

Chapitre 3

Paramètres, délégation et bonnes pratiques

1. Introduction

Dans les chapitres précédents, nous avons créé notre première liste SharePoint, généré automatiquement une application Canvas connectée à cette liste, et commencé à personnaliser les écrans et les formulaires. Nous avons pu constater à quel point Power Apps facilite la transformation de simples données en une application métier fonctionnelle.

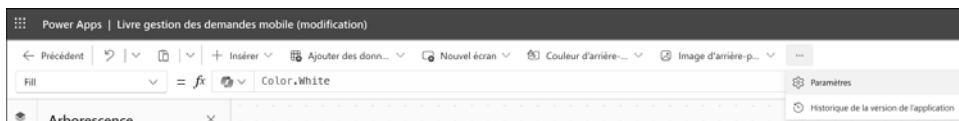
Mais avant de continuer à ajouter de nouvelles fonctionnalités, il est essentiel de prendre un moment pour explorer trois aspects fondamentaux qui déterminent la qualité et la viabilité de nos applications à long terme :

- les paramètres disponibles dans Power Apps Studio ;
- la délégation, concept clé lié aux performances et aux volumes de données ;
- et enfin les bonnes pratiques de conception et de maintenance.

Ces trois éléments sont souvent mis de côté et peu évoqués lors des formations Power Apps. Pourtant, ils font la différence entre une application qui fonctionne « à peu près » et une application robuste, performante, utilisable par des centaines de collaborateurs et maintenable dans le temps. Dans ce chapitre, nous allons donc ralentir le rythme sur la partie développement pour observer les coulisses de Power Apps, comprendre ce qui se joue derrière nos écrans et mettre en place des bases solides pour la suite de notre projet fil rouge.

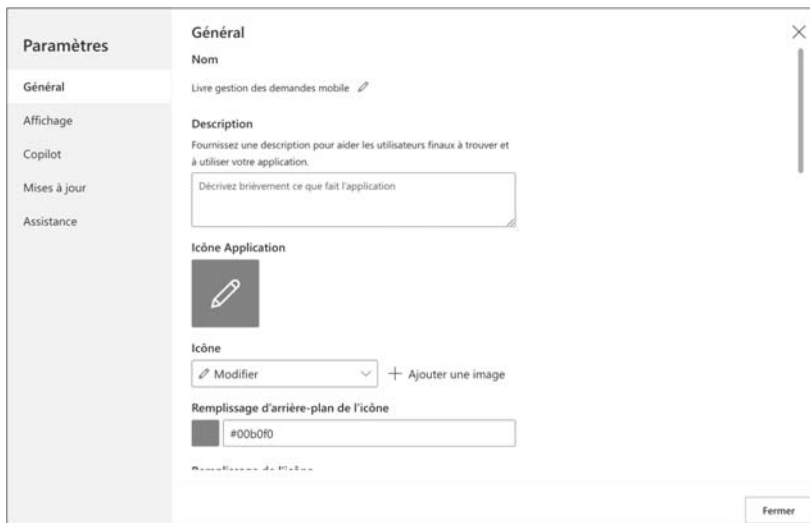
2. Explorer les paramètres de Power Apps Studio

Lorsqu'on ouvre Power Apps Studio, l'attention est souvent portée sur le canevas, les galeries, les formulaires et les contrôles visuels ; en somme, sur tout ce qui constitue la partie *front* de l'application. Pourtant, derrière l'icône en forme de roue dentée, en haut à droite, ou dans les trois petits points au-dessus de la barre de fonction, se cachent des paramètres qui influencent directement le comportement et les limites de notre application.

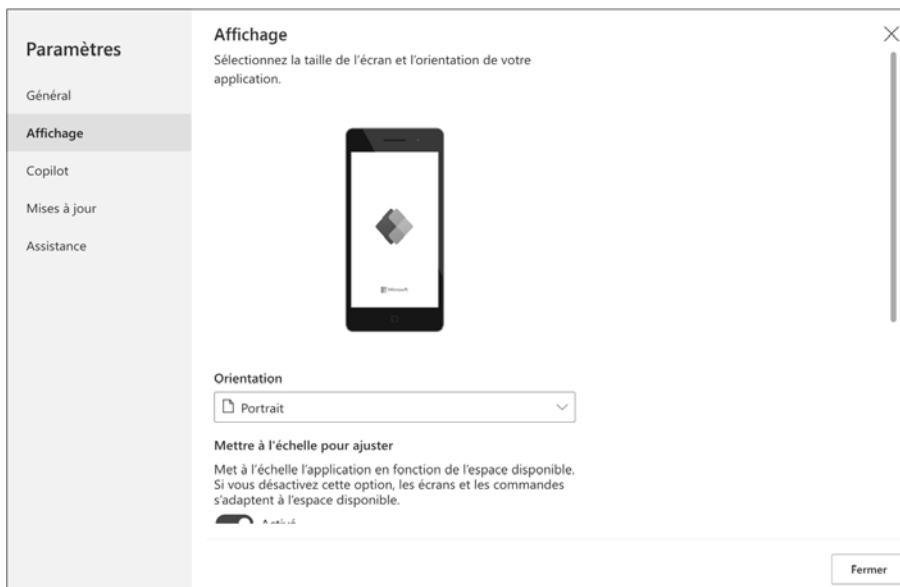


2.1 Paramètres généraux et d'affichage de l'application

Dans l'onglet **Paramètres** et plus précisément dans le volet **Général**, nous pouvons définir le nom de l'application, son icône et sa couleur de thème. Ces éléments peuvent paraître anecdotiques, mais ils contribuent à l'identité visuelle de l'application et à sa reconnaissance par les utilisateurs dans le catalogue d'applications.



Dans l'onglet **Affichage**, nous pouvons définir l'orientation (portrait ou paysage) et la taille d'écran par défaut (mobile ou tablette). Ces choix ne sont pas anodins : une application pensée pour un smartphone ne s'affichera pas de la même manière sur un écran large. Pour notre fil rouge, nous passerons sur une orientation paysage en choisissant une taille tablette à la création de notre prochaine application, car nous imaginons que les collaborateurs saisiront leurs demandes depuis leur ordinateur pour la partie Dataverse.



Astuce - Adapter aux écrans multiples

Il est possible de cocher l'option **Mise à l'échelle réactive**, qui ajuste automatiquement les contrôles en fonction de la taille de l'écran. Cela permet d'éviter les décalages lorsque l'application est ouverte sur une tablette ou sur un PC.

2.2 Paramètres liés aux données

Dans la section **Paramètres - Général**, on trouve une option cruciale : la limite du nombre d'enregistrements. Par défaut, Power Apps lit au maximum 500 enregistrements depuis une source de données comme SharePoint. Il est possible d'augmenter cette limite jusqu'à 2 000.

Cela signifie que si votre liste contient 10 000 éléments, Power Apps n'en chargera que 500 (ou 2 000 si vous modifiez le paramètre). Pour un petit projet, cette limite ne pose pas de problème. Mais pour une application utilisée à grande échelle, il faut apprendre à raisonner autrement, en utilisant la délégation, que nous détaillerons dans la section suivante.

Attention - Une fausse solution

Certains débutants pensent qu'il suffit de mettre la limite à 2 000 pour résoudre leurs problèmes. Mais cette approche atteint vite ses limites : si votre liste dépasse 50 000 éléments, la performance sera catastrophique. Le vrai sujet est d'apprendre à « déléguer » le travail à la source de données.

2.3 Paramètres de performances et médias

Power Apps propose également un paramètre permettant d'activer la mise en cache des données. Cela permet à l'application de fonctionner plus rapidement en gardant temporairement en mémoire les données déjà chargées.

Dans l'onglet **Médias**, on peut également gérer les images et vidéos intégrées à l'application. Il est conseillé de ne pas surcharger l'application de médias lourds, car cela ralentit le chargement, surtout sur mobile.

Astuce - Alléger son application

Préférez héberger vos images dans SharePoint plutôt que de les importer directement dans Power Apps. Votre application sera plus légère et plus rapide à publier.

3. Comprendre la délégation

La délégation est probablement l'un des concepts les plus importants, mais aussi l'un des plus mal compris de Power Apps. Elle désigne la capacité de Power Apps à transmettre une requête à la source de données pour qu'elle exécute elle-même le filtrage, le tri ou l'agrégation.

3.1 Une métaphore pour comprendre

Pour expliquer le concept de délégation de manière plus concrète, nous pouvons imaginer que nous sommes dans une bibliothèque contenant 50 000 livres. Nous cherchons uniquement les romans publiés après 2015.

- **Sans délégation** : nous demandons à l'assistant de nous ramener les 500 premiers livres, puis nous les parcourons un à un pour voir lesquels répondent au critère. Évidemment, nous risquons de passer à côté de milliers d'ouvrages pertinents.
- **Avec délégation** : nous demandons directement à la bibliothécaire de ne sortir que les romans publiés après 2015. Elle connaît toute la base et peut nous donner un résultat complet et fiable.

C'est exactement ce que fait la délégation : au lieu de rapatrier des données partielles, Power Apps envoie la requête à SharePoint (ou SQL, ou Dataverse), qui renvoie uniquement les données pertinentes.

3.2 Les fonctions déléguables

Toutes les fonctions Power Fx ne sont pas déléguables. Les plus courantes, comme **Filter**, **Search**, **SortByColumns** ou encore **Lookup**, peuvent être déléguées, à condition que la source de données le supporte. En effet, la délégation n'est pas qu'une question de fonction mais aussi de connecteurs. Nous allons voir par exemple dans la suite de notre fil rouge que la fonction **Search** n'est pas déléguable avec SharePoint mais elle l'est avec Dataverse.

Dans notre application de gestion des demandes, si nous écrivons une formule comme :

■ `Filter('Gestion des demandes'; Statut.Value = "En cours")`

Alors SharePoint est capable de filtrer lui-même et de ne renvoyer que les demandes en cours.

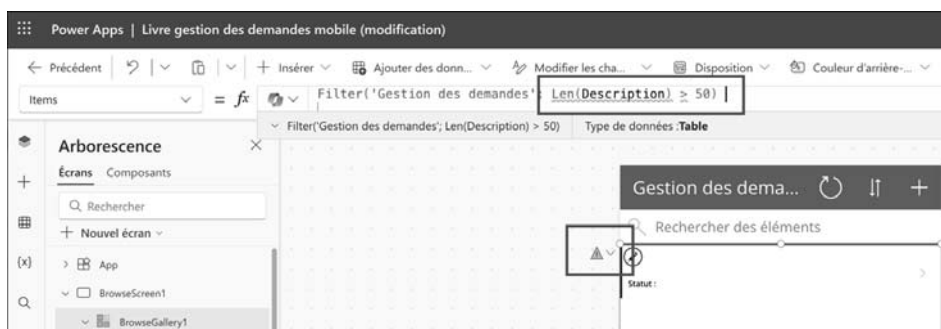
En revanche, si nous écrivons :

■ `Filter('Gestion des demandes'; Len(Description) > 50)`

Power Apps ne peut pas demander à SharePoint de calculer la longueur des descriptions. Il rapatriera donc les 500 premiers éléments, puis appliquera la condition localement. Résultat : le filtre ne sera pas fiable si votre liste contient plus de 500 éléments.

Reconnaître les avertissements de délégation

Lorsque vous saisissez une formule non déléguée, Power Apps affiche un petit triangle bleu avec un point d'exclamation. Ne l'ignorez pas ! C'est l'indicateur que vos résultats risquent d'être tronqués.



3.3 Exemple pratique dans notre fil rouge

Dans l'écran de liste de notre application, nous voulons permettre à l'utilisateur de rechercher des demandes par titre. La formule générée automatiquement est :

```
■ Filter('Gestion des demandes'; StartsWith(Title; TextSearchBox1.Text))
```

StartsWith est une fonction déléguée pour SharePoint. Cela signifie que même si la liste contient 20 000 demandes, la recherche renverra toujours des résultats complets.

Si nous ajoutons une recherche sur la description avec :

```
■ Filter('Gestion des demandes'; StartsWith(Title;  
TextSearchBox1.Text) || StartsWith(Description; TextSearchBox1.Text))
```

Alors Power Apps nous avertira que **StartsWith** appliqué sur plusieurs colonnes n'est pas délégable. Le résultat sera correct uniquement sur les 500 premiers éléments. Attention, le **StartsWith** est souligné et désigné non délégable mais en réalité c'est l'ensemble de l'opérateur **||** et du **StartsWith** sur une autre colonne. Cependant, si je fais plusieurs **StartsWith** sur la même colonne alors la fonction sera délégable. Par exemple, cette formule est délégable :

```
■ Filter('Gestion des demandes'; StartsWith(Title; "f") || StartsWith(Title; "t"))
```

Le sens métier n'est pas forcément pertinent mais démontre le fonctionnement de la fonction. Ici, nous recherchons toutes les demandes dont le titre commence par **f** ou commence par **t**.

4. Bonnes pratiques pour des applications performantes et maintenables

Une fois les paramètres compris et la délégation assimilée, il est temps d'aborder un troisième pilier : les bonnes pratiques. Ces recommandations ne sont pas des obligations, mais elles vous feront gagner en performance, en lisibilité et en fiabilité.

4.1 Structurer et nommer ses éléments

Dès le début du projet, prenez l'habitude de renommer vos contrôles. Au lieu de garder `TextInput1`, utilisez `txtRechercheTitre`. Cela semble fastidieux, mais vous vous remercirez lorsque vous écrirez des formules plus complexes.

De même, organisez vos écrans et vos collections avec des noms clairs. Cela rendra l'application compréhensible par un autre développeur (ou par vous-même dans six mois).

4.2 Limiter les colonnes inutiles

Lorsque vous connectez une galerie à une liste SharePoint, Power Apps charge par défaut toutes les colonnes. Or, chaque colonne supplémentaire alourdit le transfert. Pensez à réduire le nombre de champs affichés en utilisant la fonction `ShowColumns` si vous souhaitez lister les colonnes à garder ou `DropColumns` si vous souhaitez lister les colonnes à retirer.

Par exemple, si je ne veux garder que les colonnes **Titre** et **Services**, je peux écrire cette formule dans la propriété `Items` de ma Galerie :

```
■ ShowColumns('Gestion des demandes';Titre; Service)
```