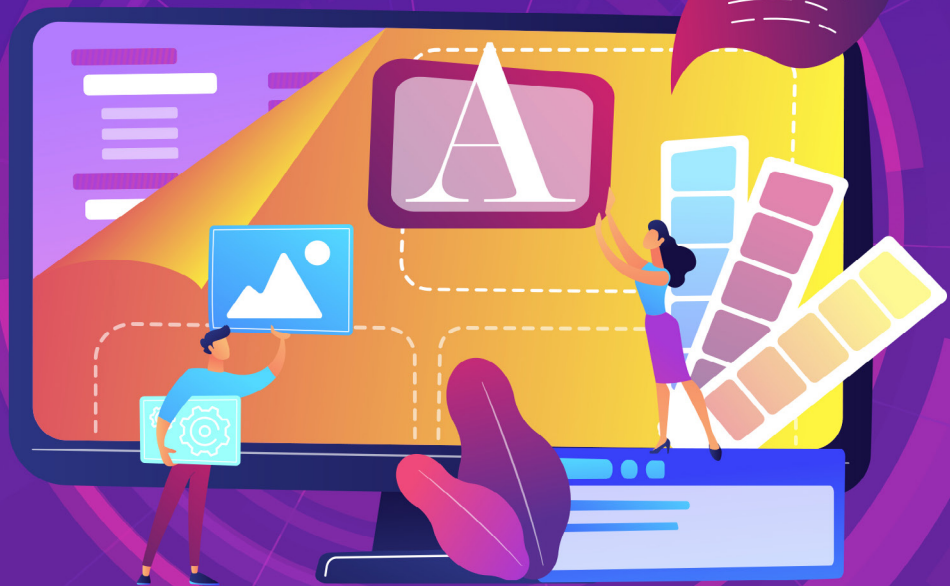


Take Power Apps to the next level: **Overcoming Common Limitations**



What is Power Apps?

Power Apps is an extremely versatile yet complex platform that enables users to build low-code apps that modernize processes and solve tough challenges. Learning how to navigate such a complex platform is critical and necessary in order to successfully utilize this enterprise ready application.

In this article, we will highlight three development challenges users often encounter when working with Power Apps along with a solution for overcoming each.

CHALLENGE 1.

SharePoint has a hard delegation limit where no more than 2,000 rows can be returned when querying a SharePoint list.

SOLUTION

You can imagine trimming a result set to a maximum of 2,000 records can be limiting for many scenarios. Microsoft has put this restriction in place to prevent any one query from taking their servers to their knees and affecting other Power Apps clients. One way to overcome this hurdle is to breakout your query into subqueries where the result set for any one subquery will not be over 2,000 items. Then execute the subqueries (Chunks) concurrently feeding them into a single collection within Power Apps. By following this method, we stay under the delegation limit of 2,000 items for any one query while retaining our full data set.

The key is to break out your subqueries in a way that you can guarantee none of them will reach 2,000 records. If you want an entire list of data, you can create logical batches based on row ID and pull them into a single Power App collection. This method works because we can find out the largest ID in any one list, and logically create subqueries pulling back records within a controlled range each time. Now if you don't want the entire data set or don't know the IDs of your target rows, you have to logically build your subqueries into batches where you can comfortably guarantee any one subquery will not return more than 2,000 rows. There are multiple ways to overcome this limitation, but the key is to understand the data and future-proof the structure of your queries in a way to keep each result set within the delegation limit.

Challenge 2 + 3 on next page



CHALLENGE 2.

User's must press "Allow" and validate their connection every time they launch a Power App.

SOLUTION

Any Power App is likely to contain one or more connections e.g. Outlook, SharePoint, Salesforce etc. Each time one of your users launches the app, they will be asked to verify the connection the app is about to use. While this safety measure is put in place as a stop gap so the user can verify the connections are accessing the right credentials, the likelihood that the user will need to change the cached credentials is very low. This means users must confirm their credentials at every launch which can be frustrating and take away from a seamless experience..

Thankfully, the Power Apps team has created the below PowerShell module that allows for a Global Admin to overwrite this setting.

Step 1. Install the Power Apps module for PowerShell. [Microsoft has published a tutorial for first time setup.](#)

Step 2. Get the ID of your Power App by clicking the ellipses menu next to your app details, then copy the App ID.

Step 3. Run the following cmdlet after replacing 'APPID' with your Power Apps ID.

```
Set-AdminPowerAppApisToBypassConsent -AppName {APPID}
```

It is important to note that Steps 1 through 3 must be repeated every time you republish your Power App.

Once this process has been completed the users will no longer be prompted to allow connections when launching the Power App.

CHALLENGE 3.

Ongoing costs for a Power Apps are too high because every user is required to carry a Premium Power App license.

SOLUTION

Microsoft Office 365 subscriptions come with a "free" Power Apps license for every user. This type of Power Apps license has full functionality except when it comes to Premium connectors. Microsoft categorizes connectors as either "free" or "premium" and if your Power App uses a premium connector, any user launching an app must have a premium Power Apps license associated to their Office 365 account. Depending on the size of your user base, this can render your Power Apps out of reach economically and no longer viable.

One common use case for premium connectors is connecting to external data sources: either databases or through HTTP calls. A possible work around is to first import your data into a source that does not require a premium license such as SharePoint. You can do this through either a one-time import or a scheduled job, and then connect your Power Apps to the new source. This will result in users no longer requiring a premium license when accessing your application.

If you choose to use SharePoint and your data set will be over 5,000 items, be sure to index your columns on the list to avoid any native view limitations. [Microsoft has also published an article on how to properly setup new list indexes.](#)

These are only a few examples of development challenges you will encounter when working with the Power Apps platform. You are likely to come across many other challenges making partnering with a team of experts a key component of success. With our experience and knowledge of the Power Apps platform, the Alitek team can help your organization overcome all these challenges ensuring you fully utilize this enterprise ready application.