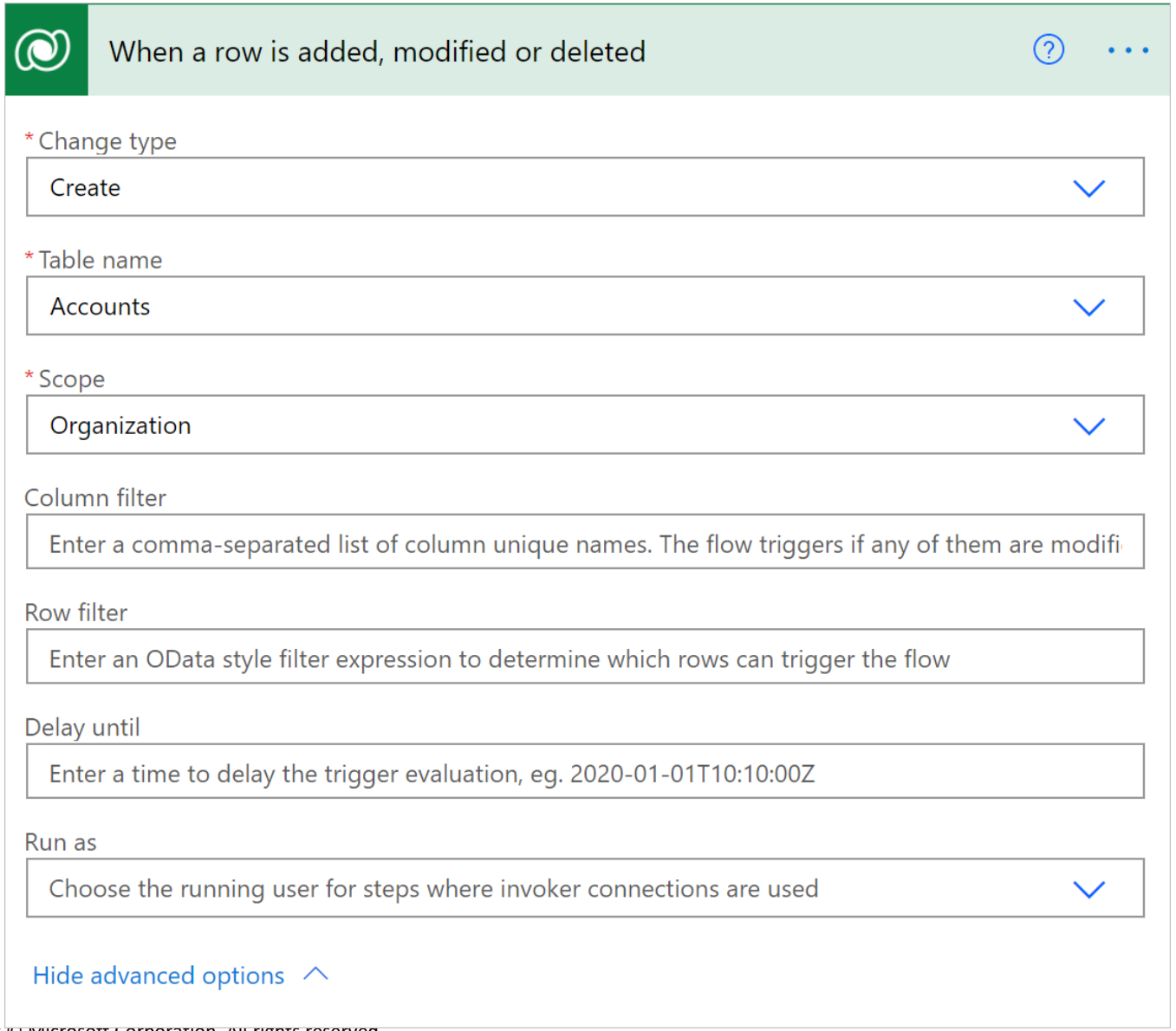


[Use a flow to perform a change set request in Dataverse - Power Automate | Microsoft Learn]

Power Automate Flow – Optimize Triggering

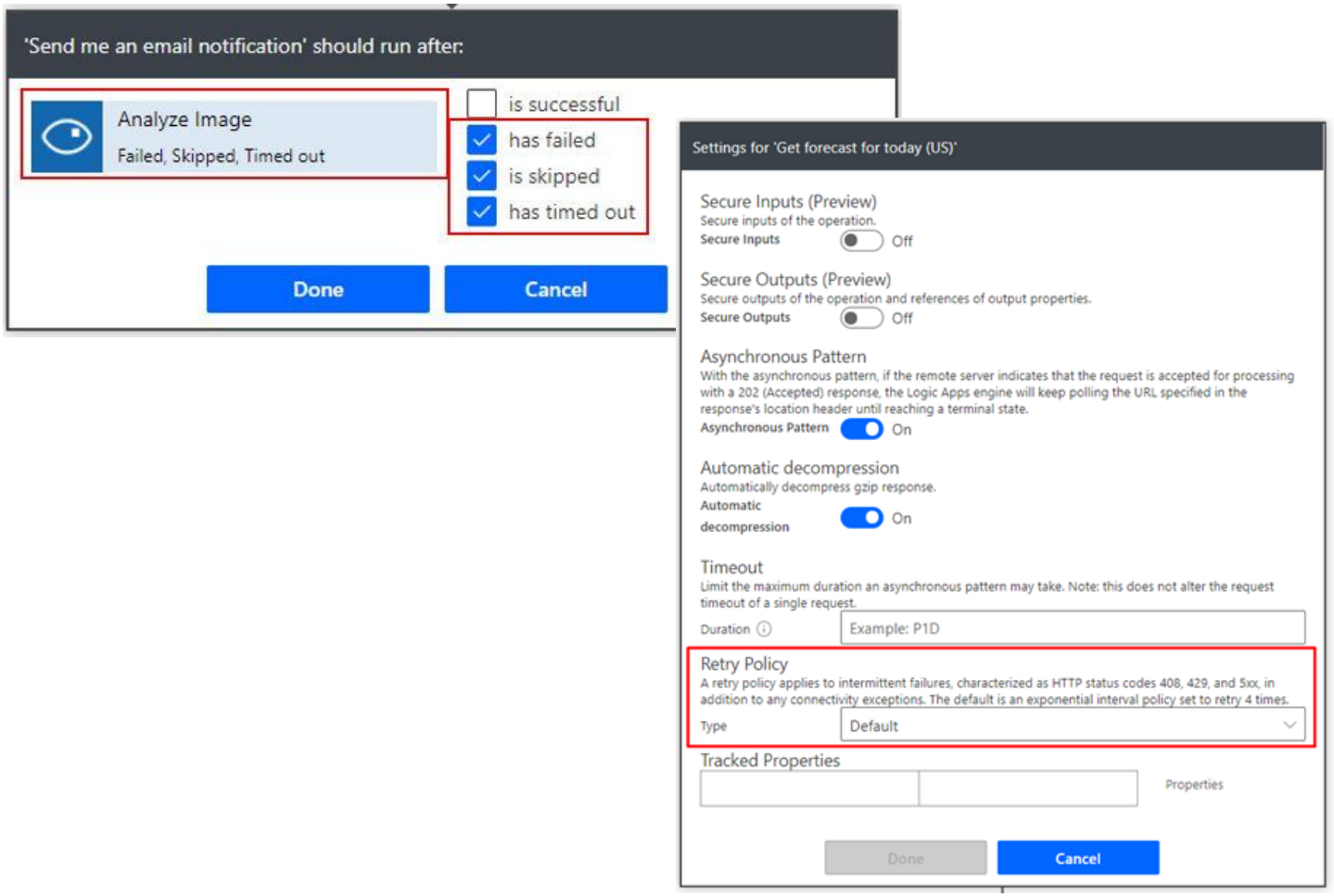
- Use Scope to reduce which user action can cause the flow to Trigger
- Use Row filter to trigger only when a row has a certain criteria
- User Column filter to trigger only if the filtered columns are modified

[Trigger flows when a row is added, modified, or deleted - Power Automate | Microsoft Docs]



Power Automate Error Handling

- Use “Configure Run After” to direct the follow according to last step outcome.
- Parallel Branch is typically done first to branch into a success and fail paths
- Use “Retry Policy” to configure Flow to retry an action if it fails



[Configure run after option - Learn | Microsoft Docs]

[Reducing risk and planning for error handling in a Power Automate project - Power Automate | Microsoft Docs]

Business Rules Troubleshooting

- Business Rules will not trigger/execute if:
  - Rule is deactivated
  - The column referenced in the rule is not added to the form.
  - If the “Scope” is set to a specific form different from the one being targeted.
  - If the “Scope” is set to a specific form or All Forms but the validation is needed in Power Apps Portal or Canvas Apps (Scope need to be set to “Table/Entity”)
  - Make sure you custom Javascripts (if you use it) are not competing or conflicting with the Business Rule (probably disable them temporary to troubleshoot)

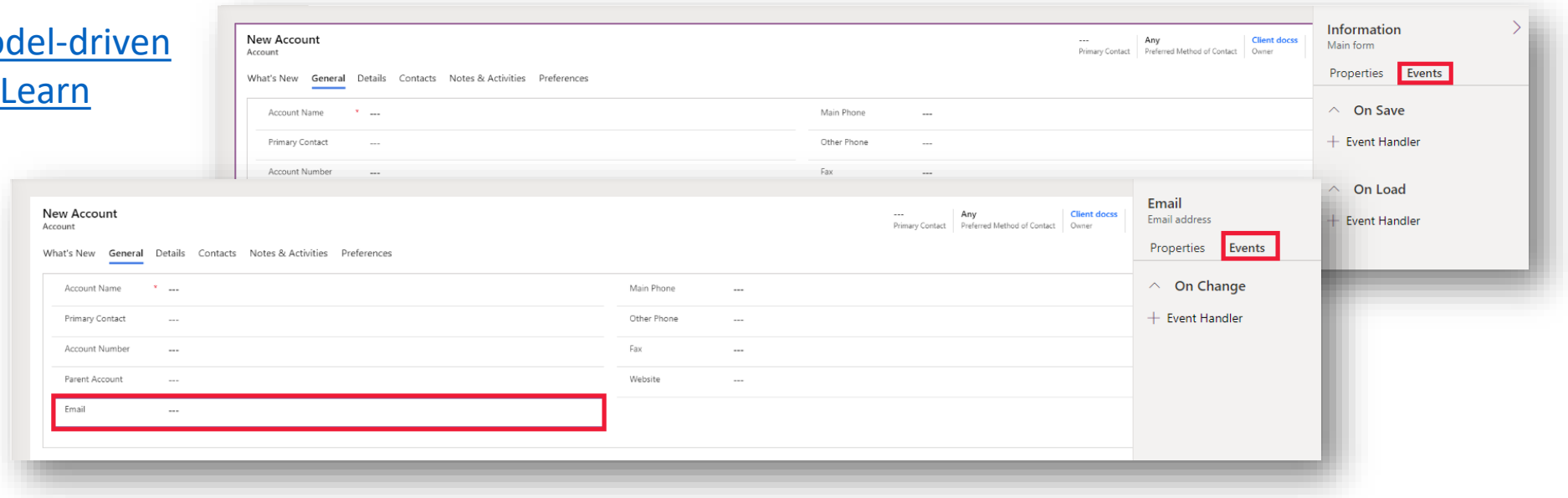
Create model-driven app business rules and recommendations - Power Apps | Microsoft Learn

Key Points on Extending User Experience

Events in forms and grids in model-driven apps

- In model-driven apps, you associate a specific function in a JavaScript library (Script web resource) to be executed when an event occurs.
- For events that are not available to be associated through UI, Client API provides methods that can be used to attach event handlers to such events.

[Events in forms and grids in model-driven apps - Power Apps | Microsoft Learn]



Understand the Client API object model

- The Client API object model for model-driven apps provides you objects and methods that you can use to apply custom business logic in model-driven apps using JavaScript.

Object	Description
executionContext	Represents the execution context for an event in model-driven apps forms and grids. More information: Client API execution context
formContext	Provides a reference to a form or an item on the form against which the current code executes. To get the **formContext** object, use the **executionContext**.getFormContext method. More information: Client API form context
gridContext	Provides a reference to a grid or a subgrid on a form against which the current code executes. More information: Client API grid context
Xrm	Provides a global object for performing operations that do not directly impact the data and UI in forms, grids, subgrids, controls, or columns. For example, navigate forms, create and manage records using Web API. More information: Client API Xrm object

Understand the Client API object model in model-driven apps - Power Apps | Microsoft Learn

Controls (Client API reference)

- Can be accessed from the FormContext (formContext.ui.controls)
- The control object provides methods to change the presentation or behavior of a control and identify the corresponding column.
- Important Functions
  - addNotification (Client API reference) in model-driven apps - Power Apps | Microsoft Learn
  - setFormNotification (Client API reference) in model-driven apps - Power Apps | Microsoft Learn
  - addGlobalNotification (Client API reference) in model-driven apps - Power Apps | Microsoft Learn
  - addPreSearch (Client API reference) in model-driven apps - Power Apps | Microsoft Learn

Xrm.Navigation (Client API reference)

- Provides navigation-related methods.

Method	Description
navigateTo	Navigates to the specified table list, table record, HTML web resource, or custom page.
openAlertDialog	Displays an alert dialog containing a message and a button.
openConfirmDialog	Displays a confirmation dialog box containing a message and two buttons.
openErrorDialog	Displays an error dialog.
openFile	Opens a file.
openForm	Opens an entity form or a quick create form.
openUrl	Opens a URL, including file URLs.
openWebResource	Opens an HTML web resource in a new window.

Xrm.WebApi (Client API reference)

Property	Description	Funcitons
online	Provides methods to use Web API to create and manage records and execute Web API actions and functions in model-driven apps when connected to the model-driven apps server (online mode).	• createRecord • deleteRecord • retrieveRecord • retrieveMultipleRecords • updateRecord • execute • executeMultiple
offline	Provides methods to create and manage records in model-driven apps in mobile clients while working in the offline mode.	• createRecord • deleteRecord • retrieveRecord • retrieveMultipleRecords • updateRecord • isAvailableOffline

Power Apps Component Framework (PCF)

[Take some time on this]

- Empowers professional developers and app makers to create code components for model-driven and canvas apps.
- can be used to enhance the user experience for users working with data on forms, views, dashboards, and canvas app screens.
- Code components consist of three elements:
  - **Manifest**
    Manifest is the metadata file that defines a component. It is an XML document that describes:
    - The name of the component.
    - The kind of data that can be configured, either a field or a dataset.
    - Any properties that can be configured in the application when the component is added.
    - A list of resource files that the component needs.
  - **Component implementation**
    The object implements the following methods:
    - init (Required)
    - updateView (Required)
    - destroy (Required)
    - getOutputs (Optional)
  - **Resources**
    the resource files that component requires to implement it's visualization (i.e. CSS, Images, etc)

Create your first component using Power Apps Component Framework in Microsoft Dataverse - Power Apps | Microsoft Learn

Power Apps component framework overview in Microsoft Dataverse(contains video) - Power Apps | Microsoft Learn

What are code components? - Power Apps | Microsoft Learn

Customize Commands and the Ribbon

[Take some time on this]

- **Community tool**
  - The SDK describes the process of editing the ribbon by editing the customization.xml file directly. You can also use a community tool, Ribbon Workbench, to visually edit ribbons using the UI.
- **Modern App Designer**

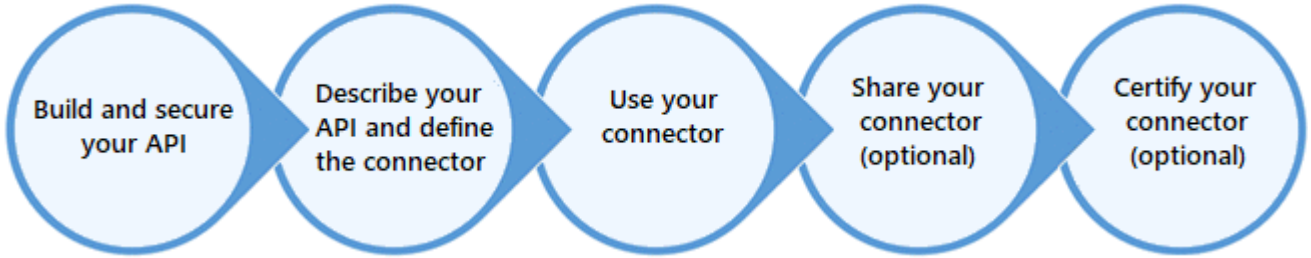
  Customize the command bar - Power Apps | Microsoft Learn
  - The new Modern App Designer allows customization of the Command Bar visually without 3rd Party Tools.
  - Actions taken by the commands can be implanted by JavaScript or PowerFX
- **Low Level Customization (using XML)**

  Ribbons available in model-driven apps - Power Apps | Microsoft Learn

  Pass parameters to a URL by using the ribbon (model-driven apps) - Power Apps | Microsoft Learn

Key Points on Extending The Platform

Custom connectors



Custom connectors overview | Microsoft Learn

- 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.
- The Custom Connector describes the API you wants to connect to.
- You can create a custom connector using:
  - Custom Connector Wizard
    Create a custom connector from scratch | Microsoft Learn
  - Postman Collection
    Create a custom connector from a Postman collection | Microsoft Learn
  - OpenAPI Definition
    Create a custom connector from an OpenAPI definition | Microsoft Learn
  - CLI
    Microsoft Power Platform Connectors CLI | Microsoft Learn

Custom Connectors Policy Templates

- Policies allow you to modify the behavior of a custom connector at runtime.
- Common policy use cases are:
  - **Set host URL** - By default, the host URL is hardcoded into the connector configuration. A policy, combined with a connection parameter, allows the host URL to be specified every time you create a connection by using the connector. For example, this use case could handle a dev, test, and production version of the system API.
  - **Set header** - Use to promote data from the connection parameters, query, or body of the request to the header. Commonly, this process is done to accommodate APIs that want specific information configured in the header. For example, this use case could be a correlation ID to track the full business process that is being performed.
  - **Set query parameter** - Use for handling default values, if necessary, but the maker doesn't configure one. For example, if an API required a top parameter for how many records to return by using a policy, you could set a default value to be used if the maker doesn't configure one.

Introduction To Policies - Training | Microsoft Learn

What is policy and where is it used? | Microsoft Learn

Custom APIs in Dataverse

- Use Custom APIs to create your own APIs in Dataverse. You can consolidate one or more operations into a Custom API that you and other developers can call in their code or from Power Automate.
- You can create a Custom API using:
  - Plug-in Registration Tool
    Create a Custom API using the plug-in registration tool (Microsoft Dataverse) - Power Apps | Microsoft Learn
  - Power Apps Maker Portal
    Create a Custom API in Power Apps (Microsoft Dataverse) - Power Apps | Microsoft Learn
  - Also with Code or Solution XML Files
- Your Custom API can be
  - **Function:** retrieving data without an impact, called with GET HTTP Request
  - **Action (Function=No):** making impact and changing data, called with POST HTTP Request
  - **Public:** Discoverable via Dataverse Metadata, other developer can see it and use it.
  - **Private:** not disclosed in Metadata, other developers in your company can still use it if they know about it.

Create and use Custom APIs (Microsoft Dataverse) - Power Apps | Microsoft Learn

Plug-ins

[Take some time on this]

- A plug-in is a .NET assembly that you can upload to the Microsoft Dataverse. Classes within the assembly can be registered to specific events (steps) within the event framework.
- Within the Execute method you can:
  - Cancel the event and display an error to the user
  - Make changes to the data in the operation
  - Initiate other actions to add automation
- Your Plug-in can be registered with Execution Mode
  - **Synchronous -** execute immediately according to the stage of execution and execution order. The entire operation will wait until they complete.
  - **Asynchronous –** Runs in system job, which will execute after the operation completes.
- Registering your Plug-in to the Event Pipeline
  - **PreValidation**
    Before Operation – Out of Database Transaction
  - **PreOperation**
    Before Operation – In Database Transaction
  - **PostOperation**
    After Operation – Out of Database Transaction
- Asynchronous plug-ins **can only be registered for the PostOperation** stage.

Register a plug-in (Microsoft Dataverse) - Power Apps | Microsoft Learn

Tutorial: Write and register a plug-in (Microsoft Dataverse) - Power Apps | Microsoft Learn

3 Parts Tutorial, Use them ALL

Use OAuth authentication with Microsoft Dataverse

- **App Registration**
  - When you connect using OAuth you must first register an application in your Azure AD tenant.
- **Giving access to Dataverse**
  - **Interactive User:** If your app will be a client which allows the authenticated user to perform operations, you must configure the application to have the Access Dynamics 365 as organization users delegated permission.
  - **No Interactive User:** If your app will use Server-to-Server (S2S) authentication, this step is not required. That configuration requires a specific system user and the operations will be performed by that user account rather than any user that must be authenticated.

Use OAuth authentication with Microsoft Dataverse (Dataverse) - Power Apps | Microsoft Learn

Tutorial: Register an app with Azure Active Directory (Microsoft Dataverse) - Power Apps | Microsoft Learn

Learn about MSAL - Microsoft Entra | Microsoft Learn

{Step by Step Guide} Query Dynamics CRM Web API using Server to Server Authentication with Application User - Microsoft Dynamics CRM Community

Application Users

- Application Users are Identities used by your Application interacting with Dataverse via API or SDK.
- You associate a Dataverse Application User with an Azure AD Application
- Application Users are automatically set to Non-Interactive (they don’t access the Portals or UI), but they don’t count against the limit of total 7 Non-Interactive.
- Application Users can be assigned Security Roles to restrict & control permission within Dataverse

Manage application users in the Power Platform admin center - Power Platform | Microsoft Learn

Integration within Service API Protection Limits Error – 429 Too Many Requests

- To ensure consistent availability and performance for everyone we apply some limits to how APIs are used. These limits are designed to detect when client applications are making extraordinary demands on server resources.
- Service protection API limits are enforced based on three facets:
  1. The number of requests sent by a user (5 Minutes sliding window).
  2. The combined execution time required to process requests sent by a user. (5 Minutes sliding window)
  3. The number of concurrent requests sent by a user.

  Service protection API limits (Microsoft Dataverse) - Power Apps | Microsoft Docs
- Common Scenarios For Integration Applications:
  - Retry Pattern: API returns a "Retry-After" duration when a limit is hit, Applications can use it to retry the request.
  - Create Custom API (runs as a plug-ins combining multiple operations per request, service limits don't apply to plugins)
  - Combine multiple operations in a batch of a single request. Less number of requests, but longer combined execution time.

  Create and use Custom APIs (Microsoft Dataverse) - Power Apps | Microsoft Docs

Identify HTTP Status Codes & Handle Errors

- When you develop an application to connect to Dataverse, this happens over HTTPs Protocol. Dataverse responds to API calls as per the situation of processing that call.
- Its important to understand which HTTP Response Code means what, to handle the Dataverse response properly.

Compose HTTP requests and handle errors (Microsoft Dataverse) - Power Apps | Microsoft Learn

Key Points on Integration Development

Webhooks

- Use Webhooks to create external handlers for server events.
- With Microsoft Dataverse, you can send data about events that occur on the server to a web application using webhooks.
- Webhooks is a lightweight HTTP pattern for connecting Web APIs and services with a publish/subscribe model.
- Webhook senders (Dataverse) notify receivers (Your App) about events by making requests to receiver endpoints with some information about the events.
- Azure Functions provide an excellent way to deliver a solution using Webhooks, but it is not a requirement, you can use any technology or skill you know.

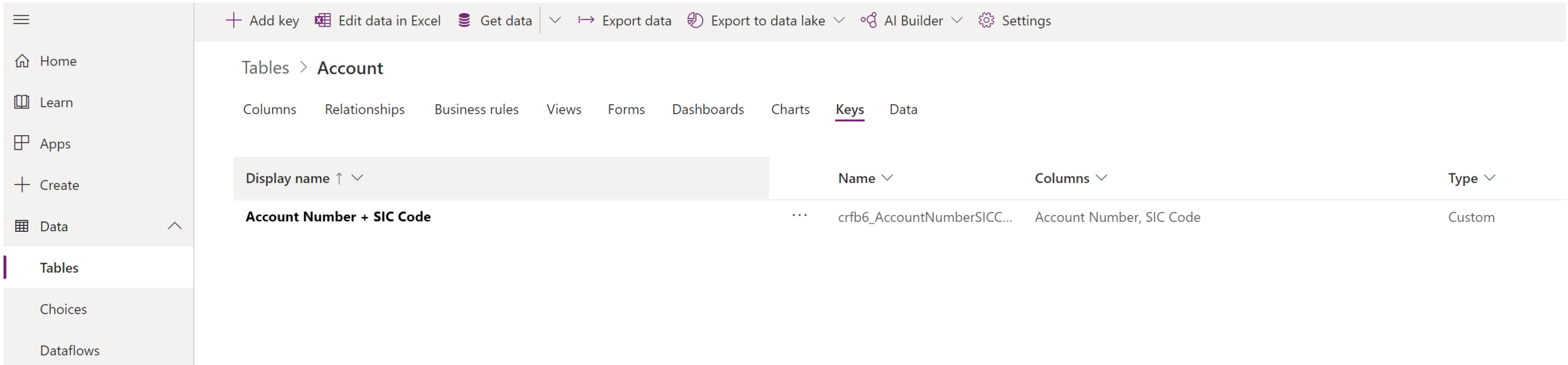
1. Use Webhooks to create external handlers for server events (Microsoft Dataverse) - Power Apps | Microsoft Learn

2. Register a WebHook (Microsoft Dataverse) - Power Apps | Microsoft Learn

3. Test WebHook registration with request logging site (Microsoft Dataverse) - Power Apps | Microsoft Learn

Table Alternate Keys

- Alternate keys provide an efficient and accurate way of integrating data with external systems (external system doesn't store the Globally Unique Identifier (GUID) IDs)
- Alternate Keys add unique nullable index on the column.



[Work with alternate keys (Microsoft Dataverse) - Power Apps | Microsoft Docs]

Change Tracking to Sync with External Systems

- Change Tracking helps to detect what data has changed since the data was initially extracted or last synchronized.
- Change Tracking must be enabled at Table level in Dataverse
  - Via Power Apps Maker Portal
  - Or Programatically
- First Time Call
  - Application send request (with Tracking Preferred) for Data from Dataverse.
  - Dataverse returns requested Data + Delta Token.
- Next Time Call
  - Application send request for Data + Previous Delta Token.
  - Dataverse returns requested Data + Changes Happened + New Delta Token
- And so ..

[Use change tracking to synchronize data with external systems (Microsoft Dataverse) - Power Apps | Microsoft Learn]

Web APIs & CORS

- CORS provides a way to gain access to resources on another domain.
- To protect access to data and resources on other domains, all modern browsers enforce a Same-Origin policy to prevent sites from using data and resources from sites on a different domain.
- you can use the Microsoft Authentication Library for JavaScript (MSAL.js) 2.0 for Browser-Based Single-Page Applications and it will take care of much of the CORS complexity for you.
- For the case of PowerApps Portals
  - The portals Web API enables a richer user experience inside Power Apps portals pages. You can use the Web API to perform create, read, update, and delete operations across all Microsoft Dataverse tables from your portals pages.

Use OAuth with Cross-Origin Resource Sharing to connect a Single-Page Application (Microsoft Dataverse) - Power Apps | Microsoft Learn

Overview of portals Web API - Power Apps | Microsoft Learn

Briefing on Official Practice Tests

Briefing - Microsoft Certified Official Practice Test

[Official Practice Test PL-400: Microsoft Power Platform Developer (measureup.com)]

Useful Resources

- Microsoft Learn
  - Create a canvas app in Power Apps - Training | Microsoft Learn
  - Create a model-driven application in Power Apps (microsoft.com)
  - Work with Microsoft Power Platform tenants, environments, subscriptions, and Dynamics 365 apps
  - Automate a business process using Power Automate (microsoft.com)
  - Create and use analytics reports with Power BI (microsoft.com)
  - Integrate with Microsoft Power Platform and Dataverse
  - Create components with Power Apps Component Framework (microsoft.com)
  - Administer Power Apps portals (microsoft.com)
  - Extend Power Apps portals (microsoft.com)
- Official PL400 Labs
  - GitHub - MicrosoftLearning/PL-400_Microsoft-Power-Platform-Developer

Q & A (s)

Thank You