

P09: Managing Page Content Across Model-Driven Apps and Power Pages with Dataverse

Wednesday, March 12, 2025 12:11 PM

Context

Item	Detail
Tag	Vertical Patterns Power Apps
Contributors	Vincent Kimiti
Version Control	V1.0
Application Use Case	Manage and publish page content from a Model-Driven App to Power Pages using Dataverse and Liquid Code
Reference Usage	Integrating Model-Driven Apps, Power Pages, and Dataverse for content management

Story Behind The Pattern

Item	Detail
The Problem	Manually updating and synchronizing page content across Model-Driven Apps and Power Pages is inefficient and prone to errors.
The Solution	Use a Model-Driven App to update page content stored in Dataverse, publish changes to Power Pages, and leverage Liquid Code to dynamically read and update the Power Pages site.
Dependencies	Power Apps environment, Dataverse, Power Pages site, Liquid Code configuration, permissions to create and manage apps and pages.

The Pattern

Method: Setting Up Page Content Management with Model-Driven Apps, Power Pages, and Dataverse

Here's a step-by-step guide to set up and manage page content across a Model-Driven App, Power Pages, and Dataverse, with Liquid Code integration:

1. Create or Configure the Model-Driven App
 - Log in to your Power Apps environment.
 - Navigate to the Solutions area in the Power Apps Maker Portal.
 - Create a new Model-Driven App or open an existing one (e.g., "Content Management App").
 - Add a form or page where users can update page content (e.g., a "Page Content" field or a rich text editor).
2. Define the Dataverse Table for Page Content
 - In the same solution, create or select a Dataverse table to store page content (e.g., "Page Content" table).
 - Add necessary columns, such as:
 - "Page Title" (Text)
 - "Page Content" (Multiple Lines of Text)
 - "Last Updated" (DateTime)
 - Save and publish the table to make it available for the Model-Driven App.
3. Link the Model-Driven App to the Dataverse Table
 - In the Model-Driven App designer, add the "Page Content" table to the app.

- Configure the form to display and allow edits to the "Page Title" and "Page Content" fields.
 - Save and publish the app to enable users to update page content, which will be stored in Dataverse.
4. Set Up the Power Pages Site
- Navigate to the Power Pages section in the Power Apps Maker Portal.
 - Create a new Power Pages site or open an existing one (e.g., "Company Website").
 - Add a page (e.g., "Dynamic Content Page") where the content will be displayed.
5. Configure Liquid Code to Read from Dataverse
- In the Power Pages site editor, access the page's web template or code editor.
 - Add Liquid Code to dynamically fetch content from Dataverse. Example:
 - This code retrieves the latest "Page Title" and "Page Content" from the Dataverse "Page Content" table and displays it on the Power Pages site.
6. Publish Changes from the Model-Driven App to Dataverse
- As a user, open the Model-Driven App and navigate to the form for the "Page Content" table.
 - Update the "Page Title" and "Page Content" fields with new information (e.g., "Welcome Page" and a paragraph of text).
 - Save the record. The changes are automatically stored in Dataverse.
7. Publish Changes from Power Pages to the Power Pages Site
- In the Power Pages editor, click Sync Configuration to ensure the site reflects the Dataverse data.
 - Click Publish to make the updated page content live on the Power Pages site.
 - Verify that the changes (e.g., the new "Welcome Page" content) appear on the published site.
8. Test the End-to-End Workflow
- As a user, update the page content again in the Model-Driven App.
 - Check Dataverse to confirm the update is stored.
 - Refresh the Power Pages site and verify that Liquid Code has updated the content dynamically.
 - Ensure no errors occur during the process (e.g., connectivity issues or permission errors).