

You

Let's Explore All Real Time Projects based QnA for all kinds of INFOPATH VS POWER APPS USED ON MICROSOFT 365 SHAREPOINT ONLINE VS ON PREM FROM 2003,3007,2010,2013,2016,2019 BOTH CLASSIC VS MODERN at all Levels: Admin, Development, Designing, Migration, Branding level for Super Senior Architect Consultant level IN A TABULAR REPRESENTATION MENTIONING ALL KINDS OF POINTS SUMMARY !!

Copilot

Certainly! Let's delve into the key differences between InfoPath and Power Apps, especially in the context of Microsoft 365 SharePoint. I'll provide a concise summary in a tabular format for easy reference:

Aspect	InfoPath	Power Apps
Advantages	- Familiar for InfoPath power users - Connects to external data sources - Customizable forms	- No-code platform - Leverages existing business and technology skills - Supports multiple data sources and connectors (200+ options) - Mobile-friendly
SharePoint Integration	Both InfoPath and Power Apps integrate with SharePoint.	Both InfoPath and Power Apps can be used to create form-based solutions on your intranet.
Migration Path	There is no automated migration path from InfoPath to Power Apps.	- Power Apps can replace SharePoint list forms (in the "Modern" SharePoint experience).
Deployment and Updates	- InfoPath forms can be deployed but require additional steps for users. - Updates are cumbersome.	- Power Apps can be deployed via a simple URL. - Updates are straightforward.

Remember that Power Apps offers a more robust and flexible platform, especially when it comes to mobile experiences and connecting to various services. If you've been building great things in InfoPath, transitioning to Power Apps can empower you further! 😊

You

How can I migrate from InfoPath to Power Apps?

Copilot

Migrating from InfoPath to Power Apps involves several steps. Here's a high-level guide:

1. Assessment and Planning:

- Evaluate your existing InfoPath forms and identify their functionality, data sources, and dependencies.
- Understand the business requirements and user needs that the forms address.

2. Inventory and Prioritization:

- Create an inventory of all InfoPath forms in use.
- Prioritize forms based on their criticality and usage.

3. Design and Build in Power Apps:

- Recreate the forms in Power Apps using the Canvas App or Model-Driven App approach.
- Leverage Power Apps connectors to connect to data sources (SharePoint lists, SQL databases, etc.).
- Design the user interface and implement form logic.

4. Data Migration:

- Migrate data from existing InfoPath forms to the new Power Apps forms.
- Ensure data consistency and integrity during the migration.

5. Testing and Validation:

- Thoroughly test the Power Apps forms for functionality, data validation, and user experience.
- Involve end-users in testing to validate the new forms.

6. Deployment and Training:

- Deploy the Power Apps forms to the appropriate users or groups.
- Provide training and documentation to users on how to use the new forms.

7. Retire InfoPath Forms:

- Communicate the transition to users and encourage adoption of the new Power Apps forms.
- Eventually, decommission InfoPath forms once all users have migrated.

Remember that the transition may require adjustments, especially if you're moving from a structured form-based approach (InfoPath) to a more flexible, app-based approach (Power Apps). But the benefits of modernization and scalability are worth it! 😊

You

Tell me more about connectors in Power Apps.

Copilot

Certainly! **Power Apps Connectors** play a crucial role in facilitating seamless communication between your Power Apps and external data sources, services, and platforms. Here's what you need to know:

1. Types of Connectors:

- **Built-in Connectors:** These are pre-built connectors provided by Microsoft for popular services like SharePoint, SQL Server, Office 365, Salesforce, and Twitter.
- **Standard Connectors:** These are connectors available in Power Apps that allow you to retrieve and push data to various sources.
- **Premium Connectors:** Some connectors require additional licensing (e.g., Azure services, SAP, etc.).

2. Using Connectors:

- To add data to your canvas app, you can connect to various data sources using connectors.
- Connectors provide access to tables of data or specific actions (e.g., retrieving records, creating items, updating data).

3. Popular Connectors:

- **SharePoint:** Interact with SharePoint lists, libraries, and documents.
- **SQL Server:** Connect to on-premises or cloud-based SQL databases.
- **Office 365:** Access data from Outlook, OneDrive, and other Office 365 services.
- **Salesforce:** Integrate with Salesforce records and objects.
- **Twitter:** Retrieve tweets and user data.

Remember, connectors empower you to build powerful apps by bridging the gap between your app and external data! 😊

You

How can I create a custom connector in Power Apps?

Copilot

Certainly! Creating a custom connector in Power Apps allows you to integrate external APIs and services. Let's walk through the steps:

1. Sign In:

- Sign in to Power Apps or Power Automate.
- Navigate to **Data > Custom connectors**.

2. Create from Blank:

- Select **New custom connector > Create from blank**.
- Provide a meaningful **name** for your connector.

3. General Details:

- Update the **description** (helpful for others).
- Specify the **API host** (e.g., for Cognitive Services Text Analytics API: `westus.api.cognitive.microsoft.com`).

4. Authentication:

- On the **Security** tab, choose **API Key** as the authentication type.
- Specify the API key parameter label, name, and location (e.g., `Ocp-Api-Subscription-Key`).

5. Save and Close:

- Save your custom connector.

Remember, custom connectors empower you to seamlessly connect to external data sources within your Power Apps! 😊

You

What if the external API requires OAuth authentication?

Copilot

When dealing with an external API that requires OAuth authentication, you can create a custom connector in Power Apps to handle it. Here's how:

1. Create a Custom Connector:

- Navigate to **Data > Custom connectors**.
- Select **New custom connector > Create from blank**.
- Provide a name and description for your connector.

2. Authentication:

- On the **Security** tab, choose the appropriate authentication type (OAuth 2.0 in this case).
- Configure the OAuth settings:
 - **Authorization URL:** The URL where users authenticate and grant access.

- **Token URL:** The URL to obtain an access token.
- **Refresh URL:** Optional, for token refresh.
- **Client ID** and **Client Secret:** Obtain these from the OAuth provider (e.g., Azure AD).

3. **Scopes and Parameters:**

- Define the required scopes (permissions) for your API.
- Set up any additional parameters needed for authentication.

4. **Test and Save:**

- Test your connector to ensure authentication works.
- Save your custom connector.

Remember to follow the OAuth flow, obtain the necessary credentials, and configure your custom connector accordingly! 😊