

Power Automate Cloud-Flüsse

Bewährte Praktiken

Weißbuch

Zusammenfassung: Dieses technische Whitepaper richtet sich an Microsoft Power Automate-Macher in Unternehmen. Es enthält die besten Praktiken für den Aufbau effizienter Power Automate Cloud-Flows.

Die Autoren: Reza Dorrani, Rasika Chaudhary, Foo Shen Wu

Inhalt

Einführung	3
Zweck des Weißbuchs	4
Umfang des Whitepapers.....	4
Best Practices für Cloud-Flows	4
Konsistente Namensgebung für Flow-Komponenten	4
Lösungsorientierte Cloud-Flows vs. nicht lösungsorientierte Cloud-Flows	5
Geltungsbereiche erstellen	9
Komplexe Geschäftslogik auslassen	10
Wiederverwendbaren Code erstellen (Child flows)	10
Parallele Ausführung und Gleichzeitigkeit (innerhalb der For each-Schleife).....	12
Arbeiten mit Datenoperationen.....	17
Leistung optimieren Auslöser automatisieren	20
Arbeiten mit relevanten Daten	22
Batch verwenden, wo möglich, maximale Seitengröße	25
Verschachtelte For-Each-Schleifen vermeiden	27
Unendliche Läufe vermeiden	29
Asynchrones Ablaufmuster	32
Umgebungsvariablen.....	33
Fehlerbehandlung	34
Durchfluss Abschaltverhalten	42
Flow Checker	43
Cloud Flow Testing.....	44
Wiedervorlage von Cloud-Flow-Läufen	45
Profi-Tipps	46
Peek-Code/Code-Ansicht	46
Eine Notiz hinzufügen	48
Kopieren Einfügen Aktionen	49
Fehlerbehebung bei Cloud-Flüssen	50
Plattformgrenzen verstehen / Drosselung vermeiden	51
Wie Sie Ihre Lizenzpläne überprüfen können	51

Grenzen für API-Anfragen	52
API-Durchsatzgrenzen für Konnektoren	54
Grenzwerte für die Gleichzeitigkeit der Ströme	55
Aktion Burst-Grenzen	56
Zusätzliche Grenzwerte	57
Sichere Abläufe entwerfen	58
Sichere Eingaben/Ausgaben verwenden	58
Absicherung des HTTP-Anfrageauslösers	61
Flow-Konfiguration generisch halten	62
Konfigurieren von AAD/Entra-Zugangsberechtigungsrichtlinien	63
Verständnis des Zugangs zu Strömen	63
Verwendung von Auditing in Dataverse und Purview	65
Verwendung von Richtlinien für Datenverlust	66
CMK (Customer Managed Keys) für Cloud-Flows	66
Bewährte ALM-Verfahren für Power Automate	67
Verstehen, wann Flows mit Power Apps integriert werden sollten	68
Wann Sie Power Fx vs. Power Automate in Power Apps verwenden sollten	69
Überwachung und Alarmierung	71
Leverage Process Mining (Vorschau)	71
Power Automate Analytics	73
Power Automate Management Steckverbinder	75
Power Automate App Insights	77
Verlauf des Flusslaufs in Dataverse	79
Automatisierungszentrum	71

Einführung

Microsoft Power Automate bietet eine leistungsstarke Suite von Automatisierungsfunktionen, die es Unternehmen ermöglichen, Prozesse zu rationalisieren, sich wiederholende Aufgaben zu automatisieren und neue Wachstumschancen zu erschließen.

Microsoft Power Automate, Teil der Microsoft Power Platform, bietet eine umfassende Automatisierungslösung, die es Unternehmen ermöglicht, Arbeitsabläufe zu automatisieren, Daten und Anwendungen zu integrieren und künstliche Intelligenz zu nutzen, um Effizienz und Innovation zu fördern. Power Automate bietet eine benutzerfreundliche Oberfläche und eine Vielzahl von Konnektoren, die eine Automatisierung ohne umfangreiche Programmierkenntnisse ermöglichen.

Durch einfaches Ziehen und Ablegen von Komponenten und Konfigurieren von Einstellungen können Benutzer schnell leistungsstarke Workflows erstellen, die Zeit sparen, die Effizienz steigern und die Produktivität erhöhen. Von

Von einfachen Benachrichtigungsabläufen bis hin zu fortschrittlichen mehrstufigen Prozessen bietet Power Automate die Flexibilität und Skalierbarkeit, um ein breites Spektrum an Automatisierungsanforderungen zu erfüllen, was es zu einem unverzichtbaren Werkzeug für moderne digitale Arbeitsplätze macht.

[Dokumentation zu Microsoft Power Automate](#)

Zweck des Weißbuchs

Dieses vom Power CAT Team herausgegebene Dokument enthält wichtige Anleitungen für Entwickler, die Power Automate-Flows in verschiedenen Umgebungen entwerfen, erstellen, testen, bereitstellen und warten, darunter kleine Unternehmen, Konzerne und Behörden.

Die Erstellung eines einfachen Cloud-Flows in Power Automate ist einfach, aber wenn die Anforderungen komplexer werden, wird die Aufrechterhaltung der Codequalität entscheidend. Bei der Fülle an Informationen, die im Internet über Power Automate verfügbar sind, kann es schwierig sein, die besten Praktiken zu erkennen. [Dieses Whitepaper fasst Erkenntnisse von Power Automate-Experten aus der ganzen Welt zusammen und stellt sicher, dass Sie Ihre Abläufe/Automatisierungen korrekt und effizient erstellen.](#)

Umfang des Whitepapers

Power Automate-Flows umfassen Cloud-Flows, Desktop- und Geschäftsprozess-Workflows. In diesem Whitepaper werden die Best Practices für **Cloud-Flows** beschrieben.

Best Practices für Cloud-Flows

Wenn Sie Microsoft Power Automate zur Automatisierung von Aufgaben verwenden, ist es wichtig, dass alles reibungslos und schnell abläuft. Das bedeutet, dass Sie einige intelligente Vorgehensweisen, die so genannten "Best Practices", befolgen müssen. Diese Praktiken tragen dazu bei, dass Ihre Cloud-Abläufe schnell, zuverlässig und effizient sind. In diesem Artikel werden die Best Practices in die Kategorien Entwurfszeit und Laufzeit eingeteilt. Einige der besten Entwurfspraktiken sind die folgenden:

Konsistente Namensgebung für Flow-Komponenten

Die Beibehaltung konsistenter Namenskonventionen für die Komponenten in Ihren Power Automate-Abläufen ist entscheidend für Klarheit, Organisation und einfache Verwaltung. Durch die Einführung eines standardisierten Ansatzes für die Benennung können Sie die Zusammenarbeit zwischen den Teammitgliedern verbessern, die Fehlerbehebung vereinfachen und die Entwicklung und Wartung Ihrer Workflows rationalisieren.

Hier sind einige Leitlinien zu beachten:

Hinweis: Es handelt sich um Empfehlungen, die sich je nach den Anforderungen der einzelnen Unternehmen ändern können.

1. **Beschreibende und aussagekräftige Namen:** Geben Sie Ihren Abläufen aussagekräftige Namen, bevor Sie sie speichern. Wählen Sie außerdem Namen, die den Zweck oder die Funktion der einzelnen Komponenten genau beschreiben. Vermeiden Sie allgemeine oder zweideutige Namen, die zu Verwechslungen führen könnten. Benennen Sie beispielsweise einen Auslöser nicht mit "Auslöser1", sondern verwenden Sie einen beschreibenden Namen wie "Neue E-Mail erhalten", um den Zweck des Auslösers deutlich zu machen.

2. **Verwenden Sie CamelCase oder Unterstriche:** Verwenden Sie CamelCase (Großschreibung des ersten Buchstabens eines jeden Wortes außer dem ersten) oder Unterstriche, um Wörter in Ihren Komponentennamen zu trennen. Dies verbessert die Lesbarkeit und erleichtert die Unterscheidung zwischen verschiedenen Teilen des Namens. Sie könnten zum Beispiel eine Aktion "SendEmailNotification" oder "send_email_notification" nennen, um die Konsistenz zu gewährleisten.
3. **Präfixe oder Tags:** Ziehen Sie die Verwendung von Präfixen oder Tags in Betracht, um Komponenten auf der Grundlage ihres Typs oder ihrer Funktionalität zu kategorisieren. Sie könnten zum Beispiel Präfixe wie "Trg_" für Auslöser, "Act_" für Aktionen oder "Var_" für Variablen verwenden. Auf diese Weise lässt sich die Rolle der einzelnen Komponenten innerhalb des Ablaufs schnell erkennen.
4. **Konsistenz über Abläufe hinweg:** Sorgen Sie für konsistente Namenskonventionen in all Ihren Power Automate-Abläufen, um die Navigation und Standardisierung zu erleichtern. Dadurch wird sichergestellt, dass Teammitglieder die verschiedenen Abläufe leicht verstehen und mit ihnen arbeiten können, ohne dass sie unterschiedliche Benennungsstile entziffern müssen.
5. **Dokumentieren Sie die Benennungskonventionen:** Dokumentieren Sie Ihre Namenskonventionen in einem Styleguide oder einer Dokumentation, um sicherzustellen, dass alle Teammitglieder die zu befolgenden Standards kennen. Dies fördert die Einheitlichkeit und verringert die Wahrscheinlichkeit von Unstimmigkeiten oder Missverständnissen.
6. **Kommentare hinzufügen** - Das Hinzufügen von Kommentaren zu den Aktionen erleichtert das Verständnis der Logik der Flussimplementierung.

Lösungsorientierte Cloud-Flows im Vergleich zu nicht lösungsorientierten Cloud-Flows

Cloud-Flows, die aus einer [Lösung](#) heraus erstellt oder in eine Lösung eingefügt werden, werden als

lösungsorientierte Cloud-Flows oder Solution Cloud-Flows bezeichnet. Lösungsorientierte Cloud-Flows sind portabel, verwaltbar und können leicht von einer Umgebung in eine andere verschoben werden, was starke ALM-Praktiken ermöglicht. Sie können mehrere Flows zu einer einzigen Lösung hinzufügen.

Um besser zu verstehen, wie man Abläufe in einer Lösung erstellt, lesen Sie das folgende [Dokument](#). Darüber hinaus gibt es einige [bekannte Einschränkungen](#) bei lösungsorientierten Abläufen, die die Benutzer kennen sollten.

Verwenden Sie Pipelines in Power Platform zur Bereitstellung von Lösungen: Bietet die einfache Bereitstellung von Lösungen zu Test- und Produktionszwecken. Sobald die Pipelines eingerichtet sind, können Hersteller mit wenigen Klicks produktinterne Bereitstellungen initiieren. Die Hersteller tun dies direkt in ihren Entwicklungsumgebungen. Weitere Informationen: [Überblick über die Pipelines in Power Platform](#)

Wenn Abläufe in einer Lösung hinzugefügt werden, werden die Metadaten der Abläufe in Dataverse in der Tabelle Process (Workflow) gespeichert.

Nicht lösungsgebundene Cloud-Flows hingegen sind die Flows, die außerhalb der Lösungen erstellt werden und einzeln verwaltet und bereitgestellt werden müssen.

Verbindungsreferenzen vs. Verbindungen

In Microsoft Power Automate beziehen sich die Begriffe "Verbindungsreferenzen" und "Verbindungen" auf unterschiedliche Aspekte der Interaktion von Flows mit externen Diensten und Datenquellen. Das Verständnis der Unterscheidung zwischen den beiden Begriffen ist für die effektive Verwaltung und Bereitstellung von Flows unerlässlich, insbesondere im Kontext von lösungsorientierten Cloud-Flows und nicht lösungsorientierten Cloud-Flows.

Verbindungen (nicht lösungsorientierte Cloud-Flows) - Verbindungen sind Konfigurationen, die eine Verbindung zwischen Power Automate und externen Diensten herstellen, z. B. Office 365, SharePoint, Dynamics 365, Twitter oder einem anderen Connector. Eine Verbindung enthält Authentifizierungsdetails (wie API-Schlüssel, OAuth-Tokens oder Benutzeranmeldeinformationen), die für den Zugriff auf den externen Dienst erforderlich sind.

Wenn Sie einen Ablauf erstellen, müssen Sie eine Verbindung zu dem Dienst, mit dem Sie interagieren möchten, erstellen oder auswählen. Verbindungen sind benutzerspezifisch, d. h. sie sind an die Anmeldedaten des Benutzers gebunden, der sie erstellt.

Wenn Sie eine Verbindung für einen Cloud-Flow ändern, muss jede Aktion, die mit der Verbindung verbunden ist, aktualisiert werden.

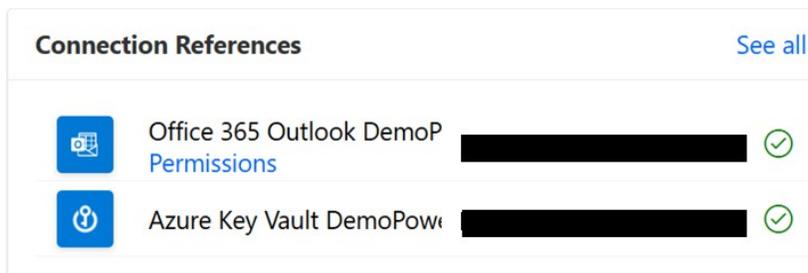


Verbindungsreferenzen (lösungsorientierte Cloud-Flows) - Verbindungsreferenzen sind eine Abstraktionsschicht, die die Verwaltung von Verbindungen innerhalb einer Power Platform-Lösung vereinfacht. Sie fungieren als Platzhalter oder Zeiger auf tatsächliche Verbindungen und erleichtern so die Verwaltung und Aktualisierung von Verbindungen in lösungsorientierten Szenarien.

Verbindungsreferenzen werden in lösungsorientierten Cloud-Flows verwendet, die Teil einer umfassenderen Power Platform-Lösung sind (z. B. solche, die mit Power Apps, Dataverse oder anderen Komponenten integriert sind). Sie ermöglichen die Trennung des Flow-Designs von den spezifischen Benutzerverbindungen und erleichtern so die Bereitstellung und Aktualisierung in verschiedenen Umgebungen (Entwicklung, Test, Produktion).

Da Verbindungsreferenzen eine Umhüllung für Verbindungen sind, muss nur die Verbindungsreferenz aktualisiert werden, wenn die Verbindung für Aktionen innerhalb eines lösungsorientierten Cloud-Flows aktualisiert werden muss.

Alle Aktionen innerhalb des Ablaufs beziehen sich automatisch auf die aktualisierte Verbindung in der Verbindungsreferenz.



Begrenzung der Anzahl der Bewegungen, die ein Benutzer besitzt

Bei nicht lösungsbezogenen Cloud-Strömen kann ein Benutzer maximal 600 Ströme besitzen. Diese Grenze gilt nicht für Bewegungen, die innerhalb von Lösungen erstellt wurden.

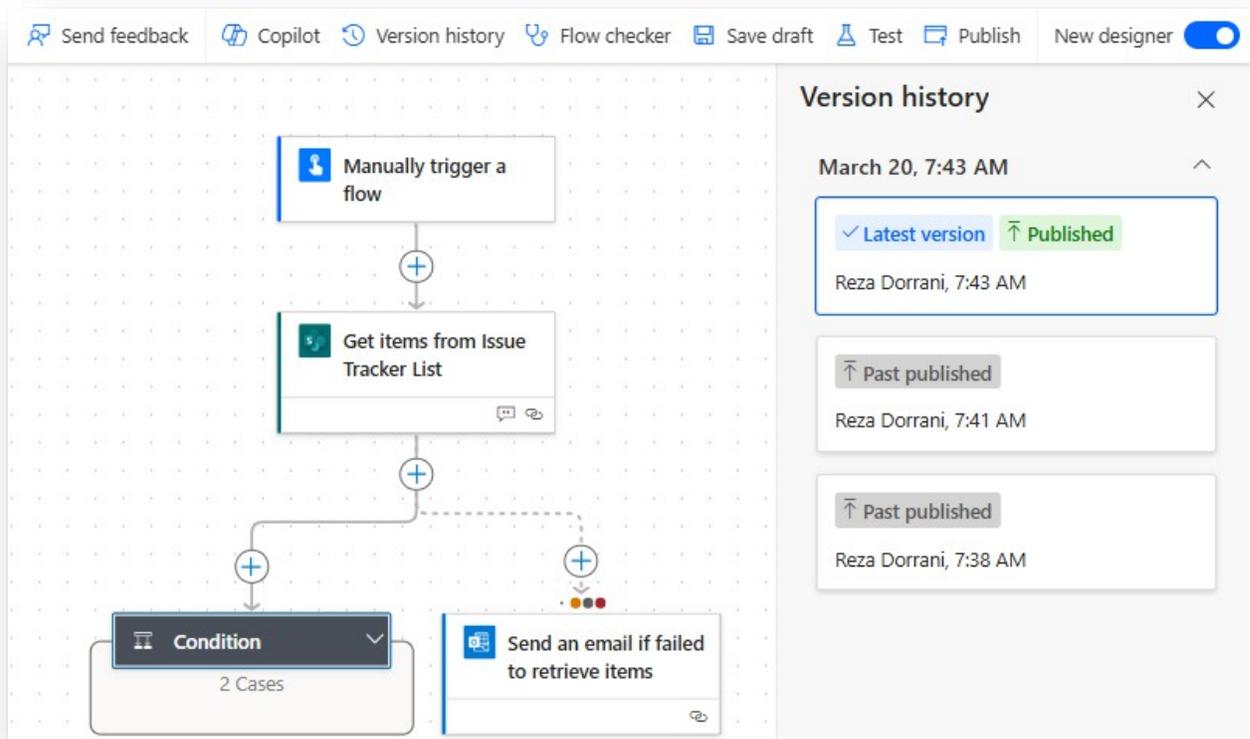
[Hinweis auf die Dokumentation](#)

Flussversionierung

Die Versionierung von Abläufen ist eine neue Funktion, die nur für lösungsorientierte Cloud-Abläufe verfügbar ist, bei denen Benutzer Entwürfe von Abläufen erstellen können, während sie die Ablauflogik weiterentwickeln und nach Fertigstellung veröffentlichen. Die Versionsinformationen (Metadaten des Ablaufs) werden in Dataverse gespeichert.

Sie ermöglicht es dem Flow Maker, Änderungen zu verfolgen, bei Bedarf zu früheren Versionen zurückzukehren und sicherzustellen, dass Ihre Flows stabil und zuverlässig sind.

In der folgenden Dokumentation finden Sie weitere Informationen zur [Flow-Versionierung](#)



Umgebungsvariablen

Umgebungsvariablen ermöglichen das grundlegende Szenario des Application Lifecycle Management (ALM), bei dem eine Anwendung zwischen Power Platform-Umgebungen verschoben wird. In diesem Szenario bleibt die Anwendung bis auf einige wichtige externe Anwendungsreferenzen (z. B. Tabellen, Verbindungen und Schlüssel), die sich zwischen der Quell- und der Zielumgebung unterscheiden, identisch. Die Anwendung setzt voraus, dass die Struktur der Tabellen oder Verbindungen zwischen der Quell- und der Zielumgebung mit einigen Unterschieden genau gleich ist. Mit Hilfe von Umgebungsvariablen können Sie angeben, welche dieser unterschiedlichen externen Referenzen aktualisiert werden sollen, wenn die Anwendung zwischen den Umgebungen verschoben wird

Nur lösungsorientierte Cloud-Flows (ALM-freundlich) unterstützen die Verwendung von

Umgebungsvariablen. [Mehr erfahren](#)

Verwalten des Verlaufs von Cloud-Flows in Dataverse

Mit der Cloud-Flow-Laufhistorie in Dataverse können Sie die Erweiterbarkeit von [Dataverse](#) nutzen, um die Ergebnisse Ihrer Cloud-Flow-Ausführungen im Maßstab zu verfolgen.

Nur Solution Cloud-Flows, deren Definitionen in Dataverse liegen, können ihre Laufhistorie in Dataverse speichern.

Im Rahmen dieser Funktion hat jede Cloud-Flow-Ausführung einen Eintrag in der Tabelle [FlowRun](#). Dieses Merkmal ist

Verwendung der nicht-relationalen Datenbank von Dataverse, [Elastic Tables](#), zur Speicherung der Cloud-Flow-Laufhistorie. Die Cloud-Flow-Laufhistorie in Dataverse wird vom [Automatisierungszentrum](#) verwendet, um umfassende Überwachungs- und Fehlerbehebungsfunktionen für Automatisierungsprozesse in Power Automate bereitzustellen.

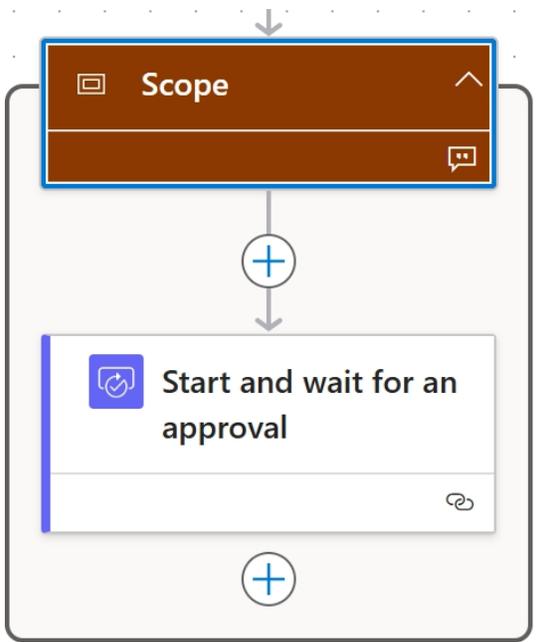
Standardmäßig werden die Daten eines Ablaufs für 28 Tage (2.419.200 Sekunden) gespeichert. Wenn Sie die Dauer, wie lange die Ausführungen gespeichert werden können, ändern möchten, können Sie die ["Time to live" \(in Sekunden\) für den Ablauf in der Tabelle "Organisation"](#) in einer mit Dataverse gesicherten Umgebung aktualisieren. Abhängig von Ihren der Speicherkapazität der Umgebung, können Sie die Speicherdauer für diese Laufprotokolle anpassen

[Mehr erfahren](#)

Geltungsbereiche erstellen

Da Arbeitsabläufe immer komplizierter werden, ist es wichtig, sie gut zu verwalten, um Probleme zu bewältigen, sie zu testen und sie reibungslos funktionieren zu lassen. Ein Scope ist wie ein Container, der ein Bündel von Aktionen zusammenhält.

So können Sie die einzelnen Teile Ihres Arbeitsablaufs leichter organisieren und sehen. Wenn Ihr Arbeitsablauf größer und komplexer wird, können Sie mit dem Umfangswerkzeug leichter Ordnung halten und auftretende Probleme beheben.



Die Funktionsweise von Geltungsbereichen ist wie folgt:

1. **Aktionen gruppieren:** Sie können mehrere Aktionen innerhalb eines Bereichs hinzufügen. Dies ist nützlich, wenn Sie eine Reihe von Aktionen haben, die miteinander

verbunden sind oder zusammen ausgeführt werden müssen.

2. **Organisation Ihres Ablaufs:** Durch die Verwendung von Bereichen können Sie eine hierarchische Struktur innerhalb Ihres Ablaufs erstellen. Sie können verschachtelte Bereiche haben, in denen ein Bereich einen anderen Bereich enthält, so dass Sie Ihren Arbeitsablauf in kleinere, besser zu verwaltende Abschnitte unterteilen können.

3. **Fehlerbehandlung:** Scopes sind auch für die Fehlerbehandlung hilfreich. Sie können Bereiche so einrichten, dass sie Fehler abfangen und behandeln, die innerhalb der darin enthaltenen Aktionen auftreten. Dies erleichtert die Fehlersuche und die Behebung von Problemen in Ihrem Arbeitsablauf. [Die Fehlerbehandlung wird hier ausführlicher behandelt](#)
4. **Sichtbarkeit und Ausklappbarkeit:** Bereiche können eingeklappt werden, um ihren Inhalt auszublenden, so dass Sie sich leichter auf andere Teile Ihres Ablaufs konzentrieren können. So bleibt Ihr Arbeitsbereich übersichtlich und organisiert, insbesondere bei der Arbeit mit großen und komplexen Abläufen.
5. **Vermeiden Sie die Verwendung von Abläufen für jede einzelne Aktion:** Nicht jede Aktion muss in einen Bereich fallen. Achten Sie darauf, dass Sie nicht zu viele Aktionen in einem Bereich durchführen, wenn es nicht notwendig ist.

Hinweis: Es ist ratsam, Kommentare oder Notizen in den Geltungsbereich einzufügen, um die Abläufe besser lesbar zu machen.

Komplexe Geschäftslogik auslassen

Wir haben schon oft Anwendungsfälle gesehen, in denen Millionen von Datensätzen mit Hilfe von Flows verarbeitet werden müssen. Im Allgemeinen können Abläufe zwar zur Automatisierung alltäglicher Aufgaben verwendet werden, aber für eine groß angelegte Datenumwandlung oder -integration ist Power Automate möglicherweise nicht das richtige Werkzeug. Hier sind einige Empfehlungen.

Wenn Sie eine große Anzahl von Dataverse-Datensätzen verschieben oder verarbeiten müssen, sollten Sie Dataverse-Plugins als benutzerdefinierte Aktionen verwenden. Dadurch können Benutzer Aktionen mit geringem Code in Dataverse für die Nutzung von Power Fx verwenden.

[Im folgenden Abschnitt](#) erfahren Sie mehr darüber, wann Sie Abläufe mit anderen Lösungen integrieren sollten, um die Komplexität zu verringern

Darüber hinaus können Sie benutzerdefinierte Konnektoren für eigenständige Dienste wie Azure Functions, API Management, Azure App Services oder benutzerdefinierte Konnektoren mit benutzerdefiniertem Code verwenden.

Wiederverwendbaren Code erstellen (Child flows)

Power Automate Cloud-Flows können bei der Automatisierung komplexer Lösungen helfen. Allerdings können sie schnell zu groß und komplex werden, sodass es schwierig wird, den Flow zu navigieren und zu pflegen. Anstatt eine komplexe einzelne Automatisierungslösung zu erstellen, sollten Sie untergeordnete Abläufe nutzen, die die Komplexität, Tiefe und Effizienz dieser Automatisierungsprozesse erhöhen. Sie ermöglichen Ihnen die Aufgliederung Komplexe Prozesse werden in überschaubare Teile zerlegt, wodurch die Abläufe modularer und leichter zu verwalten sind.

Warum Child Flows verwenden?

- **Modularität:** Untergeordnete Abläufe fördern die Modularität und ermöglichen es Ihnen, wiederverwendbare Komponenten zu erstellen, die leicht in verschiedene Arbeitsabläufe integriert werden können. Dadurch wird nicht nur die Entwicklung rationalisiert, sondern auch die Wartungsfreundlichkeit verbessert.
- **Skalierbarkeit:** Indem Sie Ihre Automatisierung in kleinere, besser handhabbare Teile zerlegen, können Sie Ihre Prozesse effektiver skalieren. Müssen Sie Änderungen vornehmen oder Funktionen hinzufügen? Mit untergeordneten Abläufen ist es so einfach wie die Aktualisierung der einzelnen Komponenten.

- **Granulare Kontrolle:** Untergeordnete Abläufe bieten eine granulare Kontrolle über Ihre Automatisierungslogik. Sie können Daten zwischen übergeordneten und untergeordneten Abläufen weitergeben und so dynamische Entscheidungen und personalisierte Erlebnisse ermöglichen.
- **Kollaboration:** In einer Teamumgebung fördern Child Flows die Zusammenarbeit, indem sie es den Teammitgliedern ermöglichen, gleichzeitig an separaten Komponenten eines größeren Automatisierungsprojekts zu arbeiten.

Hinweis: Sie sollten die übergeordnete Bewegung und alle untergeordneten Bewegungen *direkt* in derselben Lösung erstellen. Beachten Sie die [bekannten Einschränkungen](#).

Beispiel für ein Parent/Child-Flow-Szenario - Die Organisation Contoso benötigt Genehmigungen für Szenarien wie Projektvorschläge, Urlaubsanträge und Ausgabenanträge. Die Genehmigenden sind Teil verschiedener SharePoint-Gruppen.

Übergeordnete Abläufe - In diesem Szenario kann jeder übergeordnete Ablauf wie Projektvorschlag, Urlaubsantrag oder Spesenabrechnung einen einzelnen wiederverwendbaren untergeordneten Ablauf aufrufen, um die Liste der SharePoint-Genehmiger aus den SP-Gruppen zu extrahieren.

Kinderfluss -

The image shows a screenshot of a Power Automate flow configuration interface on the left and its visual flow diagram on the right.

Configuration Interface (Left):

- Title:** GetGroupMembers
- Parameters:**
 - Site Address *: `https://thepoweraddicts.sharepoint.com/sites/HR`
 - Method *: `GET`
 - Uri *: `_api/web/siteGroups/getByName('ApprovingGroupNa...')/Users`
- Advanced parameters:** Showing 0 of 2. Buttons: Show all, Clear all.
- Connection:** Connected to SharePoint. [Change connection reference](#)

Visual Flow Diagram (Right):

- 1. Manually trigger a flow
- 2. GetGroupMembers (Action)
- 3. Filter array
- 4. Respond to a Power App or flow

Elterlicher Fluss -

The screenshot shows the configuration interface for a 'Get Approvers' connector in Power Automate. On the left, the 'Parameters' tab is active, showing a 'Child Flow' dropdown set to 'ChildFlowTogetSP Group members' and an 'ApprovingGroupName' field set to 'Expense approvers'. Below this, an 'Advanced parameters' section shows 'Showing 0 of 1' with 'Show all' and 'Clear all' buttons. On the right, a workflow preview shows a sequence of steps: 'When a new expense is submitted' (trigger), followed by 'Get Approvers' (connector), and then 'Start and wait for an approval' (action). Each step is connected to the next by a downward arrow with a plus sign in a circle, indicating a sequential flow.

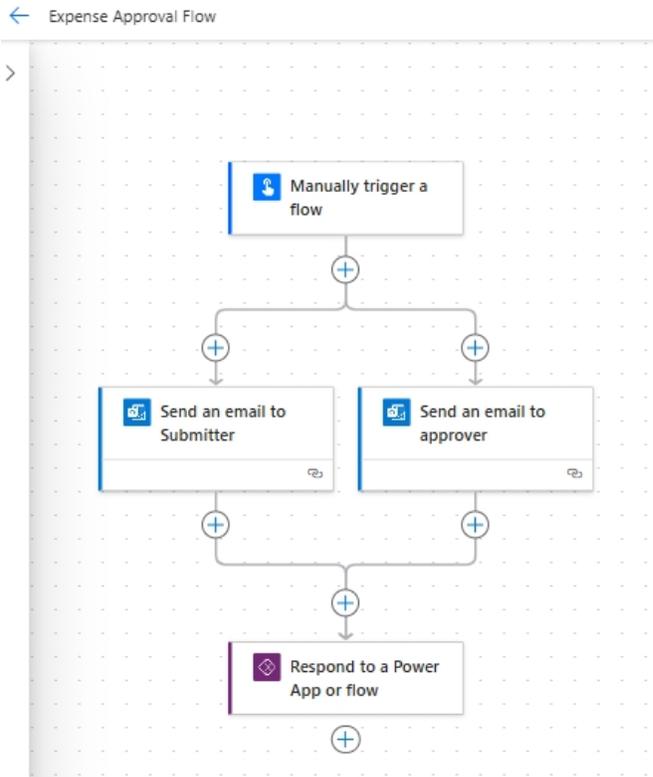
Weitere Einzelheiten zu Child Flows finden Sie unter folgendem Link - [Child Flows](#)

Parallele Ausführung und Gleichzeitigkeit (innerhalb der For each-Schleife)

Parallele Ausführung

Power Automate unterstützt die parallele Ausführung, d. h. Abläufe können zwei oder mehr Schritte haben, die gleichzeitig ablaufen, und der Workflow wird erst berechnet, wenn alle parallelen Schritte abgeschlossen sind.

Dies ermöglicht es Flow, mehr Aufgaben gleichzeitig zu verarbeiten, insbesondere wenn die Aufgaben asynchron sind.

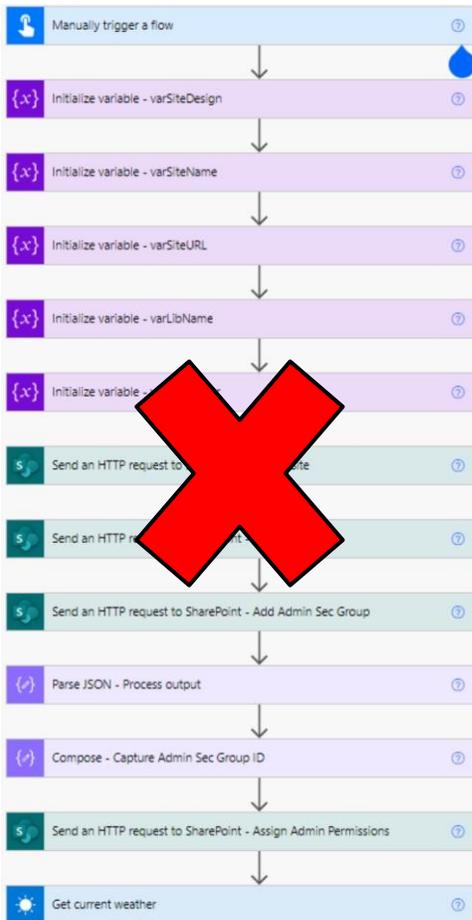


Eine gute Faustregel ist, nur dann zu parallelisieren, wenn **die Ausführung von Aktionen mehr als 5 Sekunden dauert**. Sie können parallele Verzweigungen verwenden, um einige der folgenden Szenarien zu erreichen:

- Senden von nicht blockierenden Genehmigungsanfragen ([Blogpost](#))
- Erstellung von Genehmigungen auf der Grundlage von "Quorum" ([Blogbeitrag](#))
- Erstellen oder Aktualisieren von Datensätzen in mehreren Systemen
- Daten aus verschiedenen Quellen abrufen und in einer einzigen konsolidieren

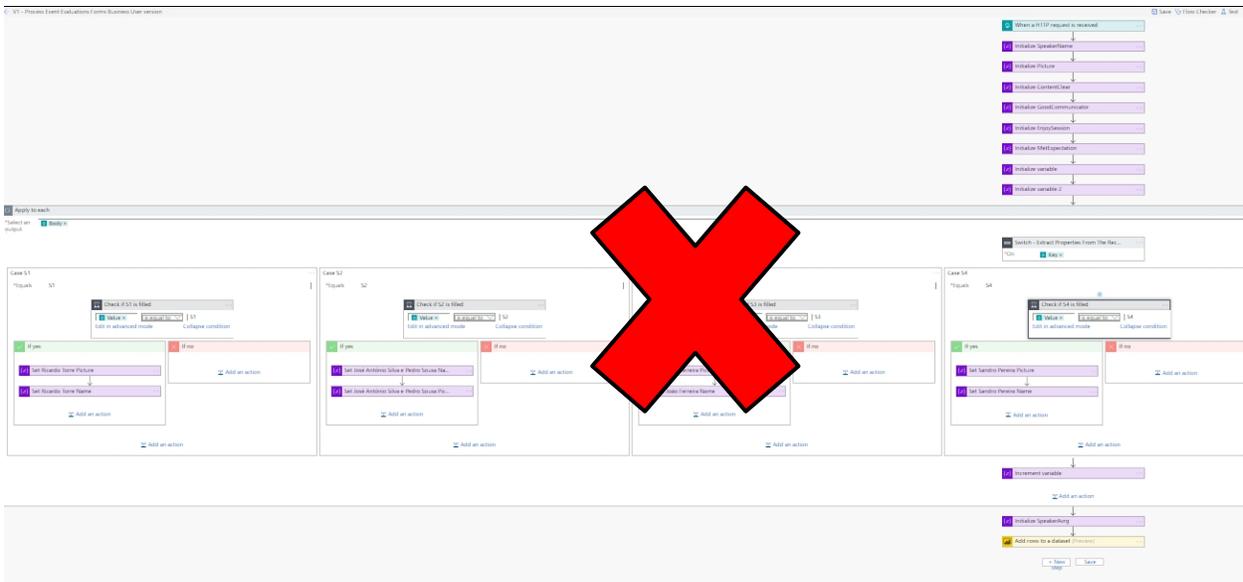
Reduzierung des Planungsaufwands

Geradlinige Aktionen können in der Maschine sequentiell ablaufen. Eine parallele Strukturierung kann helfen, den Ablauf besser zu organisieren und Zeit zu sparen. Insbesondere die Initialisierung mehrerer Variablen kann parallel durchgeführt werden, um Zeit zu sparen.



Hinweis: Eine Sache, die Sie immer im Auge behalten sollten, ist die Vermeidung einer großen Anzahl von übersprungenen Aktionen. Sehr breite Switch-Anweisungen mit vielen Aktionen in jeder Verzweigung oder parallele Verzweigungen, bei denen der weniger genutzte Pfad deutlich mehr Aktionen enthält und die Lesbarkeit des Ablaufs beeinträchtigt

Abhilfe: Die Verwendung von Child Flows kann dieses Problem lösen. Anstatt viele übersprungene Aktionen in einem Switch-Zweig zu haben, rufen Sie stattdessen Child Flows aus dem Switch-Zweig auf, so dass wir die irrelevanten Child Flows anstelle einer beträchtlichen Anzahl von Aktionen überspringen. Das macht die Wartung einfacher.



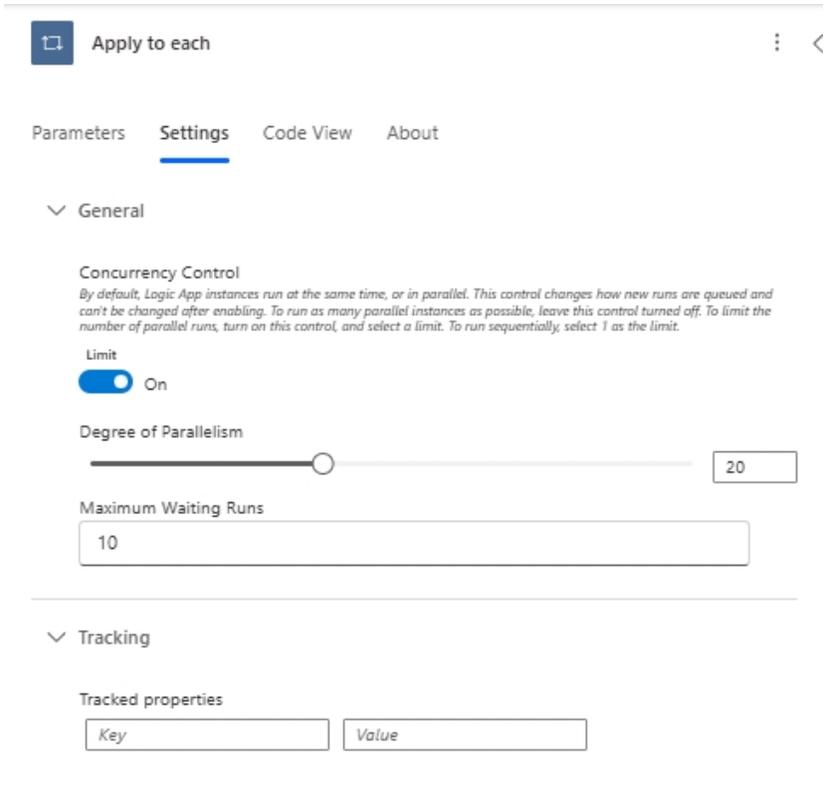
Gleichzeitigkeitskontrolle

Gleichzeitigkeit ermöglicht die parallele Ausführung in For Each-Schleifen. Standardmäßig wird die For Each-Schleife sequentiell ausgeführt. Bei der Verarbeitung großer Daten kann dies daher viel Zeit in Anspruch nehmen. Wenn die Elemente in der Schleife nicht sequentiell ausgeführt werden müssen, ermöglicht die Parallelität die gleichzeitige Verarbeitung von X Elementen. Derzeit können die Benutzer den Grad der Parallelität von 1 bis 50 einstellen.

Nehmen wir ein Szenario, in dem 100 Datensätze mit dem Status "Aktiv" aktualisiert werden müssen. Anstatt einen Datensatz nach dem anderen zu aktualisieren, ermöglicht die Gleichzeitigkeitssteuerung die gleichzeitige Aktualisierung von bis zu 50 Datensätzen.

Achten Sie auf den Grad der Parallelität, der auf

- Die Arbeitsteilung, das Einreihen zusätzlicher Threads in die Warteschlange, Verzögerungen durch den aufgerufenen Endpunkt usw. verursachen einen Overhead.
- Eine hohe Zahl (z. B. 50) führt nicht unbedingt dazu, dass es schneller geht.



Einige andere reale Szenarien, in denen dieser Ansatz anwendbar ist:

- Versenden einzelner E-Mails an viele Empfänger
- Aktualisieren von Datensätzen in Dataverse, SharePoint-Listen, SQL
- Massenerstellung von Benutzern in Azure Active Directory
- Parallelisierte Genehmigungen erstellen ([mehr Details](#))
- Auffächern von Vorgängen in SharePoint ([mehr Details](#))

Im Folgenden werden die Auswirkungen der Gleichzeitigkeitssteuerung für die Array-Verarbeitung in jeder Schleife verglichen.

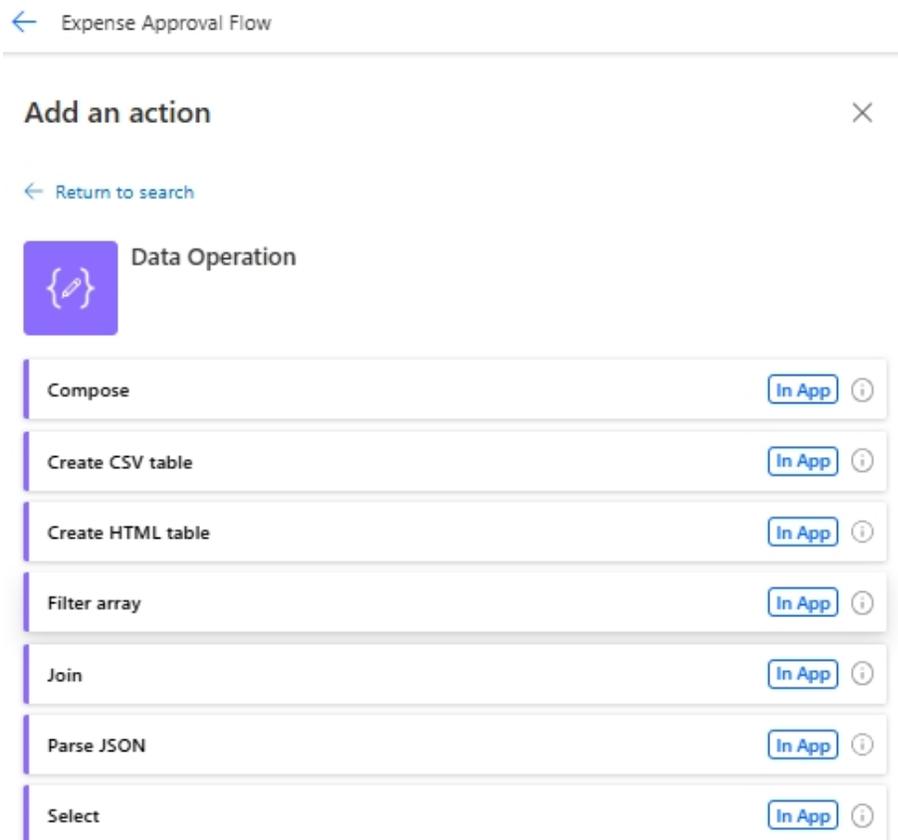
Array Länge	Grad der Parallelität	Zeit für die Ausführung der Schleife
4	Aus	21 Sekunden
4	2	11 Sekunden
4	4	6 Sekunden
4	6	6 Sekunden

Beachten Sie, dass die Gleichzeitigkeitskontrolle für "Apply to each"-Aktionen nur auf der höchsten Ebene im Cloud Flow wirksam wird. Bei der Verschachtelung von "Apply to Each"-Aktionen werden die inneren Aktionen immer seriell ausgeführt.

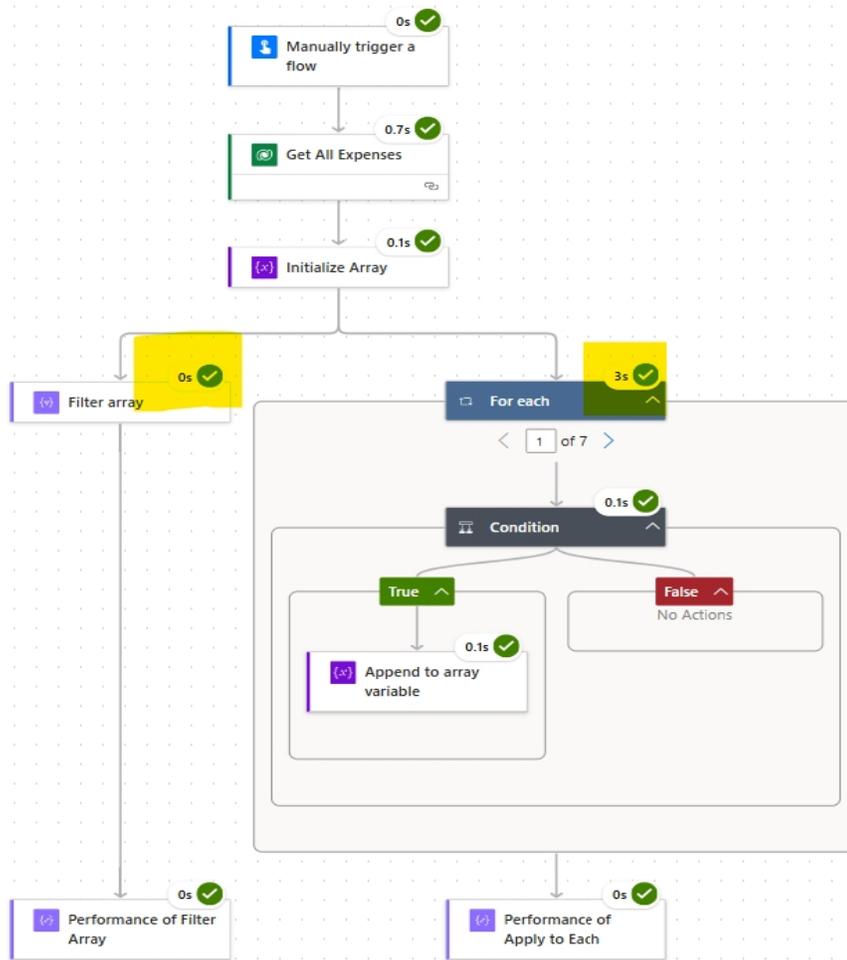
Arbeiten mit Datenoperationen

Datenoperationen in Power Automate beziehen sich auf die Manipulation, Transformation und Verwaltung von Daten innerhalb Ihrer Automatisierungsabläufe auf effiziente Weise.

Aktionen wie Filter Array, Select und Join können helfen, unnötige Schleifen bei der Datenverarbeitung zu vermeiden. Wenn es zum Beispiel notwendig ist, an Variablen anzuhängen, um ein endgültiges Array zu bilden, um eine Filterung vorzunehmen, wenn das Element nicht aus Informationen aus einer externen Datenquelle stammt, können sie oft durch Datenoperationen wie Filter oder Select vereinfacht werden.



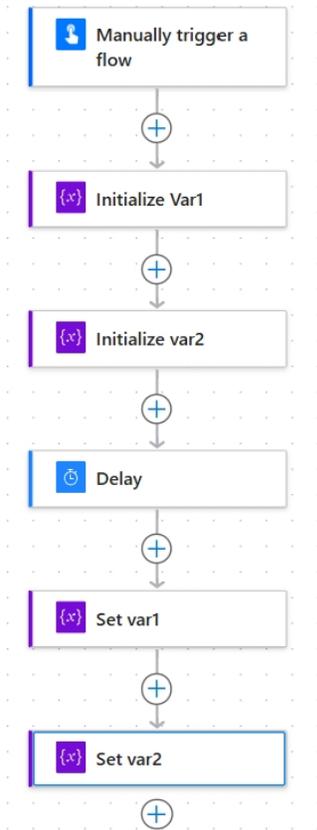
Hier sehen Sie, wie die Ausführung des Ablaufs bei der Verwendung von Append Variable innerhalb einer Schleife im Vergleich zu Filter Array aussieht. Im folgenden Beispiel zum Filtern eines Arrays mit 100 Datensätzen benötigte die Aktion "Filtern" nur wenige Millisekunden im Vergleich zu "Anhängen an Array" innerhalb einer Schleife, die 48 Sekunden dauerte und dem Ablauf mehr Aktionen und Komplexität hinzufügte.



Variablen im Vergleich zu Compose

Compose sind zur Laufzeit nicht aktualisierbar. Sie sind nützlich für die Verwendung von einmaligem Schreiben und vielem Lesen. Wenn wir an vielen Stellen im Ablauf schreiben oder aktualisieren müssen, sind Variablen viel besser geeignet. Ebenso kann die Verwendung von Compose zum Erstellen von Variablen schneller sein als das Initialisieren und Deklarieren von Variablen.

Wenn Sie mit einer Reihe von Variablen arbeiten, die innerhalb der gleichen logischen Blöcke im Ablauf aktualisiert werden, können Sie JSON-Variablen anstelle von einzelnen Variablen verwenden. Auf diese Weise lässt sich die Anzahl der Aktionen im Ablauf reduzieren.



Betrachten Sie das obige Beispiel: Anstatt mit Var1 und Var2 als separate Variablen zu arbeiten, können wir eine Objektvariable verwenden, um beide Werte zu verwalten.

← Variables/Compose Comparison

Initialize variable

Parameters Settings Code View About

Name *

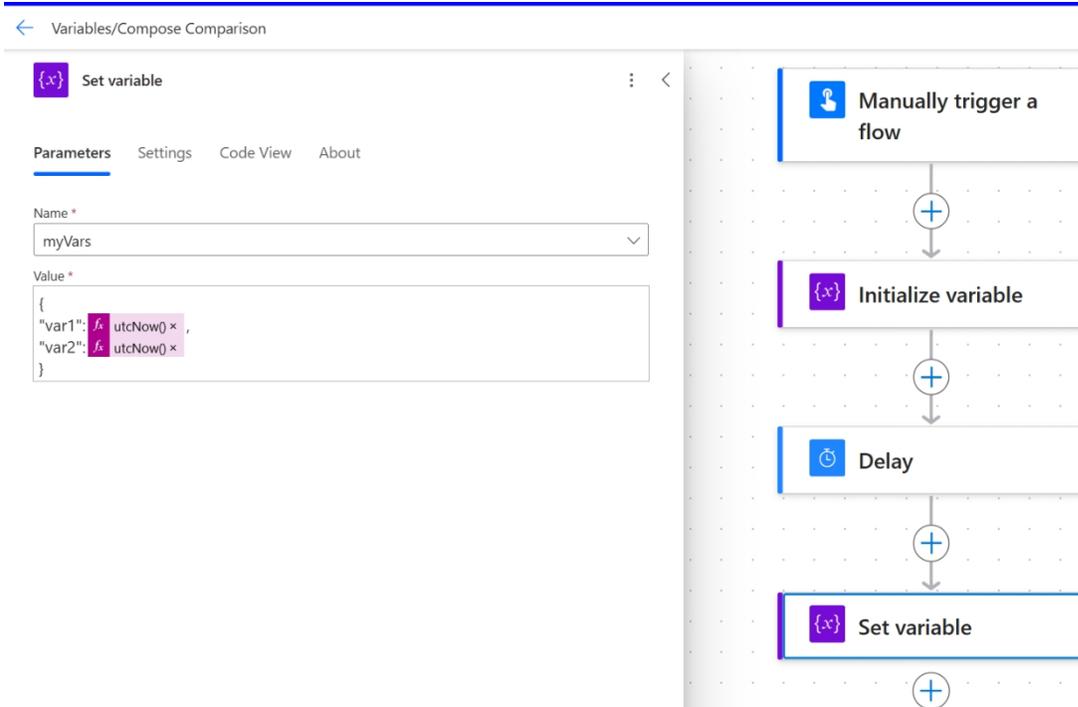
Type *

Value

```
{
  "var1": "one",
  "var2": "two"
}
```

```

    graph TD
      A[Manually trigger a flow] -- "+" --> B[Initialize variable]
      B -- "+" --> C[Delay]
      C -- "+" --> D[Set variable]
      D -- "+" --> E[ ]
  
```



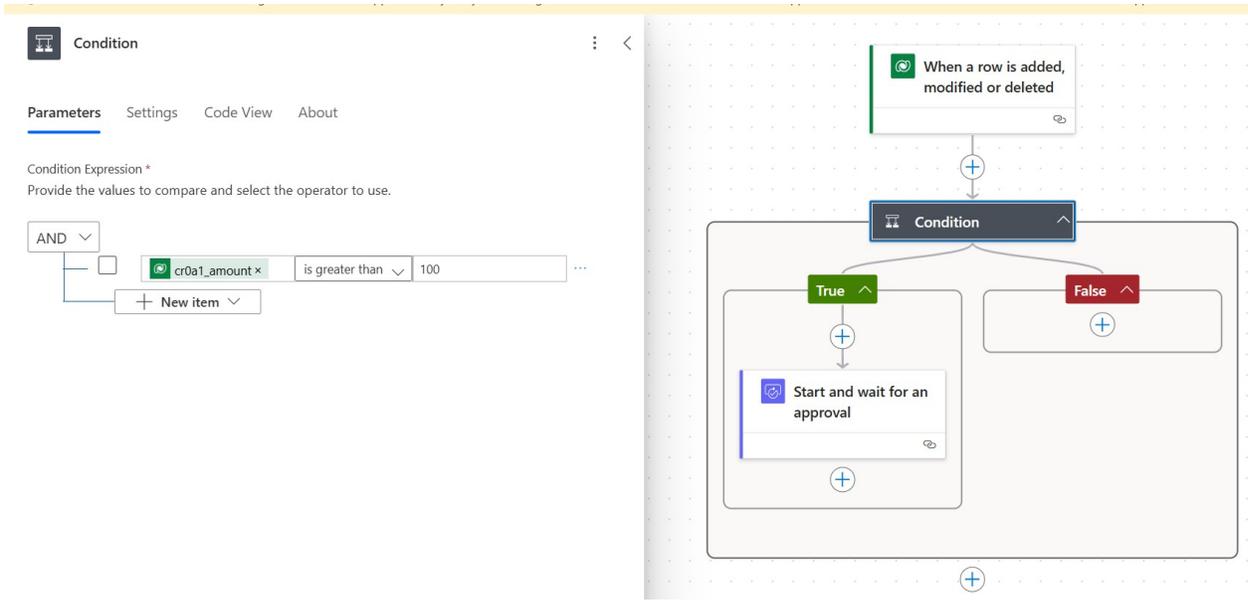
Leistung optimieren Auslöser automatisieren

Auslösebedingungen festlegen

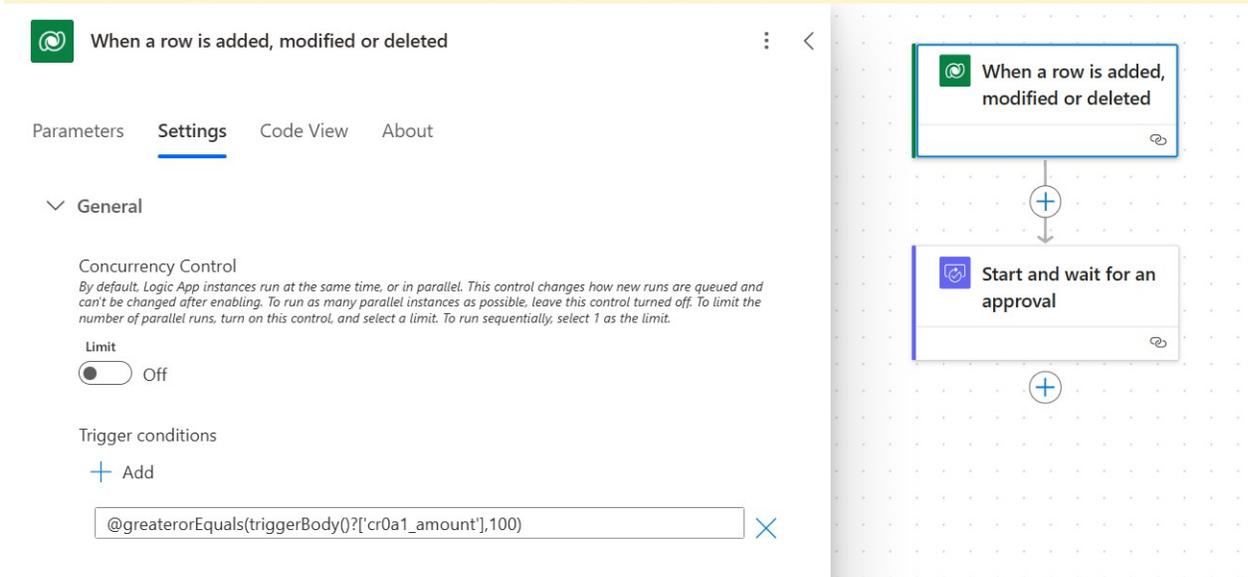
Ein häufiges Problem vieler Benutzer, die mit Power Automate arbeiten, ist, dass ihr Ablauf immer dann ausgeführt wird, wenn eine neue Zeile oder eine vorhandene Zeile in der Datenquelle geändert wird. Der Ablauf muss jedoch nur ausgeführt werden, wenn eine bestimmte Bedingung erfüllt ist. Wenn Sie die Auslöser richtig einstellen, kann Ihr Ablauf nur bei Bedarf ausgeführt werden.

Stellen Sie sich ein Szenario vor, in dem jeder Benutzer, der eine Ausgabe über \$5000 einreicht, von seinem Management genehmigt werden muss. Wenn die Auslösebedingungen nicht angegeben sind, wird der Fluss für jede eingereichte Ausgabe ausgeführt, und der Autor des Flusses muss zusätzliche Bedingungen angeben, um die Ausgaben herauszufiltern, die größer als \$5000. Mit Auslösebedingungen wird die Bewegung nur ausgelöst, wenn die Ausgaben über

\$5000 liegen. Hier ist die Bewegung für das obige Szenario ohne Auslösebedingungen:



Hier ist der Ablauf mit Auslösebedingungen:



Hinweis: Bei Dataverse-Triggern ermöglicht die OData-Filtereigenschaft auch, dass der Trigger nur ausgelöst wird, wenn die Filterbedingungen erfüllt sind. Sie können auch die Namen der Spalten angeben, bei denen der Fluss ausgelöst werden kann.

In einigen seltenen Szenarien, in denen Ihr Fluss mit Datenquellen mit sehr begrenztem Durchsatz arbeitet, können Sie die Konfiguration der Gleichzeitigkeitssteuerung des Triggers in Betracht ziehen. Standardmäßig führt der Cloud Flow-Trigger so viele Läufe wie möglich gleichzeitig aus, wenn seine Bedingungen erfüllt sind. Wir können dieses Verhalten ändern, indem wir die Gleichzeitigkeitskontrolle ändern, mit der Möglichkeit, die Anzahl der gleichzeitigen Läufe von mindestens 1 bis zu 100 zu begrenzen. Alle weiteren Läufe werden automatisch in eine Warteschlange gestellt.

Wir können beispielsweise eine Automatisierung durchführen, die von einer lokalen Ressource abhängt, die keine parallelen Ausführungen unterstützt, oder um Wettlaufbedingungen zu verhindern, bei denen ein Dirty-Read möglich ist, wenn es parallele Ausführungen gibt.

Beachten Sie, dass dies nicht mehr rückgängig gemacht werden kann. Um die Gleichzeitigkeitssteuerung zu entfernen, müssen Sie einen neuen Ablauf erstellen. Daher müssen Sie dies mit Vorsicht tun. Am besten belassen Sie es bei der Standardeinstellung, aber für den Fall, dass dies erforderlich ist, tun Sie es in einem Ablauf, der die geringste Anzahl von Aktionen hat - zum Beispiel können Sie Ihre Aktionen, die mit Datenquellen arbeiten, so organisieren, dass eine solche Steuerung in einem speziellen untergeordneten Ablauf erforderlich ist, so dass Sie diese Steuerung nur in dem untergeordneten Ablauf anwenden können.

Concurrency Control
 Limit number of concurrent runs of the flow, or leave it off to run as many as possible at the same time. Concurrency control changes the way new runs are queued. It cannot be undone once enabled.

Limit On

Degree of Parallelism 1

Start	Duration	Status
Jul 29, 04:16 PM (1 sec ago)	00:00:01	Running
Jul 29, 04:16 PM (1 sec ago)	00:00:01	Waiting
Jul 29, 04:16 PM (1 sec ago)	00:00:01	Waiting

Arbeiten mit relevanten Daten

In Power Automate ist es für die Effizienz und Genauigkeit Ihrer automatisierten Arbeitsabläufe entscheidend, dass Sie nur mit relevanten Daten arbeiten. In den [Limits und der Konfiguration](#) haben wir die Durchsatzgrenzen wie folgt festgelegt:

Name	Limit	Notes
Content throughput per 5 minutes	120 MB for Low; 1.2 GB for all others	You can distribute workload across more than one flow as necessary.
Content throughput per 24 hours	200 MB for Low; 2 GB for Medium; 10 GB for High	You can distribute workload across more than one flow as necessary.

Daher ist es wichtig, die Daten, die wir einbringen, auf die relevanten Daten zu beschränken. Dies kann sowohl auf der Ebene der Auslöser als auch auf der Ebene der Aktionen geschehen. In den meisten Fällen können Sie auf Trigger- und Aktionsebene Odata-Ausdrücke, Top Count und relevante Spaltennamen angeben.

Hier sind einige Beispiele dafür, wie dies geschehen kann:

Auslöser -

Datenquellen wie SharePoint ermöglichen die Angabe von Ansichten, um nur relevante Spalten und gefilterte Datensätze zu erhalten. In Triggern können Benutzer die Ansicht angeben, die für ihren Anwendungsfall relevant ist, und nur relevante Daten und ausgewählte Spalten abrufen.

The screenshot shows the configuration interface for the 'When an item is created or modified' trigger. The interface includes a 'Parameters' tab, a 'Site Address' dropdown, a 'List Name' dropdown, and an 'Advanced parameters' section with 'Showing 1 of 1' items. A 'Limit Columns By View' dropdown is set to 'Issues grouped by priority'. A plus sign icon is visible on the right side of the interface.

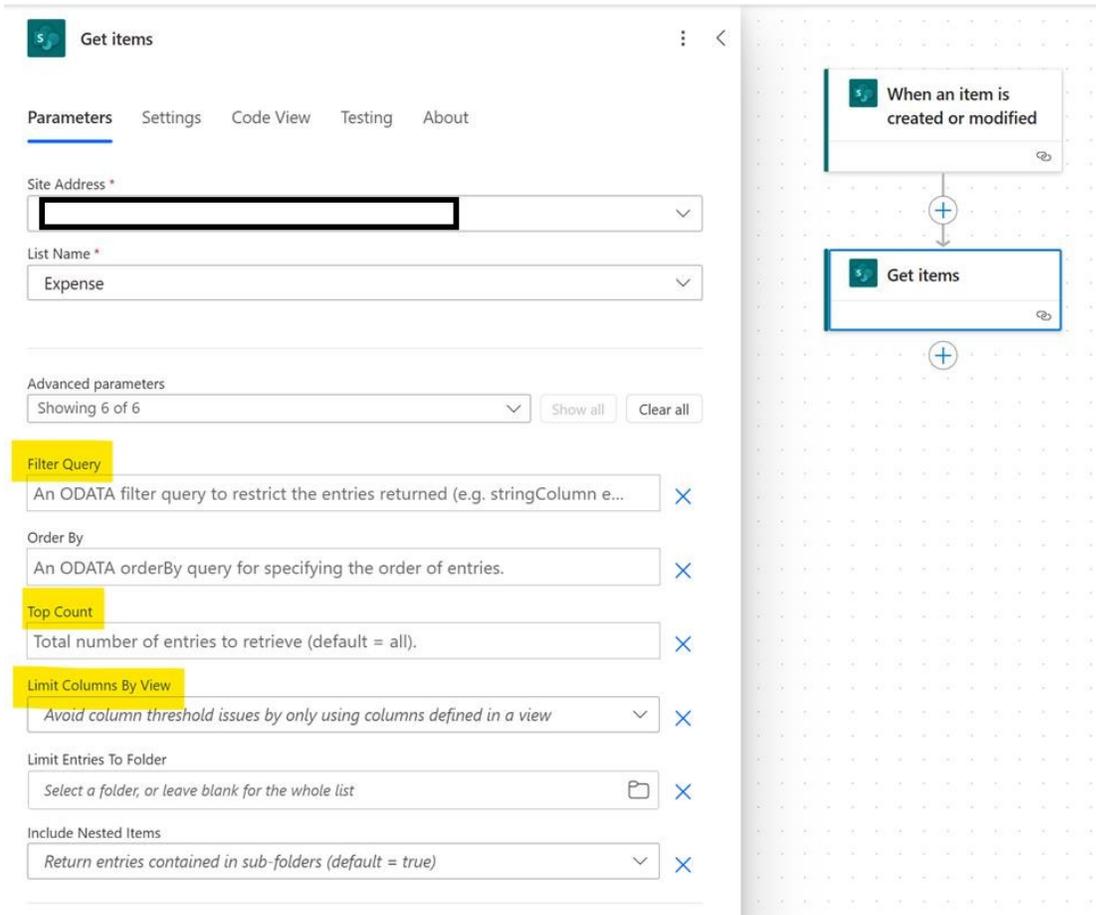
Aktionen-

Verwenden Sie erweiterte Optionen wie Select columns (nur Daten abrufen, die den in der Abfrage angegebenen Spaltennamen entsprechen), Filter rows (Daten basierend auf der Filterbedingung abrufen) und Row count (nur die angegebene Anzahl von Zeilen abrufen).

The screenshot shows the configuration interface for the 'List rows' action. The interface includes a 'Parameters' tab, a 'Table Name' field with the value 'cr0a1_travelexpenseses', an 'Advanced parameters' section with 'Showing 3 of 8' items, a 'Select Columns' field with the value 'cr0a1_date,cr0a1_description', a 'Filter Rows' field with the value 'cr0a1_amount ge 100', and a 'Row Count' field with the value '1'. A plus sign icon is visible on the right side of the interface.

Ergebnis ohne Select-Spalte	Ergebnis mit Select-Spalte
<pre> "value": [{ "@odata.type": "#Microsoft.Dynamics.CRM.cr0a1_travelexpenses", "@odata.id": "[REDACTED]", "@odata.etag": "W/\"4299492\"", "@odata.editLink": "cr0a1_travelexpenseses [REDACTED]", "cr0a1_category@odata.Community.Display.V1.FormattedValue": "Acc", "cr0a1_category": "704550001", "cr0a1_date@odata.Community.Display.V1.FormattedValue": "1/2/202", "cr0a1_date@odata.type": "#DateTimeOffset", "cr0a1_date": "2022-01-02T00:00:00Z", "_owningbusinessunit_value@odata.Community.Display.V1.FormattedV", "_owningbusinessunit_value@Microsoft.Dynamics.CRM.associatednavi", "_owningbusinessunit_value@Microsoft.Dynamics.CRM.lookuplogicaln", "_owningbusinessunit_value@odata.type": "#Guid", "_owningbusinessunit_value": "[REDACTED]", "statecode@odata.Community.Display.V1.FormattedValue": "Active", "statecode": 0, "statuscode@odata.Community.Display.V1.FormattedValue": "Active", "statuscode": 1, "_createdby_value@odata.Community.Display.V1.FormattedValue": "S", "_createdby_value@Microsoft.Dynamics.CRM.lookuplogicalname": "sy", "_createdby_value@odata.type": "#Guid", "_createdby_value": "[REDACTED]", "timezoneruleversionnumber@odata.Community.Display.V1.FormattedV", "timezoneruleversionnumber": 4, "_transactioncurrencyid_value@odata.Community.Display.V1.Format", "_transactioncurrencyid_value@Microsoft.Dynamics.CRM.associatedn", "_transactioncurrencyid_value@Microsoft.Dynamics.CRM.lookuplogi", "_transactioncurrencyid_value@odata.type": "#Guid", "_transactioncurrencyid_value": "3c534fdb-f27a-ee11-8179-000d3af", "cr0a1_expenseid": "002", "_ownerid_value@odata.Community.Display.V1.FormattedValue": "Sar", "_ownerid_value@Microsoft.Dynamics.CRM.associatednavigationprope", "_ownerid_value@Microsoft.Dynamics.CRM.lookuplogicalname": "syst", "_ownerid_value@odata.type": "#Guid", "_ownerid_value": "bd75f194-8614-ef11-9f89-002248af0c20", "cr0a1_amount@odata.Community.Display.V1.FormattedValue": "\$100.", "cr0a1_amount@odata.type": "#Decimal", "cr0a1_amount": 100, "_modifiedby_value@odata.Community.Display.V1.FormattedValue": "", "_modifiedby_value@Microsoft.Dynamics.CRM.lookuplogicalname": "s", "_modifiedby_value@odata.type": "#Guid", "_modifiedby_value": "bd75f194-8614-ef11-9f89-002248af0c20", "_owninguser_value@Microsoft.Dynamics.CRM.lookuplogicalname": "s", "_owninguser_value@odata.type": "#Guid", "_owninguser_value": "[REDACTED]", "createdon@odata.Community.Display.V1.FormattedValue": "7/31/202", "createdon@odata.type": "#DateTimeOffset", "createdon": "2024-07-31T21:19:56Z", "versionnumber@odata.Community.Display.V1.FormattedValue": "4,29", "versionnumber@odata.type": "#Int64", </pre>	<pre> "@odata.type": "#Microsoft.Dynamics.CRM.cr0a1_trave", "@odata.id": "[REDACTED]", "@odata.etag": "[REDACTED]", "@odata.editLink": "cr0a1_travelexpenseses(e0a60d9d", "cr0a1_date@odata.Community.Display.V1.FormattedVal", "cr0a1_date@odata.type": "#DateTimeOffset", "cr0a1_date": "2022-01-02T00:00:00Z", "cr0a1_description": "Hotel Stay", "cr0a1_travelexpensesid@odata.type": "#Guid", "cr0a1_travelexpensesid": "e0a60d9d-824f-ef11-accd- </pre>

In SharePoint verwenden Sie die folgenden Parameter, um mit relevanten Daten zu arbeiten.



Batch verwenden, wo möglich, maximale Seitengröße

Oft müssen Benutzer mit Power Automate 1000e von Datensätzen in der Datenquelle erstellen/aktualisieren, wenn ein Fluss ausgelöst wird. Die meisten Benutzer würden am Ende eine For Each-Schleife verwenden, die jeden dieser 1000 Datensätze durchläuft und sie nacheinander in die Datenquelle schiebt, was zu Latenz und Verzögerungen führt.

Um dieses Problem zu lösen, können Sie die folgenden 2 Ansätze ausprobieren:

- Erstellen/Pflegen von Datensätzen in der Datenquelle in BATCH. Die meisten Konnektoren/Dienste stellen API-Dienste zur Verfügung, um Anfragen in BATCH zu stellen. Mit einem Batch-Vorgang können Sie mehrere Vorgänge in einer einzigen HTTP-Anfrage zusammenfassen. Diese Vorgänge werden nacheinander in der angegebenen Reihenfolge ausgeführt. Die Reihenfolge der Antworten entspricht der Reihenfolge der Anfragen im Batch-Vorgang.
- Die Parallelität in der For Each-Schleife ermöglicht die Verarbeitung von bis zu 50 Datensätzen für die Dienste, die keine BATCH-Anfragen annehmen können.

Die folgenden Links zu Rest-APIs von [Sharepoint](#) und [Dataverse](#) zeigen, wie man Batch-Anfragen stellt.



Send an HTTP request to SharePoint - Update default list view batch action

Parameters Settings Code View Testing About

Site Address *

Contoso [redacted] ▾

Method *

POST ▾

Uri *

_api/\$batch

Advanced parameters

Showing 2 of 2 ▾ Show all Clear all

Headers

Content-Type	multipart/mixed;boundary= [redacted] BatchGUID ×	× 🗑️ ×
Expect	100-continue	

Body

```
-- [redacted] BatchGUID ×  
Content-Type: multipart/mixed;boundary=" [redacted] ChangeSetGUID × "  
Host: https:// [redacted]  
Content-Transfer-Encoding: binary  
  
[redacted] Outputs ×  
-- [redacted] ChangeSetGUID × --  
-- [redacted] BatchGUID × --
```

Bei der Arbeit mit Dataverse können Sie die Vorteile der neuen [Bulk Operations Web APIs](#) nutzen. Die Bulk Operations API unterscheidet sich von Batch Operations - während Batch Operations in einer einzigen Anfrage gebucht werden, werden sie als mehrere Operationen ausgeführt. Massenvorgänge werden in einer einzigen Anfrage gebucht und als ein einziger Vorgang gezählt. Die Web-API für Massenoperationen kann über HTTP mit der Entra-ID oder über den HTTP-Connector bei Verwendung von Service Principals aufgerufen werden. Zusätzlich können Sie mit Dataverse Bulk Operations die Anzahl der Aktionen reduzieren.

Bulk Create

Parameters Settings Code View About

From *
Outputs

Map *

demo_name	Name	X
demo_quantity	Quantity	X
demo_reviewstatus	If(,)	X
@odata.type	Microsoft.Dynamics.CRM.demo_sampleitem	X
Enter key	Enter value	

↓

Invoke an HTTP request

Parameters Settings Code View Testing About

Method *
POST

Url Of The Request *
https://[redacted]api/data/v9.2/demo_sampleitems/Microsoft.Dynamics.CRM.CreateMultiple

Advanced parameters
Showing 2 of 2 Show all Clear all

Headers

Content-Type	application/json	X
Accept	application/json	X
If-None-Match	null	X
OData-MaxVersion	4.0	X
OData-Version	4.0	X
MSCRM.ByPassCustomPluginExecution	true	X
MSCRM.SuppressCallbackRegistrationExpanderJob	true	X
Enter key	Enter value	

Body Of The Request

```
{
  "Targets": [Outputs]
}
```

Im obigen Beispiel bereiten wir die Datensätze im JSON-Format mit der Select-Aktion vor und verwenden HTTP mit der Entra-ID, um die Anfrage mit der CreateMultiple-Web-API zu posten. Wenn wir 100 Datensätze in der JSON-Ausgabe haben, erfordert dies nur eine einzige Aktion anstelle von 100 Create Row-Aktionen in Dataverse.

Verschachtelte For-Each-Schleifen vermeiden

Verschachtelte "For Each"-Schleifen können in Power Automate (und in vielen anderen Programmierkontexten) aufgrund ihrer potenziellen Auswirkungen auf die Leistung und den Ressourcenverbrauch eine teure Angelegenheit sein.

- **Erhöhte Ausführungszeit:** Bei verschachtelten Schleifen vervielfacht sich die Anzahl der Iterationen. Wenn Sie zum Beispiel 2 Schleifen mit jeweils 10 Iterationen haben, beträgt die Gesamtzahl der Iterationen $10 \times 10 = 100$. Dies kann die Ausführungszeit Ihres Ablaufs erheblich verlängern, insbesondere wenn die Schleifen große Datensätze verarbeiten oder komplexe Operationen innerhalb jeder Iteration durchführen.
- **Begrenzungen und Quoten:** Power Automate legt Grenzen und Quoten für verschiedene

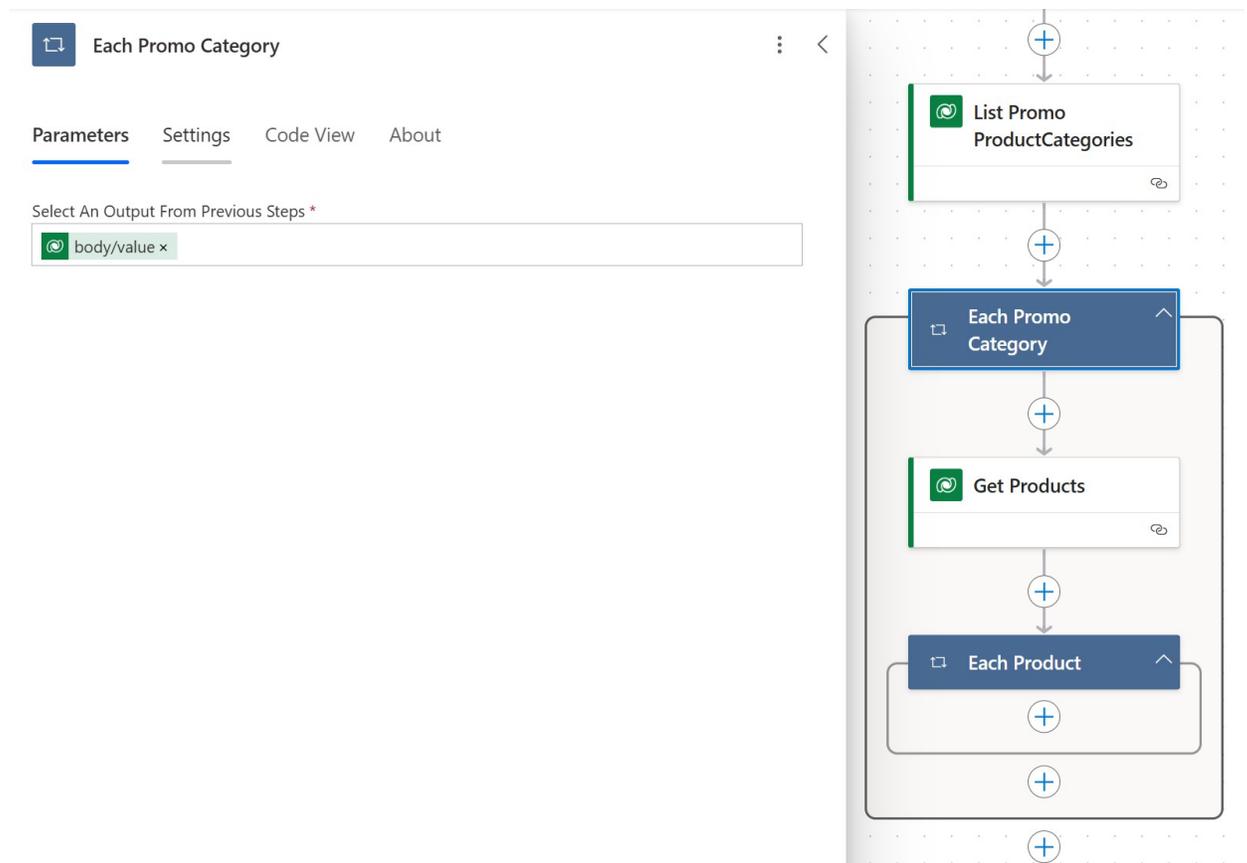
Aspekte der Ablaufausführung fest, z. B. die Anzahl der zulässigen Iterationen innerhalb einer Schleife oder die maximale Ausführungsdauer.

Zeit. Bei verschachtelten Schleifen können diese Grenzen schnell erreicht werden, so dass der Datenfluss ausfällt oder gedrosselt wird, insbesondere wenn er mit großen Datensätzen arbeitet oder häufig ausgeführt wird.

Weitere Informationen zu den Grenzwerten finden [Sie hier](#).

Je nach Ihrem Szenario können Sie verschachtelte Schleifen mit den folgenden Alternativen vermeiden.

-Wenn verschachtelte Schleifen zur Verarbeitung von Bezugsdatensätzen aus einer übergeordneten Tabelle verwendet werden: Eine äußere Schleife kann z. B. List Rows in Dataverse für die Tabelle ProductCategory verwenden, um eine Liste von Produktkategorien abzurufen, bei denen die Spalte IsPromotion = true ist, und eine innere Schleife wird verwendet, um die Bezugsdatensätze in der Tabelle Product zu durchlaufen, die zu einer bestimmten Kategorie aus der ersten Schleife gehören.



Eine effizientere Implementierung ist die Verwendung der OData Query Expansion, bei der wir nur mit einer einzigen Apply-for-Each-Schleife arbeiten müssen. Dadurch werden auch die gesamten Anfragen an Dataverse auf nur einen RetrieveMultiple-Aufruf reduziert. Verwenden Sie den Parameter Expand Query, um den Namen der Suchspalte anzugeben. Wir können auch (\$select=...) hinzufügen, um die Anzahl der aus der Bezugstabelle zurückgegebenen Spalten zu reduzieren. Der Parameter filter rows kann verwendet werden, um Bedingungen auf die Spalte der Nachschlagetabelle anzuwenden.

The image shows the configuration interface for a Power Automate flow named "List Promo ProductCategories". The configuration includes:

- Table Name:** Products
- Advanced parameters:** Showing 8 of 8
- Select Columns:** demo_name
- Filter Rows:** demo_Category/demo_ispromotion eq true
- Sort By:** Columns to sort by in OData orderBy style (excluding lookups)
- Expand Query:** demo_Category(\$select=demo_name)
- Fetch Xml Query:** Enter a Fetch XML query for advanced customization
- Row Count:** Enter the number of rows to be listed (default = 5000)
- Skip Token:** Enter the skip token obtained from a previous run to list rows from the ne...
- Partition ID:** An option to specify the partitionId while retrieving data for NoSQL tables

To the right, a flow diagram illustrates the process:

- Start with a connector (+).
- Execute the "List Promo ProductCategories" action.
- Enter an "Each Promo Category" loop.
- Inside the loop, execute the "Get Products" action.
- Enter an "Each Product" loop.
- End with a connector (+).

Unendliche Läufe vermeiden

Mit Power Automate kann man sehr leicht in eine Falle laufen, in der die Abläufe unendlich oft ausgelöst werden, insbesondere dann, wenn die gleiche Tabelle aktualisiert werden muss, auf der der Ablauf ausgelöst werden soll.

Beachten Sie, dass der Maker eine Warnung erhält, wenn er einen Ablauf speichert, der zu einer unendlichen Triggerschleife führen kann.

▼ Update a row (1)

Actions in this flow may result in an infinite trigger loop. Please ensure you add appropriate conditional checks to prevent this flow from triggering itself.

Um das Problem der unendlichen Läufe zu lösen, können Sie die angegebenen Schritte befolgen:

- **Verwenden Sie Auslösebedingungen, um Endlosschleifen im Ablauf zu vermeiden** - Integrieren Sie Auslösebedingungen in Ihren Ablauf, um zu prüfen, ob bestimmte Kriterien erfüllt sind, bevor Sie mit den nachfolgenden Aktionen fortfahren. Dadurch wird sichergestellt, dass der Ablauf nur dann ausgeführt wird, wenn es notwendig ist, und die Wahrscheinlichkeit, dass eine Endlosschleife ausgelöst wird, verringert. Alternativ können Sie den Ablauf beenden, wenn die Bedingung, die die Schleife auslöst, erfüllt ist.

Beispiel für einen Fluss, der unendlich lang sein könnte

The image displays two screenshots from the Salesforce Flow Builder interface. The top screenshot shows the configuration for a trigger: "When a row is added, modified or deleted". The parameters are set to "Change Type" (Added or Modified), "Table Name" (Accounts), and "Scope" (Organization). Below this, there are advanced parameters and a "Show all" button. The bottom screenshot shows the configuration for an action: "Update a row". The parameters include "Table Name" (Accounts), "Row ID", and various fields for account details such as "Account Name", "Address 1: City", "Address 1: Street 1", "Address 1: Street 2", "Address 1: ZIP/Postal Code", "Annual Revenue", "Description", "Updated from Flow", "Main Phone", "Number Of Employees", and "Primary Contact (Contacts)". A flow diagram on the right side of the image shows the trigger connected to the update action, with a plus sign indicating a loop.

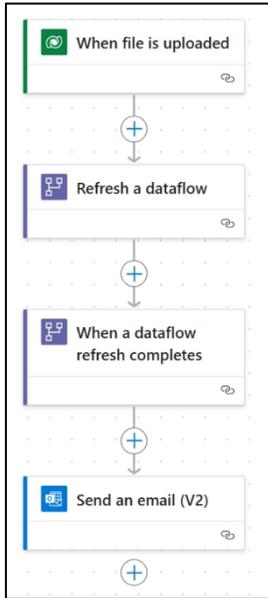
Beispiel dafür, wie Sie den unendlichen Fluss stoppen können

The image shows the configuration for the 'Update a row' action in Power Automate. On the left, the 'Parameters' tab is active, showing fields for 'Table Name' (Accounts), 'Row ID', and various address and revenue fields. The 'Description' field contains 'Updated from Flow'. On the right, the flow logic is visualized: a trigger 'When a row is added, modified or deleted' leads to a 'Condition' step. The 'Condition' has two paths: 'True' leads to a 'Terminate' action, and 'False' leads to the 'Update a row' action.

Vermeiden Sie die Durchführung einer großen Anzahl von Datenumwandlungen

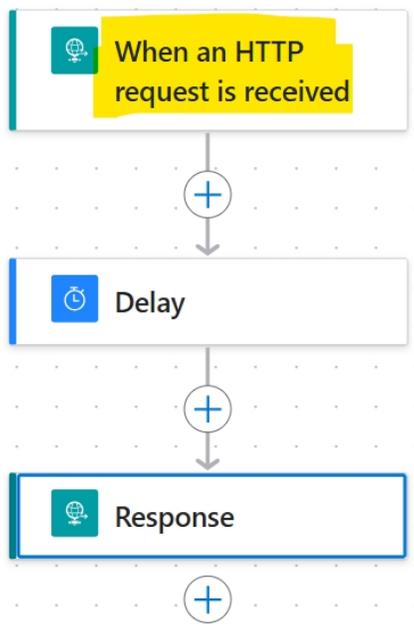
Wenn Sie mit großen Datenmengen arbeiten müssen, sollten Sie überlegen, ob es sich wirklich um ein Extract-Transform-Load-Szenario (ETL) handeln sollte. Beispielsweise kann ein Power Automate Cloud Flow verwendet werden, um Daten aus einer großen Excel-Tabelle zu lesen, einige Datenformatierungen oder -validierungen vorzunehmen und sie dann in Dataverse zu schreiben. In einem solchen Szenario ist es sinnvoller, Power Platform Dataflow oder ein anderes gleichwertiges ETL-Tool anstelle von Cloud Flows zu verwenden.

Wenn Sie das Laden von Daten mit einer Art Orchestrierungslogik verwalten müssen, die Sie in Cloud Flows implementieren können, sollten Sie eine Kombination mit Dataflow in Betracht ziehen. Wir können den Dataflow-Konnektor verwenden, um eine Aktualisierungsaktion aufzurufen, und seinen Auslöser "Wenn eine Dataflow-Aktualisierung abgeschlossen ist" nutzen, um Post-ETL-Aktionen durchzuführen.



Asynchrones Ablaufmuster

Verwenden Sie asynchrone Ablaufmuster für lang laufende Abläufe. Wenn ein Ablauf von einem übergeordneten Ablauf oder einer Power App aufgerufen wird, muss er im Allgemeinen innerhalb von 120 Sekunden eine Antwort an die aufrufende Entität zurücksenden. Wenn der aufgerufene Ablauf dies nicht schafft, erhält die aufrufende Entität eine Zeitüberschreitung und bricht den Vorgang ab. Daher kann der Datenfluss in solchen Fällen mit der Einstellung Asynchrone Antwort mit einer 202 antworten, um anzuzeigen, dass die Anfrage zur Bearbeitung angenommen wurde. Außerdem wird ein Standort-Header bereitgestellt, um den endgültigen Status abzurufen.



 Response ⋮ <

Parameters **Settings** Code View About

∨ Networking

Asynchronous Response On

Content Transfer
Specify the behavior and capabilities for transferring content over HTTP.

Chunk size

∨ Run After

+ Select Actions ∨

>  Delay ●... ● 

∨ Security

Secure Inputs
Enabling secure inputs will automatically secure outputs.

Off

Umgebungsvariablen

Umgebungsvariablen in den Cloud-Flows von Power Automate sind eine leistungsstarke Funktion, die die Flexibilität, Wartbarkeit und Verwaltbarkeit Ihrer Workflows verbessert, insbesondere wenn Sie mit mehreren Umgebungen wie Entwicklung, Test und Produktion arbeiten. Hier erhalten Sie einen detaillierten Einblick in die Verwendung und Verwaltung von Umgebungsvariablen in Power Automate:

Warum Umgebungsvariablen verwenden?

Umgebungsvariablen sind Parameter, die zum Speichern von Konfigurationseinstellungen und anderen Werten verwendet werden können, die leicht geändert werden können, ohne die eigentliche Ablauflogik zu verändern. Auf diese Variablen kann innerhalb von Abläufen verwiesen werden, was die Verwaltung umgebungsspezifischer Einstellungen wie URLs, API-Schlüssel, Verbindungszeichenfolgen oder anderer Konfigurationsdaten erleichtert.

1. **Trennung von Konfiguration und Logik:** Durch die Verwendung von Umgebungsvariablen trennen Sie die Konfigurationsdaten von der Ablauflogik. Dadurch werden die Abläufe modularer und leichter zu verwalten.
2. **Leichtere Bereitstellung:** Umgebungsvariablen vereinfachen den Bereitstellungsprozess in verschiedenen Umgebungen (Entwicklung, Test, Produktion), da Sie Konfigurationen ändern können, ohne die Abläufe zu bearbeiten.
3. **Wartbarkeit:** Sie erleichtern die Aktualisierung und Wartung von Abläufen. Wenn ein Konfigurationswert geändert werden muss, können Sie die Umgebungsvariable aktualisieren, ohne jede einzelne Bewegung bearbeiten zu müssen.

Beachten Sie bei der Verwendung von Umgebungsvariablen die folgenden bewährten Verfahren

1. **Konsistente Benennungskonventionen:** Verwenden Sie klare und konsistente Namenskonventionen für Ihre Umgebungsvariablen, um sicherzustellen, dass sie leicht identifizierbar und verständlich sind.
2. **Verwenden Sie Standardwerte mit Bedacht:** Legen Sie aussagekräftige Standardwerte fest, die als Ausweichmöglichkeiten verwendet werden können, falls der aktuelle Wert nicht eingestellt ist.
3. **Variablen dokumentieren:** Dokumentieren Sie den Zweck und die Verwendung jeder Umgebungsvariablen, um sicherzustellen, dass die Teammitglieder verstehen, wozu jede Variable dient und wie sie zu konfigurieren ist.
4. **Sensible Daten begrenzen:** Vermeiden Sie es, hochsensible Daten direkt in Umgebungsvariablen zu speichern. Ziehen Sie die Verwendung sicherer Mechanismen für sensible Konfigurationen in Betracht (z. B. Azure Key Vault).

Beispiel-Szenario

Stellen Sie sich vor, Sie haben einen Fluss, der mit einem API-Endpunkt verbunden ist, der sich in Entwicklungs-, Test- und Produktionsumgebungen unterscheidet:

1. **Erstellen Sie Umgebungsvariablen:**
 - API_Base_URL mit dem Standardwert `https://api.dev.example.com` für die Entwicklung.
 - Ändern Sie den aktuellen Wert in der Testumgebung auf `https://api.test.example.com`.
 - Ändern Sie den aktuellen Wert in der Produktionsumgebung in `https://api.example.com`.
2. **Referenz im Fluss:**
 - Setzen Sie die URL in der HTTP-Aktion in Ihrem Ablauf auf `@{variables('API_Base_URL')}`.

Durch die Verwendung von Umgebungsvariablen müssen Sie beim Übergang Ihrer Lösung von der Entwicklung zum Testen und zur Produktion nur die Werte der Umgebungsvariablen aktualisieren, ohne die Ablauflogik zu ändern. Dies gewährleistet Konsistenz, reduziert Fehler und vereinfacht die Verwaltung.

Fehlerbehandlung

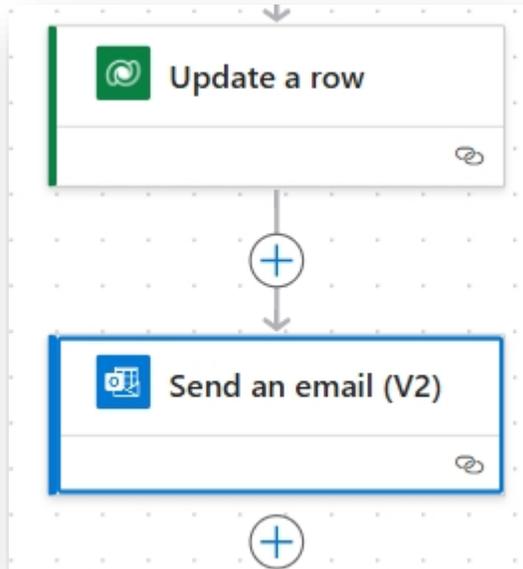
Die Fehlerbehandlung in Power Automate ist entscheidend für die Zuverlässigkeit und Robustheit Ihrer automatisierten Workflows. Im Folgenden finden Sie einige wichtige Strategien für die

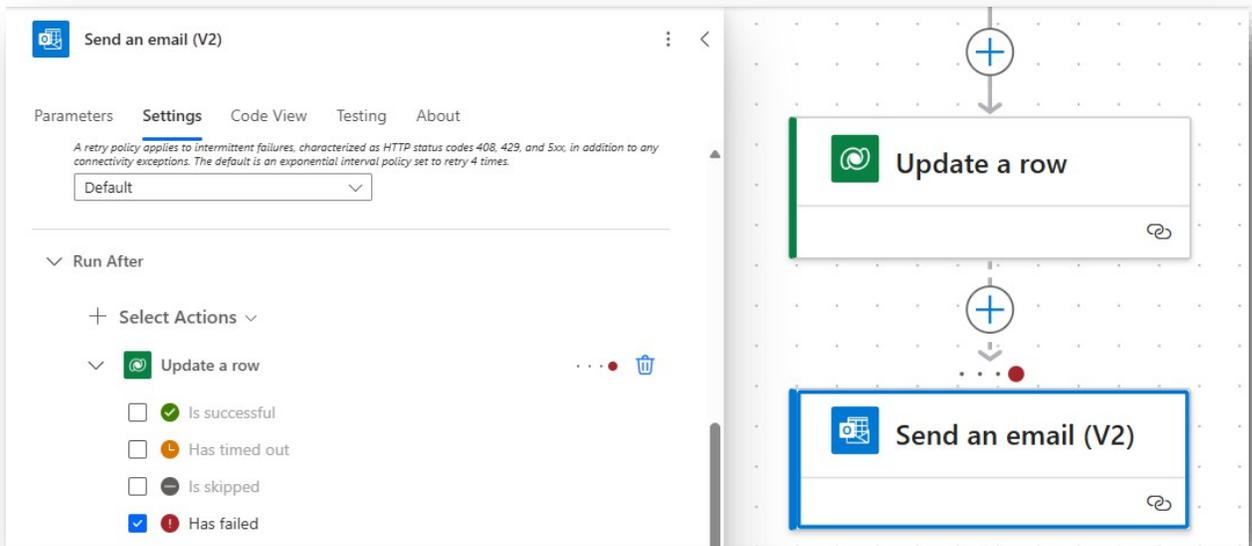
Implementierung der Fehlerbehandlung in Power Automate:

Konfigurieren der "Ausführen nach"-Einstellungen

Für jede Aktion in Ihrem Ablauf können Sie die "Ausführen nach"-Einstellungen konfigurieren, um festzulegen, ob die Aktion auf der Grundlage des Ergebnisses der vorherigen Aktionen ausgeführt werden soll. Dies ermöglicht eine bedingte Ausführung und Fehlerbehandlung. Sie können zum Beispiel eine Aktion so konfigurieren, dass sie nur ausgeführt wird, wenn die vorherige Aktion fehlgeschlagen ist, übersprungen wurde oder eine Zeitüberschreitung auftrat.

In diesem Beispiel erhält der Benutzer eine Benachrichtigung, wenn der Ablauf beim Schritt "Aktualisieren einer Zeile" fehlschlägt.





Sie können auch die Funktion **workflow()** in Power Automate verwenden, um detaillierte Informationen über den Ablauf zu erhalten, einschließlich der aktuellen Umgebungsguid, des Ablaufnamens, der Ablaufguid und der Ablauf-ID.

Workflow() json schema

```
{
  "Typ": "Objekt",
  "Eigenschaften": {
    "id": {
      "Typ": "string"
    },
    "Name": {
      "Typ": "string"
    },
    "Typ": {
      "Typ": "string"
    },
    "Standort": {
      "Typ": "string"
    },
    "Tags": {
      "Typ": "Objekt",
      "Eigenschaften": {
        "flowDisplayName": {
          "type": "string"
        }
      },
      "Umgebungsname":
    }
  }
}
```

```
{"type": "string"
```

```

    },
    "logicAppName":
      {"type":
        "string"
      },
    "environmentFlowSuspensionReason":
      {"type": "string"
      },
    "Zustand": {
      "Typ": "string"
    },
    "createdTime":
      {"type":
        "string"
      },
    "lastModifiedTime":
      {"type": "string"
      },
    "createdBy": {
      "Typ": "string"
    },
    "triggerType":
      {"type":
        "string"
      }
  }
}
},
"ausführen": {
  "Typ": "Objekt",
  "Eigenschaften": {
    "id": {
      "Typ": "string"
    },
    "Name": {
      "Typ": "string"
    },
    "Typ": {
      "Typ": "string"
    }
  }
}
}
}}

```

Parse JSON

Parameters Settings Code View About

Content *

workflow()

Schema *

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "name": {
      "type": "string"
    },
    "type": {
      "type": "string"
    },
    "location": {
      "type": "string"
    },
    "tags": {
      "type": "object",
      "properties": {
        "flowDisplayName": {
          "type": "string"
        }
      }
    }
  }
}
```

Create Failed Flow Run URL

Parameters Settings Code View About

Inputs *

https://make.powerautomate.com/environments/ Body environmen... x /flows
 Body logicAppNa... x /runs/ Body run x

Send Failure Alert

Parameters Settings Code View Testing About

To * Advanced mode
 Enter part of a name or email address to find people

Subject *
 [Critical Alert] - Your mission critical flow- Body flowDisplay... x has failed

Body *
 Normal Arial 15px B I U A
 Check out the failed flow run here
 Outputs x

Sie können diese Informationen dann auswerten und die URL für den Ablauf erstellen, die Sie als Hyperlink in die Benachrichtigungs-E-Mail einfügen können.

Warnung !! - Verwenden Sie diese Option mit Bedacht, da sie zu einer umfangreichen benutzerdefinierten Protokollierung führen und zu viele Aktionen auslösen kann, was sich auf die Gesamtleistung auswirkt. In gewisser Weise kann es sich zu einem großen Anti-Pattern entwickeln, wenn zu oft gewarnt wird.

Implementierung des Try-Catch-Musters mit Hilfe von Scopes

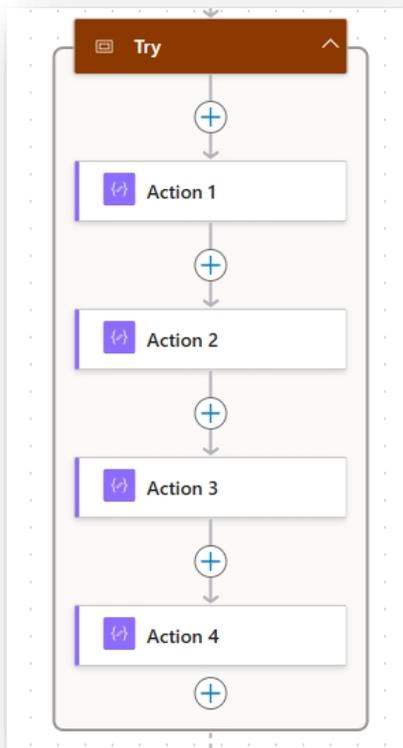
Die Ergebnisse von Try geben Ihnen alle Details der Aktion innerhalb des Try-Blocks. Im folgenden Beispiel haben wir 4 Compose Actions innerhalb des Try-Blocks wie folgt konfiguriert

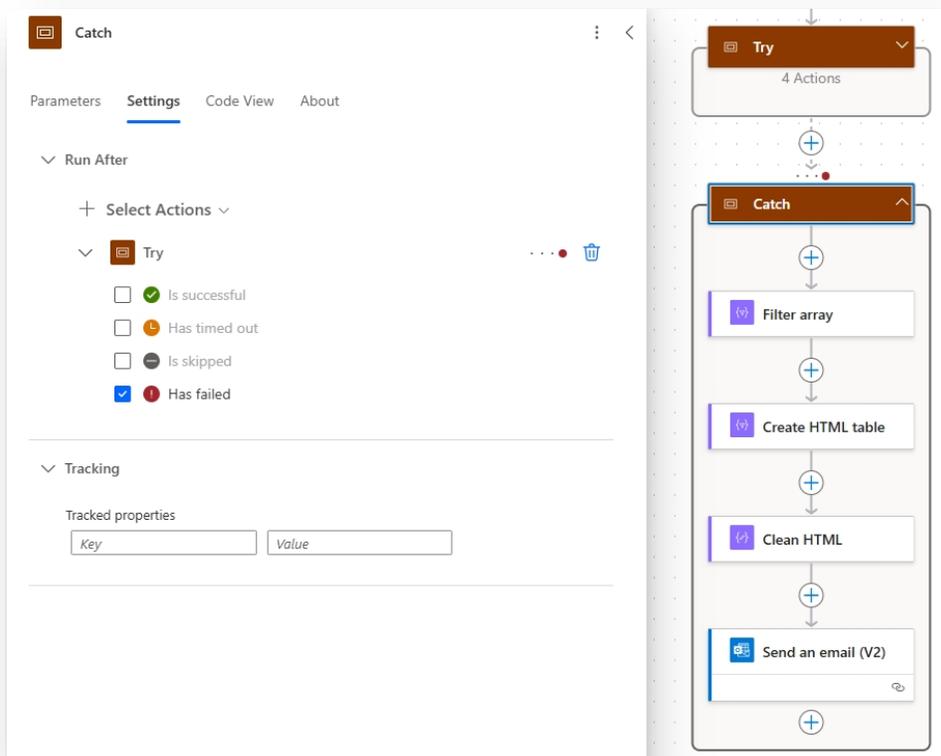
Aktion 1 - test (string)

Aktion 2 - concat('af','g')

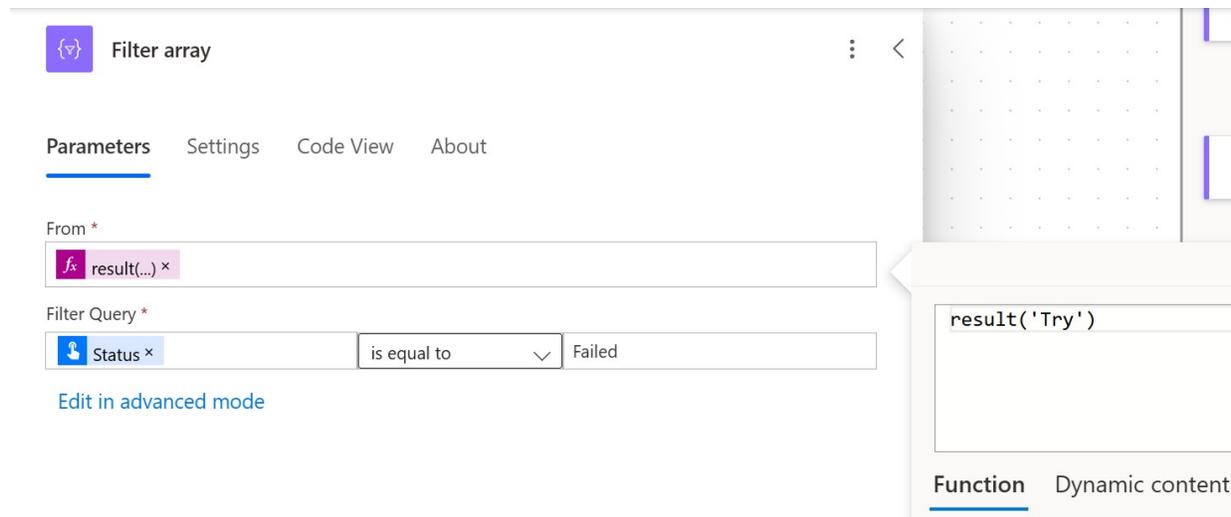
Aktion 3 - div(1,0) Aktion

4 - test (string)





Verwenden Sie das Filter-Array, um die Ergebnisfunktion zu filtern und die fehlgeschlagenen Fehler zu erhalten. Die Benutzer können die Datensätze weiter bereinigen und eine HTML-Tabelle erstellen, um E-Mail-Warnungen zu senden, wie unten gezeigt





High importance

Check flow Run

Status	Code	Error Code	Error Message	Action
Failed	BadRequest	InvalidTemplate	Unable to process template language expressions in action "Action_3" inputs at line "0" and column "0": "Attempt to divide an integral or decimal value by zero in function "div".	Action_3

Wiederholungsrichtlinie verwenden

Die Wiederholungsrichtlinie in den Datenflusseinstellungen hilft bei der Behandlung vorübergehender Standardfehler. Es wird empfohlen, sich für exponentielle Richtlinien zu entscheiden, da sie über einen längeren Zeitraum gelten können.

List rows

Parameters **Settings** Code View Testing About

General

Action Timeout
Specify the duration in ISO 8601 format
Example: P1D

Networking

Content Transfer
Specify the behavior and capabilities for transferring content over HTTP

Chunk size
Specify download chunk size between and Mb. Example: 10

Pagination
Retrieve items to meet the specified threshold by following the continuation token. Due to connector's page size, the number returned may exceed the threshold.
 Off

Retry Policy
A retry policy applies to intermittent failures, characterized as HTTP status codes 408, 429, and 5xx, in addition to any connectivity exceptions. The default is an exponential interval policy set to retry 4 times.

Default

None

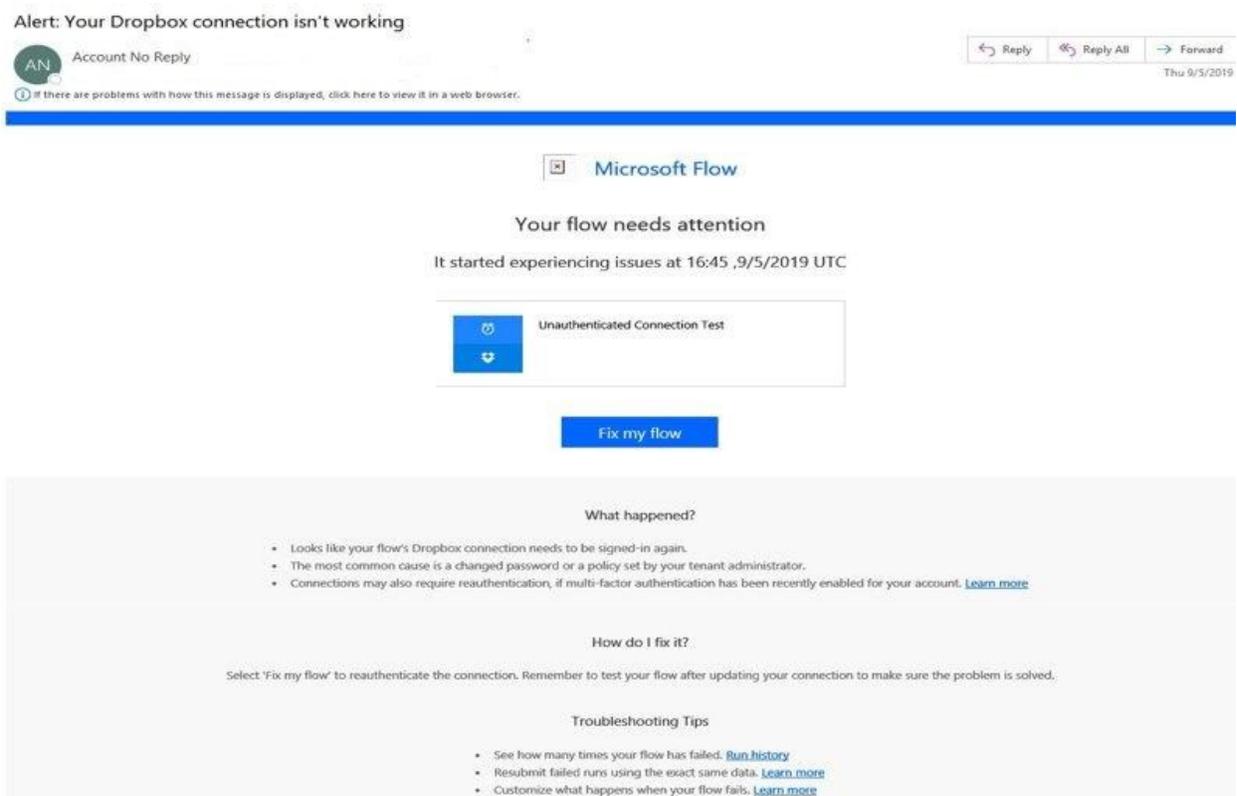
Exponential Interval

Fixed Interval

Secure Outputs
 Off

E-Mails zur Flussbereinigung verwenden

Der Power Automate Flow-Dienst generiert E-Mail-Warnungen für alle Flow-Besitzer bei bestimmten häufigen oder kritischen Fehlern wie unterbrochenen Verbindungen oder Abschaltung des Flows aufgrund von Drosselung. Diese E-Mails enthalten detaillierte Fehlerinformationen sowie Tipps zur Behebung des Problems.



Fluss Abschaltverhalten

Ein Auslöser ist ein Ereignis, das einen Cloud-Flow startet. Wenn Sie z. B. eine Benachrichtigung in Microsoft Teams erhalten möchten, wenn Ihnen jemand eine E-Mail sendet, ist in diesem Fall der Erhalt einer E-Mail der Auslöser, der diesen Fluss startet.

Wie funktioniert der Auslöser?

Es gibt 2 Arten von Auslösern:

- Auslöser für den Abruf
- Webhook-Auslöser

Sobald der Fluss erstellt ist, registriert sich der Trigger selbst, um entweder den Dienst abzufragen, mit dem er sich zu verbinden versucht, oder den Dienst abzuhören. Mit Dienst meinen wir, dass ein SQL-Trigger den SQL-Server abfragt oder abhört, mit dem er sich zu verbinden versucht. Wenn es sich um

einen Outlook-Trigger handelt, wird er den Outlook-Dienst abhören.

Auslöser-Typ	Beschreibung
Umfragen, wie zum Beispiel der Auslöser für das Wiederauftreten	Wenn der Fluss wieder eingeschaltet wird, werden alle nicht verarbeiteten oder ausstehenden Ereignisse verarbeitet. Wenn Sie ausstehende Einträge nicht verarbeiten möchten, wenn Sie Ihre Fluss wieder ein, löschen Sie ihn und erstellen Sie ihn neu.
Webhook	Wenn der Fluss wieder eingeschaltet wird, verarbeitet er neue Ereignisse, die erzeugt werden nachdem der Durchfluss eingeschaltet wurde.

Auslöser für den Abruf

Sobald der Abfrageauslöser registriert ist, fragt er den Dienst alle X Minuten ab, um die Details über alle Datensätze/Ereignisse, die auf der Grundlage der auf einen Trigger angewendeten Filter erstellt werden, abhängig von der Lizenz, die der Benutzer besitzt. Um den Dienst abzurufen, verfolgt der Auslöser im Backend den Zeitstempel, wann er den Dienst zuletzt abgerufen hat, und ruft ihn alle X Minuten auf der Grundlage dieses Zeitstempels erneut ab.

Wenn ein SQL-Datensatz erstellt wird, fragt der Trigger den SQL-Dienst beispielsweise alle 1 Minute ab und erhält Informationen über alle Datensätze, die in den Minuten seit der letzten Abfrage erstellt wurden. Wenn keine Datensätze erstellt wurden, wird der Lauf übersprungen. Sie können diese Abfragen im Abschnitt "Prüfungen" auf der Seite "Laufverlauf" sehen. Wenn der Ablauf nun gestoppt wird, zum Beispiel am 13. September um 12:30 Uhr. Dieser Zeitstempel wird vermerkt, als der letzte Abruf der

Dienst, wenn der Fluss wieder eingeschaltet wird, sagen wir am 14. September um 13:30 Uhr PT, wird der Fluss den Dienst abfragen, um alle Ereignisse zu erhalten, die zwischen der letzten Abfragezeit und der aktuellen Zeit erstellt wurden (d. h., er wird alle Datensätze zwischen dem 13. September um 12:30 Uhr und dem 14. September um 13:30 Uhr abfragen). Der Grund dafür ist, dass das Ausschalten des Flusses den Auslöser nicht abmeldet. Es wird lediglich die Abfrageuhr angehalten. Dies ist beabsichtigt, da der Datenfluss manchmal fälschlicherweise oder aufgrund einer Drosselung gestoppt wird, wir aber keine Daten für den Trigger verpassen wollen.

Die einzige Möglichkeit, dieses Verhalten zu umgehen, besteht darin, eine neue Kopie des Flusses zu erstellen, da der Auslöser erneut registriert wird, wenn Sie den Fluss zum ersten Mal einschalten und den vorhandenen Auslöser löschen.

Webhook-Auslöser

Webhook-Auslöser hingegen rufen den Dienst nicht ab. Sobald sie erstellt sind, melden sie sich bei dem Dienst an, um ihm mitzuteilen, dass sie eine Benachrichtigung erhalten möchten, wenn ein bestimmtes Ereignis bei diesem Dienst eintritt.

Webhooks sind einfache HTTP-Rückrufe, die zur Bereitstellung von Ereignisbenachrichtigungen verwendet werden. Mit Power Automate können Sie Webhooks als Auslöser verwenden. Ein Ablauf wartet auf diesen Auslöser und führt eine Aktion aus, wenn der Auslöser ausgelöst wird.

Die Idee ist, dass, wenn ein Ereignis eintritt, der Dienst eine Ereignisbenachrichtigung an den Auslöser mit allen Details dieses bestimmten Ereignisses sendet und dann der Fluss ausgelöst wird. Das bedeutet,

dass der Fluss den Dienst nicht abfragen muss und eine Menge Interaktionen reduziert.

Weitere Informationen finden Sie [hier](#)

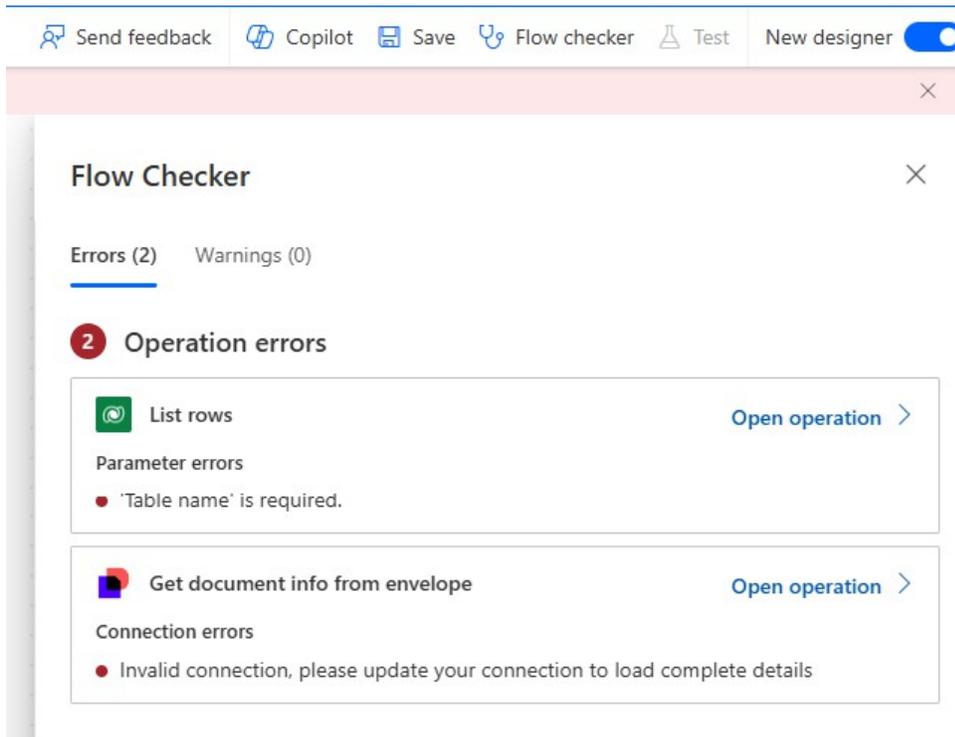
Flow Checker

Der Flow Checker in Microsoft Power Automate ist ein integriertes Tool, das Benutzern helfen soll, die Qualität und Korrektheit ihrer Cloud-Flows sicherzustellen. Es bietet Echtzeitanalysen und Feedback zu potenziellen

Probleme und trägt dazu bei, die Zuverlässigkeit und Funktionalität der Abläufe zu verbessern. Dieses Tool ist sowohl für neue als auch für erfahrene Nutzer unverzichtbar, da es den Entwicklungsprozess verbessert, indem es umsetzbare Erkenntnisse liefert.

So kann Flow Checker Machern helfen -

1. **Erkennung von Fehlern:** Der Flow Checker identifiziert Fehler, die die Ausführung des Flows verhindern, z. B. fehlende oder falsche Eingaben, falsch konfigurierte Aktionen und Konnektivitätsprobleme.
2. **Warnungen und Vorschläge:** Außerdem werden Warnungen und Vorschläge für bewährte Verfahren zur Optimierung der Leistung und der Wartbarkeit des Ablaufs angezeigt. Dazu können Empfehlungen zur Verbesserung der Logik oder zur Identifizierung veralteter Aktionen gehören.
3. **Feedback in Echtzeit:** Während Sie Ihren Ablauf erstellen oder ändern, liefert der Flow Checker Echtzeit-Feedback, so dass Sie Probleme sofort beheben können.
4. **Detaillierte Erklärungen:** Zu jedem Fehler oder jeder Warnung gibt es eine ausführliche Erläuterung und eine Anleitung zur Lösung des Problems, so dass die Benutzer die Probleme leichter verstehen und beheben können.



Mehr über Flow Checker erfahren Sie [hier](#).

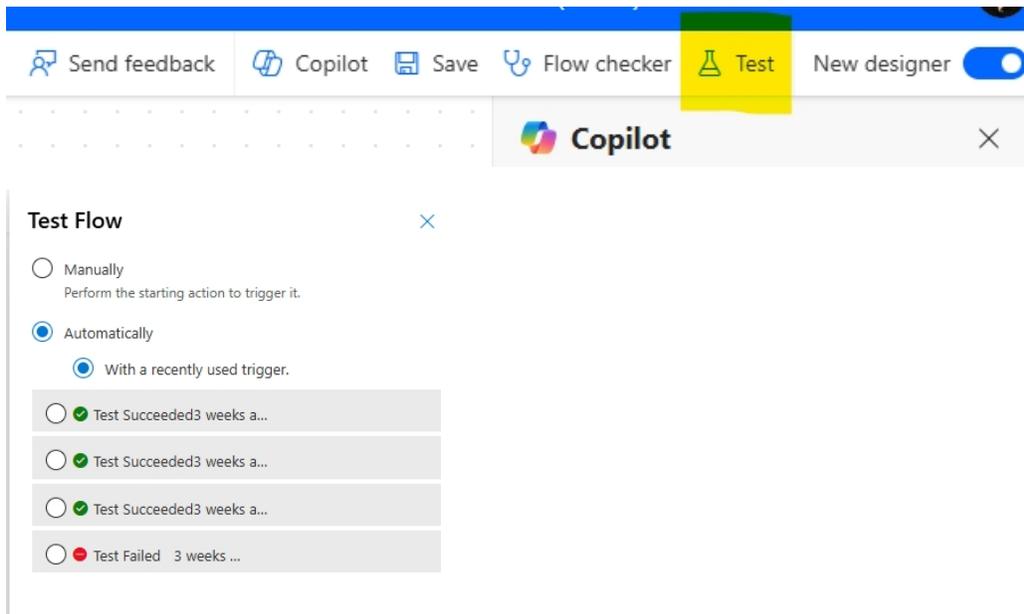
Effektives Testen von Power Automate Cloud-Flows ist wichtig, um ihre Zuverlässigkeit, Leistung und Korrektheit zu gewährleisten. Hier sind einige der Mechanismen, die Sie zum Testen Ihrer Abläufe verwenden können.

Prüfung in der Entwurfsphase

In dieser Phase sollten Sie den Flow Checker und das Test Flow Tool verwenden. Wir haben den Flow Checker im obigen Abschnitt behandelt.

Im Testbereich gibt es nun drei Optionen zum Testen Ihres Ablaufs:

- Lösen Sie den Test selbst manuell aus, indem Sie die Aktion ausführen, die den Ablauf auslöst. Sie können zum Beispiel Ihren Posteingang aufrufen und sich selbst eine Test-E-Mail schicken. Oder Sie können zu SQL gehen und eine Zeile einfügen.
- Verwenden Sie Daten aus früheren Läufen, um den Test durchzuführen.



Statische Ergebnisprüfung (Mock Data)

Verwenden Sie Scheindaten für Testaktionen, die mit externen Systemen interagieren, um unbeabsichtigte Änderungen an Produktionsdaten zu vermeiden.

Wiedervorlage von Cloud-Flow-Läufen

Manchmal kann es vorkommen, dass Cloud Flow unerwartet aufhört zu funktionieren oder aufgrund des Servers oder eines asynchronen Prozesses, der während der Ausführung einige Anforderungen nicht erfüllt hat, fehlschlägt. In diesem Fall müssen Sie den Ablauf erneut ausführen. Dies können Sie tun, indem Sie die Ausführungshistorie wählen und auf "Neu einreichen" klicken.

Wir haben auch die Möglichkeit, [Läufe in großen Mengen erneut zu übermitteln oder zu stornieren](#).

Start time	Duration	+ Add column	Status
May 8, 11:48 AM (3 wk ago)	00:00:49		Test succeeded
May 8, 11:45 AM (3 wk ago)	00:00:29		Test succeeded
May 8, 11:44 AM (3 wk ago)	00:00:03		Test succeeded
<input checked="" type="radio"/> May 8, 11:42 AM (3 wk ago)	00:00:02		Test failed

Überlegungen zur Wiedervorlage von Läufen

- Fehlerbehandlung:** Stellen Sie sicher, dass Ihr Ablauf über eine angemessene Fehlerbehandlung und Wiederholungsrichtlinien verfügt. Dies kann dazu beitragen, Probleme, die den anfänglichen Fehler verursacht haben, zu entschärfen und bessere Erkenntnisse darüber zu gewinnen, warum ein Lauf wieder scheitern könnte.
- Datendopplung:** Achten Sie auf mögliche Datenduplikationen oder andere Nebeneffekte, die sich aus der erneuten Übermittlung eines Laufs ergeben können. Wenn Ihr Ablauf beispielsweise Aktionen zum Erstellen von Datensätzen oder zum Senden von E-Mails enthält, könnte die erneute Übermittlung des Laufs diese Aktionen wiederholen.
- Eingabedaten:** Vergewissern Sie sich, dass die für den Flusslauf erforderlichen Eingabedaten noch gültig und verfügbar sind. Änderungen an den Daten oder der Umgebung (z. B. gelöschte Dateien oder Datensätze) können das Ergebnis des erneut eingereichten Laufs beeinflussen.
- Aktualisierungen der Bewegung:** Wenn Sie seit dem ursprünglichen Lauf Änderungen an der Bewegung vorgenommen haben, überlegen Sie, ob sich diese Änderungen auf die Wiedervorlage auswirken könnten. In der Regel ist es am besten, die Wiedervorlage mit der Version der Bewegung vorzunehmen, die zum Zeitpunkt des ursprünglichen Laufs vorhanden war, um die Konsistenz zu gewährleisten.

Profi-Tipps

Peek-Code/Code-Ansicht

Peek Code oder Code View ist eine Funktion, mit der Benutzer die zugrunde liegende JSON-Darstellung der Aktionen und Auslöser innerhalb eines Ablaufs anzeigen können. Diese Funktion ist besonders nützlich für fortgeschrittene Benutzer, die die genaue Struktur und Konfiguration ihrer Abläufe

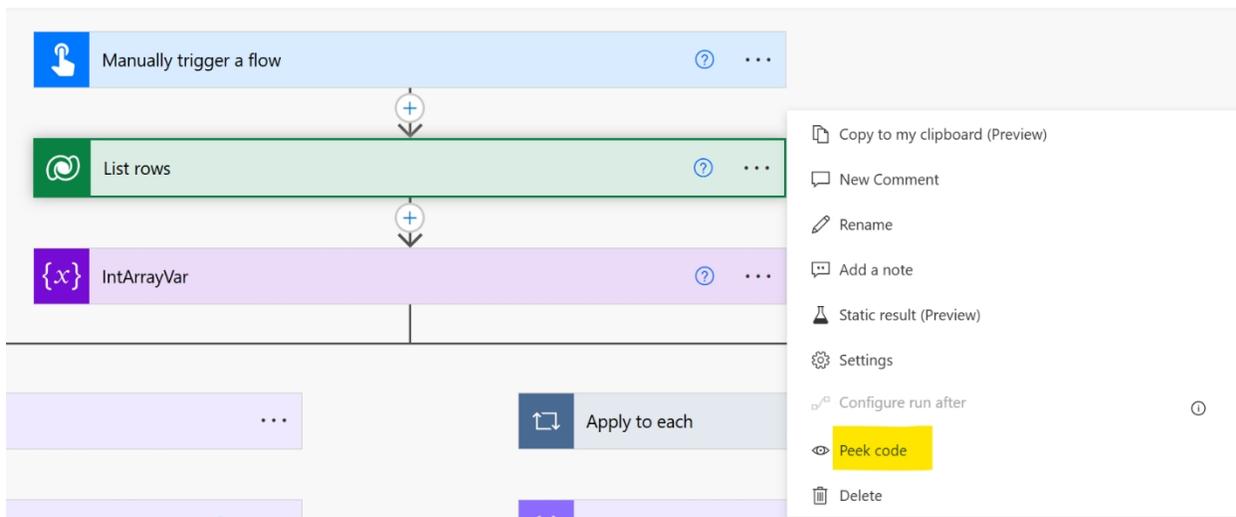
verstehen, Probleme beheben oder manuell bearbeiten möchten.
spezifische Details, die über die visuelle Standardschnittstelle nicht leicht
zugänglich sind. Verwenden Sie im neuen Designer die Codeansicht, um den
Auslöser-/Aktionscode anzuzeigen

```

1 {
2   "type": "OpenApiConnection",
3   "inputs": {
4     "parameters": {
5       "entityName": "cr38e_rasikarealestatetransactiones",
6       "$filter": "cr38e_city ne ''",
7       "$top": 100
8     },
9     "host": {
10      "apiId": "/providers/Microsoft.PowerApps/apis/
11      shared_commondataserviceforapps",
12      "connection": "shared_commondataserviceforapps",
13      "operationId": "ListRecords"
14    }
15  },
16  "runAfter": {},
17  "metadata": {
18    "operationMetadataId": "3b0ec823-7eb8-49b3-aadc-7ba978ce81bc"
19  }
20 }

```

Verwenden Sie im klassischen Designer den Peek-Code, um den JSON-Code anzuzeigen



Darüber hinaus kann die Codeansicht/der Peek-Code verwendet werden, um die Triggerabfragefrequenz, den Triggertyp, die Odata-Abfragen und sogar die JSON-Darstellung des dynamischen Inhalts usw. anzuzeigen.



When expense is submitted



Parameters Settings Code View About

```
1 {
2   "type": "OpenApiConnectionWebhook",
3   "inputs": {
4     "parameters": {
5       "subscriptionRequest/message": 4,
6       "subscriptionRequest/entityname": "cr38e_expensetravel",
7       "subscriptionRequest/scope": 4,
8       "subscriptionRequest/name":
9         "492850ea-7f0d-ef11-9f89-7c1e520d210d"
10    },
11   "host": {
12     "apiId": "/providers/Microsoft.PowerApps/apis/
13       shared_commondataserviceforapps",
14     "connection": "shared_commondataserviceforapps",
15     "operationId": "SubscribeWebhookTrigger"
16   },
17   "conditions": [
18     {
19       "expression": "@greaterOrEquals(triggerBody()?['cr38e_amount'],
20         '5000')"
21     }
22 ],
23   "metadata": {
24     "operationMetadataId": "9512511b-22cf-4335-93cc-e374a80805fa"
25   }
26 }
```

Eine Notiz hinzufügen

Das Hinzufügen von Notizen oder Kommentaren in Power Automate Cloud-Abläufen ist eine hilfreiche Methode, um den Zweck, die Logik und spezifische Details verschiedener Aktionen und Auslöser innerhalb des Ablaufs zu dokumentieren. Diese Praxis ist sowohl für einzelne Benutzer als auch für Teams von Vorteil, da sie die Lesbarkeit und Wartbarkeit der Abläufe verbessert.

← When expense is submitted -> Start and wait for an approval



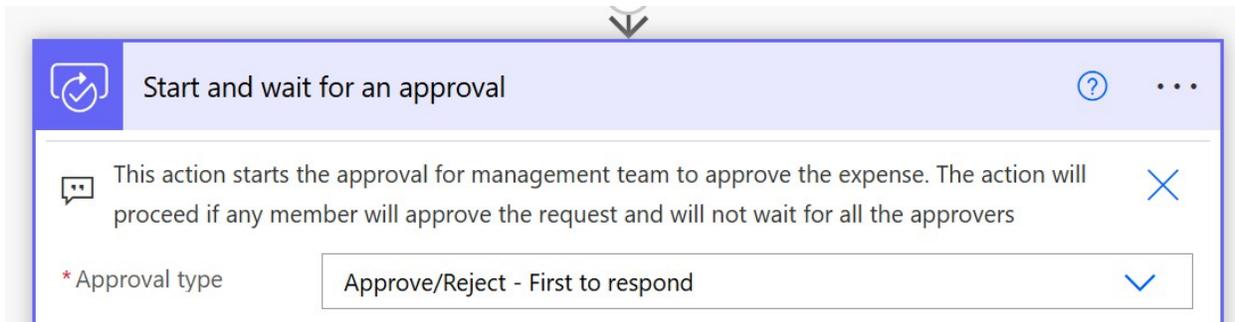
Start and wait for an approval



Add a note

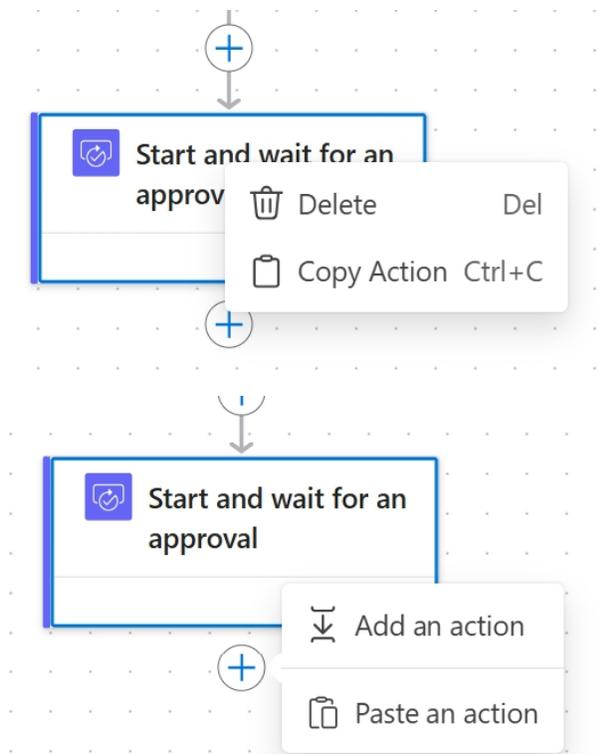
Delete

Parameters Settings Code View Testing About

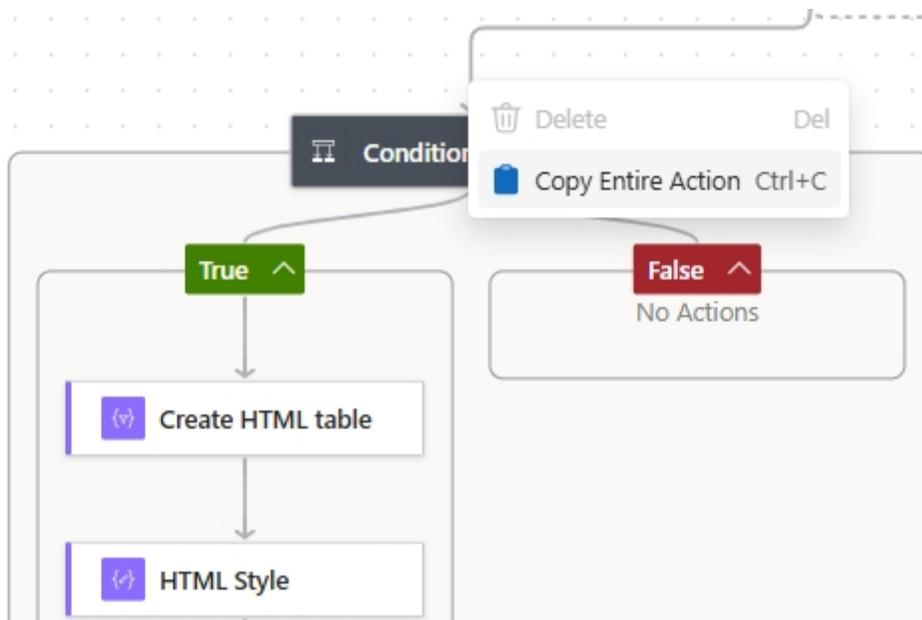


Kopieren Einfügen Aktionen

Das Kopieren und Einfügen von Aktionen ist eine nützliche Methode, um dieselbe Aktion mehrmals innerhalb eines Ablaufs oder über mehrere Abläufe hinweg zu verwenden. Dies vereinfacht den Prozess der Erstellung und Änderung von Abläufen und spart Zeit und mehrere Klicks.



Der neue Designer in Power Automate ermöglicht das Kopieren und Einfügen ganzer Bedingungen oder Bereichsaktionen.



Fehlerbehebung bei Cloud-Flüssen

Ausführliche Dokumente zur Fehlerbehebung für Power Automate Cloud-Flows sind im

Folgenden aufgeführt: [Fehlersuche Auslöser](#)
[Fehlerbehebung bei Cloud-Flüssen](#)

Hier sind einige wichtige praktische Tipps

Verstehen Sie die Fehlercodes

Die Fehlercodes 400s sind etwas, das auf der Benutzeroberfläche korrigiert werden kann.

400 - Schlechte Anfrage - Dies kann aufgrund einer falschen Konfiguration auf der Auslöser- oder Aktionsebene geschehen

401 - Zugriff verweigert - Benutzer hat keinen Zugriff auf den Dienst

403 - Verboten - Der Benutzer hat Zugriff auf den Dienst, aber es ist ihm verboten, auf einen bestimmten Endpunkt zuzugreifen

404 - Nicht gefunden - Die aufrufende Ressource existiert nicht.

Die Fehlercodes 500 oder 502 sind servicebezogen und können vorübergehend sein und sich von selbst beheben. Diese Fehler sollten auf der Serviceebene überprüft werden.

Verständnis der Plattformgrenzen/Vermeidung der Drosselung

Das Verständnis der Grenzen von Power Automate und Power Platform kann den Benutzern helfen, skalierbare Power Automate Flows zu entwerfen.

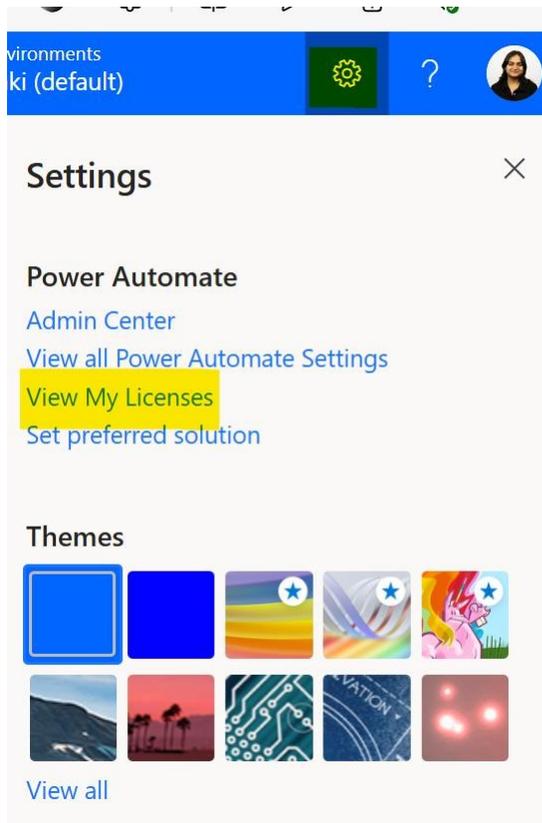
Diese Grenzwerte können die Leistung des Datenflusses bestimmen und dazu beitragen, die Drosselung (Verlangsamung) oder Abschaltung von Datenflüssen aufgrund von Verletzungen der Anforderungsgrenzen zu vermeiden. Jeder Fluss, der 14 Tage lang kontinuierlich gedrosselt wird, wird abgeschaltet. Diese Bewegungen können jederzeit wieder aktiviert werden. Weitere Einzelheiten finden Sie unter [Retention Limits](#).

Wenn Sie mehr über die Plattform und die automatisierten Leistungsgrenzen erfahren möchten, lesen

Sie die folgenden Dokumente: [Grenzen und Zuweisungen für API-Anfragen](#)
[Power Automate Grenzwerte und Konfigurationen](#)

Wie Sie Ihre Lizenzpläne überprüfen können

Einige der Plattform- oder API-Beschränkungen hängen von dem Lizenzplan ab, den der Benutzer hat. Der einfachste Weg, Ihre Lizenz zu identifizieren, ist über Einstellungen -> Meine Lizenzen anzeigen



Licenses ×

My licenses

- 🔗 Power Automate Premium
- 🔗 Power Automate for Project Shimla
- 🔗 Power Automate For CCI Bots
- 🔗 Power Automate for Office 365
- ▶ Power Automate for Dynamics 365
- 🔗 Power Automate for Dynamics 365 vTrial
- 🔗 Power Automate P2 Viral

Environment capacities

- 🔗 Power Automate Process
- 🔗 AI Builder add-on

Capabilities

- ✓ Standard connectors
- ✓ Premium connectors
- ✓ Custom connectors
- ✓ On-premise connectors
- ✓ Robotic process automation (RPA) - attended
- ✓ Robotic process automation (RPA) - unattended
- ✓ Robotic process automation (RPA) - hosted
- ✓ Business process flows
- ✓ AI Builder
- ✓ Process mining

Alternativ können Sie auch die Tastenkombination CNTRL + ALT+ A im Power Automate-Portal drücken, um tiefere Details auf Planebene zu erhalten.

API-Anforderungsgrenzen

Anfragen in Microsoft Power Platform bestehen aus verschiedenen Aktionen, die ein Benutzer in verschiedenen Produkten durchführt. Auf einer hohen Ebene wird im Folgenden beschrieben, was eine API-Anfrage in Power Automate ausmacht

Alle API-Anfragen an Konnektoren, Prozessberateranalysen, HTTP-Aktionen und integrierte Aktionen, von der Initialisierung von Variablen bis hin zu einer einfachen Compose-Aktion. Sowohl erfolgreiche als auch fehlgeschlagene Aktionen werden auf diese Grenzwerte angerechnet. Außerdem zählen Wiederholungen und andere Anfragen aus der Paginierung ebenfalls als Aktionsausführungen. Weitere Informationen finden Sie unter [Was zählt als Power Platform-Anfrage?](#)

Je nach Lizenzplan gibt es Grenzen für die Anzahl der Aktionen, die ein Cloud-Flow pro Tag ausführen kann. Diese Grenzen unterscheiden sich von den Grenzen für die Verbindungsdrosselung. Sie können die

Anzahl der Aktionen, die Ihr Fluss ausführt, sehen, indem Sie auf der Seite mit den Flussdetails **Analytics** auswählen und die Registerkarte **Aktionen** aufrufen.

Auch wenn der Fluss weniger Power Platform-Anfragen verwendet, können Sie an Ihre Grenzen stoßen, wenn der Fluss häufiger als erwartet ausgeführt wird. Sie könnten zum Beispiel einen Cloud-Flow erstellen, der Ihnen eine Push-Benachrichtigung sendet, sobald Ihr Manager Ihnen eine E-Mail schickt. Dieser Ablauf muss jedes Mal ausgeführt werden, wenn Sie eine E-Mail (von wem auch immer) erhalten, da der Ablauf prüfen muss, ob die E-Mail von Ihrem Manager stammt. Das Limit gilt für alle Abläufe aller Ihrer Abläufe in einem Zeitraum von 24 Stunden. Im Folgenden finden Sie einige Richtlinien zur Schätzung der Anforderungsnutzung einer Bewegung.

- Eine oder mehrere Aktionen werden als Teil eines Ablaufs ausgeführt. Ein einfacher Ablauf mit einem Auslöser und einer Aktion führt jedes Mal, wenn der Ablauf läuft, zu zwei "Aktionen", die zwei Anfragen verbrauchen.
- Jeder Auslöser/Aktion im Ablauf erzeugt Power Platform-Anfragen. Alle Arten von Aktionen wie Connector-Aktionen, HTTP-Aktionen, integrierte Aktionen (von der Initialisierung von Variablen über die Erstellung von Bereichen bis hin zu einer einfachen Compose-Aktion) erzeugen Power Platform-Anfragen. Bei einem Fluss, der beispielsweise SharePoint, Exchange, Twitter und Dataverse verbindet, werden all diese Aktionen auf die Power Platform-Anfragegrenzen angerechnet.
- Sowohl erfolgreiche als auch fehlgeschlagene Aktionen werden auf diese Grenzwerte angerechnet. Übersprungene Aktionen werden nicht auf diese Grenzwerte angerechnet.
- Jede Aktion erzeugt eine Anfrage. Befindet sich die Aktion in einer "apply to each loop"-Schleife, erzeugt sie bei der Ausführung der Schleife weitere Power Platform-Anfragen.
- Eine Aktion kann mehrere Ausdrücke haben, wird aber als eine API-Anfrage gezählt.
- Wiederholungen und zusätzliche Anfragen aus der Paginierung zählen ebenfalls als Aktionsausführungen.

Hier ein kurzer Überblick über die API-Limits auf Plattformebene auf der Grundlage der Benutzerlizenz.

Hinweis: Derzeit befinden sich alle Kunden in einer Übergangsphase, in der die API-Limits nicht vollständig durchgesetzt werden und höher sind. Dieser Übergangszeitraum endet, sobald die Berichte des Power Platform Admin Centers GA sind. Unternehmen haben sechs Monate Zeit, ihre Nutzung zu analysieren und entsprechende Lizenzen zu erwerben, bevor die strenge Durchsetzung der Lizenzgrenzen beginnt. Weitere Informationen über die Übergangszeit finden [Sie hier](#)

Übergangszeitraum Vergleich:

License	Power platform actual limits per 24 hours	Power platform transition period limits per 24 hours
Power Automate premium	40K/user	200K/flow
Power Automate process plan	250K/process	500K/process
Office 365	6K/user	10K/flow
Power Apps premium	40K/user	200K/flow
Dynamics 365 professional	40K/user	200K/flow
Dynamics 365 Enterprise applications	40K/user	200K/flow
Dynamics 365 Team member	6K/user	10K/flow

Zusätzliche Angaben zum Datenverbrauch:

Plan	Limits per 24 hours	Data consumption per day
Office 365 Flow licenses, Power Apps per app, and Dynamics team member and trials	Final limit: 6,000 actions across all flows created by a single user. Transition limit: 10,000 actions per flow	1 GB across all flows created by a single user.
Power Automate Premium, Power Apps Premium, Power Automate Per user, Power Automate Per user with attended RPA, and Power Apps per user	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Dynamics Professional licenses	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Dynamics Enterprise Application licenses	40,000 actions across all flows created by a single user. Transition limit: 200,000 actions per flow	10 GB across all flows created by a single user.
Power Automate Process license, and Power Automate per flow license	250,000 actions per process. Transition limit: 500,000 actions per flow	50 GB storage per flow.

Das Wichtigste zur Erinnerung

- Benutzergrenzen werden auf alle Ströme angewendet
 - Automatische Flüsse, die sie geschaffen haben
 - Manuelle Abläufe, die sie ausgeführt haben
- Grenzwerte für Bewegungen/Prozesse werden auf die einzelnen Bewegungen angewendet
 - Pro-Prozess kann auf eine Gruppe von zusammenhängenden Bewegungen angewendet werden
 - Pro Fluss gilt für einen einzelnen Fluss
- Pay-Go-Umgebungen
 - Die Flüsse haben keine tägliche Quote, da man für den Verbrauch bezahlt

API-Durchsatzgrenzen für Konnektoren

Zusätzlich zu den Plattformgrenzen hat jeder Konnektor-Dienst seine eigenen Grenzen. Die Konnektor-Drosselung in Power Automate bezieht sich auf den Mechanismus, mit dem Konnektoren Ratenbegrenzungen oder Nutzungsquoten durchsetzen, um Missbrauch zu verhindern und eine faire Ressourcenzuweisung zu gewährleisten. Wenn ein Konnektor gedrosselt wird, beschränkt er die Anzahl der Anfragen oder Operationen, die innerhalb eines bestimmten Zeitrahmens durchgeführt werden können.

Wenn ein Datenfluss an die Grenzen der Drosselung auf Verbindungsebene stößt, gibt der Dienst einen Fehlercode "429 (Too Many Requests)" mit einem Fehlertext wie "Rate limit is exceeded. Versuchen Sie es in 27 Sekunden erneut".

Jeder [Connector](#) hat seine eigene Drosselungsgrenze. Hier ist ein Beispiel für Teams Connector -

← ↻ 🏠 📁 https://learn.microsoft.com/en-us/connectors/teams/ 📁

🔍 Filter by title

- Microsoft Security Copilot
- Microsoft Teams**
- Microsoft Teams Virtual Events
- Microsoft To-Do (Business)
- Microsoft To-Do (Consumer)
- Microsoft Translator [DEPRECATED]
- Microsoft Translator V2
- Microsoft Translator V3
- Mime Automation

Throttling Limits

🔄 Expand table

Name	Calls	Renewal Period
API calls per connection	100	60 seconds
Frequency of trigger polls	1	900 seconds
Non-Get requests per connection	300	3600 seconds

Dataverse API-Grenzen

Dataverse als Konnektordienst hat seine eigenen [Dienstschutzgrenzen](#) definiert. Die API-Grenzwerte für den Dienstschutz werden pro Benutzer ausgewertet. Beim Aufruf durch einen Fluss ist der "Benutzer" derjenige, der mit der Aktion verbunden ist. In der Regel ist dies der Flow-Eigentümer, kann aber auch der aufrufende Benutzer sein, wenn der aufrufende Benutzerkontext in der Aktion verwendet wird.

Measure	Description	Limit per web server
Number of requests	The cumulative number of requests made by the user.	6000 within the 5 minute sliding window
Execution time	The combined execution time of all requests made by the user.	20 minutes (1200 seconds) within the 5 minute sliding window
Number of concurrent requests	The number of concurrent requests made by the user	52 or higher

Grenzen der Gleichzeitigkeit von Flüssen

Bei der Entwicklung skalierbarer, effizienter Abläufe müssen Sie die Grenzen der Gleichzeitigkeit, der Schleifenbildung und des Debatching kennen, um unnötige Verzögerungen zu vermeiden. Hier sind die Grenzen für einen einzelnen Lauf:

Name	Limit	Notes
Concurrent runs	- Unlimited for flows with Concurrency Control turned off - 1 to 100 when Concurrency Control is turned on (defaults to 25)	This is the limit for how many runs a flow can have at the same time. Note: Concurrency Control is set in the flow's trigger settings and is off by default. Turning on Concurrency Control can't be undone without deleting and re-adding the trigger.
Waiting runs	- Not applicable when Concurrency Control is off - 10 plus the degree of parallelism (1-100) when Concurrency Control is on	This limit describes the highest number of flow runs that can be queued when the flow is at its maximum number of concurrent runs. Note: Additional triggers that arrive while the waiting runs limit is met might be re-tried by the connector. However, the retry attempts might not succeed if the maximum waiting limit continues to be met for an extended period of time. To ensure all triggers result in flow runs, leave the Concurrency Control setting off in the flow's trigger.
Apply to each array item	5,000 for Low, 100,000 for all others	This limit describes the highest number of array items that an "apply to each" loop can process. To filter larger arrays, you can use the query action.
Apply to each concurrency	1 is the default limit. You can change the default to a value between 1 and 50 inclusively.	This limit is highest number of "apply to each" loop iterations that can run at the same time, or in parallel.
Split on items	- 5,000 for Low without trigger concurrency - 100,000 for all others without trigger concurrency - 100 with trigger concurrency	For triggers that return an array, you can specify an expression that uses a 'SplitOn' property that splits or debatches array items into multiple workflow instances for processing, rather than use a "Foreach" loop. This expression references the array to use for creating and running a workflow instance for each array item. Note: When concurrency is turned on, the Split on limit is reduced to 100 items.
Until iterations	- Default: 60 - Maximum: 5,000	
Paginated items	5,000 for Low, 100,000 for all others	To process more items, trigger multiple flow runs over your data.

Aktion Burst-Grenzen

Action Burst Limits beziehen sich auf die maximale Anzahl von Aktionen, die innerhalb eines bestimmten Zeitraums ausgelöst werden können, in der Regel gemessen in einem rollenden Zeitfenster. Gegenwärtig gibt es eine Obergrenze von 100.000 Aktionen pro Fluss und 5 Minuten. Jegliche Auslösungen oder Schleifen können zu einer Überschreitung dieser Grenze führen, was eine Verlangsamung/Drosselung des Datenflusses zur Folge hat.

Die beste Lösung wäre, die Last auf mehrere Ströme zu verteilen.

- Auslösebedingungen verwenden
- Verwenden Sie die oben in diesem Whitepaper dokumentierten Best Practices für die Gestaltung der einzelnen Schleifen.

Zusätzliche Grenzwerte

Durchfluss Bemessungsgrenzen

Bei der Gestaltung von Abläufen können Benutzer auf einige Grenzen stoßen, die auf der Entwurfs-/Definitionsebene definiert sind. Erwägen Sie eine Neugestaltung Ihrer Abläufe, wenn Sie auf eine dieser Grenzen stoßen:

Name	Limit	Notes
Actions per workflow	500	Flows with a large number of actions may encounter performance issues while you edit them, even if they have fewer than 500. Consider using child flows to reduce the number of actions in a single flow or if you need more than 500.
Allowed nesting depth for actions	8	Add child flows if you need more than eight levels of nesting.
Switch scope cases limit	25	
Variables per workflow	250	
Length of <code>action</code> or <code>trigger</code> name	80 characters	
Characters per expression	8,192	

Grenzwerte für die Flusserzeugung

The following table describes the limit for the My flows and Team flows tabs.

 Expand table

Name	Limit	Notes
Number of flows owned by a single user	600	Use flows under solutions if you need more than 600.

Grenzwerte für die Durchflusszeitüberschreitung

The following table describes the duration limits for a single flow run.

 Expand table

Name	Limit	Notes
Run duration	30 days	Run duration is calculated using a run's start time and includes flows with pending steps like approvals. After 30 days, any pending steps time out.
Run retention in storage	30 days	Run retention is calculated using a run's start time.
Minimum recurrence interval	60 seconds	
Maximum recurrence interval	500 days	
Minimum postpone interval	5 seconds for Low, 1 second for all other performance profiles	

Sichere Abläufe entwerfen

Ebenso wichtig wie die Erstellung eines gut durchdachten, automatisierten Stromflusses, der die erforderlichen Ergebnisse effizient generiert, ist es, sicherzustellen, dass der Fluss gesichert ist und sensible Daten geschützt werden, der unbefugte Zugriff verhindert und die Einhaltung von Sicherheitsstandards gewährleistet wird.

Sichere Eingaben/Ausgaben verwenden

Derzeit bietet Power Automate den Benutzern die Möglichkeit, die Historie der Flussläufe einzusehen und tief in einen Trigger oder Informationen über die Ein- und Ausgänge der Aktion.

Table Name *

Advanced parameters

Showing 2 of 8 Show all Clear all

Select Columns

Row Count

```

graph TD
    A[Manually trigger a flow] --> B[List rows]
    
```

List_rows

```

List_rows
{
  "statusCode": 200,
  "headers": {
    "Cache-Control": "no-cache",
    "Vary": "Accept-Encoding",
    "Set-Cookie": "ARRAffinity=4bf30cb2bc3d64dca59ed2d915bb93fb2213b2b087164d0",
    "x-ms-service-request-id": "[REDACTED]",
    "Strict-Transport-Security": "max-age=31536000; includeSubDomains",
    "REQ_ID": "[REDACTED]",
    "CRM.ServiceId": "CRMAppPool",
    "AuthActivityId": "[REDACTED]",
    "x-ms-dop-hint": "4",
    "x-ms-ratelimit-time-remaining-xrm-requests": "1,195.35",
    "x-ms-ratelimit-burst-remaining-xrm-requests": "7998",
    "mise-correlation-id": "[REDACTED]",
    "OData-Version": "4.0",
    "Preference-Applied": "odata.include-annotations=\"*\"",
    "X-Source": "1734019518597233163204248230120321632141999620116214418219625",
    "Public": "OPTIONS,GET,HEAD,POST",
    "Date": "Thu, 15 Aug 2024 01:18:19 GMT",
    "Allow": "OPTIONS,GET,HEAD,POST",
    "Content-Type": "application/json; odata.metadata=full",
    "Content-Length": "1134",
    "Expires": "-1"
  },
  "body": {
    "@odata.context": "https://[REDACTED]/api/data/v9.1/$me",
    "#Microsoft.Dynamics.CRM.CreateMultiple": {
      "title": "CreateMultiple",
      "target": "https://[REDACTED]/api/data/v9.1/account"
    },
    "#Microsoft.Dynamics.CRM.DeleteMultiple": {
      "title": "DeleteMultiple",
      "target": "https://[REDACTED]/api/data/v9.1/account"
    },
    "#Microsoft.Dynamics.CRM.UpdateMultiple": {
      "title": "UpdateMultiple",
      "target": "https://[REDACTED]/api/data/v9.1/account"
    },
    "#Microsoft.Dynamics.CRM.UpsertMultiple": {
      "title": "UpsertMultiple",
      "target": "https://[REDACTED]dynamics.com/api/data/v9.1/account"
    },
    "@Microsoft.Dynamics.CRM.totalrecordcount": -1,
    "@Microsoft.Dynamics.CRM.totalrecordcountlimitexceeded": false,
    "@Microsoft.Dynamics.CRM.globalmetadataversion": "4572998",
    "value": []
  }
}

```

Mit den Einstellungen für die sichere Eingabe und Ausgabe können Sie jedoch verhindern, dass Benutzer sensible Informationen sehen.

List rows

Parameters **Settings** Code View About

General

Action Timeout
Specify the duration in ISO 8601 format
Example: P1D

Networking

Content Transfer
Specify the behavior and capabilities for transferring content over HTTP.

Chunk size
Specify download chunk size between and Mb. Example: 10

Pagination
Retrieve items to meet the specified threshold by following the continuation token. Due to connector's page size, the number returned may exceed the threshold.
 Off

Retry Policy
A retry policy applies to intermittent failures, characterized as HTTP status codes 408, 429, and 5xx, in addition to any connectivity exceptions. The default is an exponential interval policy set to retry 4 times.
Default

Security

Secure Inputs
 On

Secure Outputs
 On

List rows

+



List rows



Parameters Settings Code View About

INPUTS

Content not shown due to security configuration.

OUTPUTS

Content not shown due to security configuration.

PROPERTIES

Einige der besten Praktiken zur Sicherung Ihrer Daten in Power Automate sind:

- Vermeiden Sie es, sensible Informationen direkt in Ihre Abläufe oder Variablen zu kodieren. Verwenden Sie stattdessen Secure Inputs/Outputs, um diese Daten sicher zu speichern und abzurufen.
- Ziehen Sie die Integration von Power Automate mit Azure Key Vault in Betracht, um die Sicherheit und die zentrale Verwaltung von Geheimnissen zu verbessern.
- Mit Azure Key Vault können Sie vertrauliche Informationen sicher speichern und verwalten, und Power Automate kann zur Laufzeit mithilfe von verwalteten Identitäten oder Dienstprinzipien Geheimnisse aus Key Vault abrufen.

Absicherung des HTTP-Anfrageauslösers

Sie können den Auslöser **Wenn eine HTTP-Anfrage empfangen wird** verwenden, um Workflows auszulösen, indem Sie eine Anfrage an eine HTTP-Anfrage an den Endpunkt senden, der vom Ablauf generiert wird. Sie können einschränken, welche Benutzer diesen Workflow auslösen können, indem Sie sicherstellen, dass nur authentifizierte Benutzer diesen Workflow auslösen können.

Einer der Ansätze ist die Verwendung von AAD-Token. Dieses Token kann definiert werden, um bestimmte Benutzer/Hauptverantwortliche in einem Mandanten oder jeden Benutzer innerhalb eines Mandanten einzuschränken. Weitere Informationen finden [Sie hier](#)

When an HTTP request is received

Parameters Settings Code View About

Who Can Trigger The Flow? *

Any user in my tenant

Any user in my tenant

Anyone

Specific users in my tenant

Enter custom value

Use sample payload to generate schema

Advanced parameters

Showing 0 of 2

Show all Clear all

Der zweite Ansatz ist die Verwendung von IP-Pinning. Umgebungsadministratoren können IP-Sets oder Bereiche konfigurieren, die mit Power Platform-Ressourcen interagieren dürfen. Dies ist derzeit in der Vorschau. Weitere Informationen finden Sie hier - [IP-Firewall in Power Platform-Umgebungen](#)

Generische Flow-Konfiguration beibehalten

Umgebungsvariablen

Wenn ein Power Automate-Ablauf über eine Lösung oder als Zip-Paket exportiert wird, ist es sehr einfach, die Ablaufdefinition zu öffnen, und alle hart kodierten Werte wie Kennwörter oder Geheimnisse können offengelegt werden. Um dies zu vermeiden, verwenden Sie [Umgebungsvariablen](#).

Dienstleistungsprinzipien

Ein Service-Prinzipal ist eine nicht-menschliche Sicherheitsidentität, die eine Anwendung oder einen Dienst repräsentiert, der Ressourcen in Azure und der Power Platform besitzen und verwalten kann. Um einen Service Principal in der Power Platform zu verwenden, muss ein Service Principal-Anwendungsbenutzer erstellt werden, der den Service Principal über das Portal oder die API repräsentiert. Ein Anwendungsbenutzer kann über Verbindungen verfügen, die für ihn freigegeben sind, und Ressourcen wie Flows besitzen.

Verwenden Sie Dienstprinzipien für die Eigentümerschaft von Flüssen und schaffen Sie Verbindungen, wo immer dies möglich ist. Um mehr über Serviceprinzipien zu erfahren, lesen Sie

[hier.](#)

Konfigurieren von AAD/Entra-Zugangsberechtigungsrichtlinien

Power Automate Richtlinien für den bedingten Zugriff können zusätzlich zu DLP, HTTP OAuth und IP-Pinning erstellt werden, um die Datenexfiltration zu verhindern.

Zu den Mindestzielgruppen, die in die Zugangskontrollpolitik aufgenommen werden sollten, gehören:

- Ihre eindeutige Dataverse-Org-Zielgruppe (z. B. <https://{ihre-org}.crm.dynamics.com/>)
- Energieplattform: <https://api.powerplatform.com>
- Power Automate: <https://service.flow.microsoft.com/>
- Power Apps: <https://service.powerapps.com/>
- Verbindungen: <https://apihub.azure.com>

Weitere Einzelheiten zu den Auswirkungen der Zugangskontrollpolitik auf die Verkehrsströme finden Sie [hier](#)

Verständnis des Zugangs zu Strömen

Eigentümerschaft von Flows (SPN vs. Benutzer)

Bei der Verwaltung von Power Automate-Abläufen ist die Wahl zwischen der Verwendung eines Service Principal Name (SPN) oder eines Benutzerkontos als Eigentümer des Ablaufs entscheidend. Die Verwendung eines SPN für die Eigentümerschaft von Abläufen bietet Konsistenz und Sicherheit, da SPNs nicht an einzelne Benutzer gebunden sind, wodurch sichergestellt wird, dass die Abläufe auch bei personellen Veränderungen reibungslos weiterlaufen. Diese Einrichtung verringert das Risiko von Unterbrechungen und ermöglicht eine strengere Kontrolle der Berechtigungen, da SPNs nur der notwendige Zugriff gewährt werden kann.

Service Principal Name (SPN) Vorteile für den

Eigentümer:

- **Konsistenz:** SPNs sind nicht an einen einzelnen Benutzer gebunden, wodurch sichergestellt wird, dass die Abläufe unabhängig von personellen Veränderungen (z. B. wenn ein Benutzer das Unternehmen verlässt) weiterlaufen.
- **Sicherheit:** SPNs können mit den minimal erforderlichen Berechtigungen konfiguriert werden, wodurch das Risiko eines unbefugten Zugriffs verringert wird.
- **Skalierbarkeit:** SPNs eignen sich besser für die Verwaltung automatisierter Aufgaben in großen Umgebungen, insbesondere in Szenarien mit mehreren Datenströmen oder Umgebungen.
- **Einhaltung der Vorschriften:** Die Verwendung von SPNs kann zur Einhaltung von Vorschriften beitragen, da sie einen klareren Prüfpfad für die vom Datenfluss durchgeführten Aktionen bieten, unabhängig von bestimmten Benutzern.

Vorteile des Besitzes von

Benutzerkonten:

- Einfaches Einrichten: Abläufe können von den Nutzern mit minimalem Aufwand erstellt und verwaltet werden, was es den Nutzern erleichtert, ihre eigenen Abläufe zu entwickeln und zu verwalten.
- Menschliche Interaktion: Wenn der Ablauf Aufgaben beinhaltet, die eine menschliche Genehmigung, Entscheidungsfindung oder Interaktion erfordern, ist die Verwendung eines Benutzerkontos möglicherweise besser geeignet.
- Personalisierter Kontext: Flows, die Benutzern gehören, können im Kontext des jeweiligen Benutzers ausgeführt werden, wobei dessen Berechtigungen und Einstellungen verwendet werden.

Bewährte Praktiken

- Kritische oder langlaufende Flows: Verwenden Sie einen SPN, um die Stabilität des Flusses und die Unabhängigkeit von bestimmten Benutzern zu gewährleisten.
- Benutzerspezifische oder interaktive Abläufe: Verwenden Sie ein Benutzerkonto, wenn der Ablauf den spezifischen Kontext eines Benutzers erfordert oder eine Benutzerinteraktion beinhaltet.

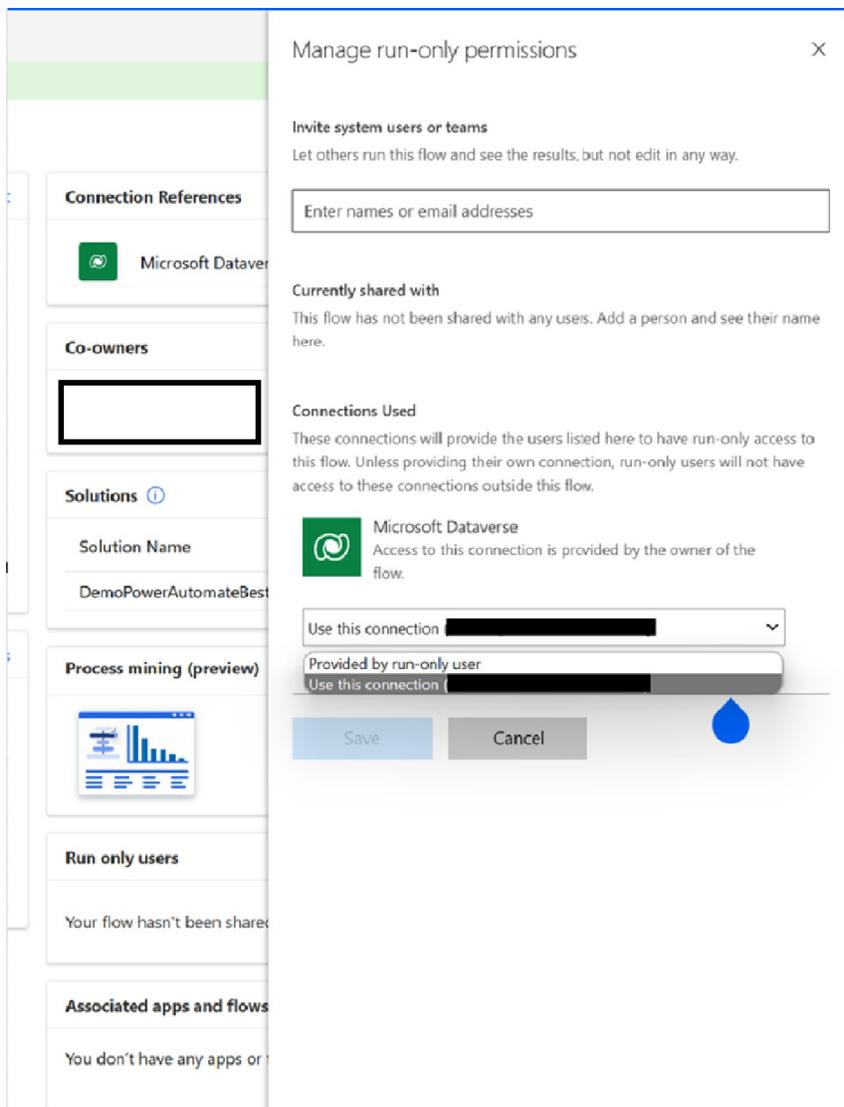
Miteigentümerschaft an Strömen

Das Hinzufügen eines [Miteigentümers](#) zu einem Cloud-Flow ist die häufigste Art, einen Cloud-Flow gemeinsam zu nutzen. Jeder Eigentümer eines Cloud-Flows kann diese Aktionen durchführen:

- Sehen Sie sich die Laufhistorie an.
- Verwalten Sie die Eigenschaften des Flusses (z. B. Starten oder Anhalten des Flusses, Hinzufügen von Besitzern oder Aktualisieren von Anmeldeinformationen für eine Verbindung).
- Bearbeiten Sie die Definition des Ablaufs (z. B. Hinzufügen oder Entfernen einer Aktion oder Bedingung).
- Hinzufügen oder Entfernen anderer Besitzer (aber nicht des Erstellers der Bewegung), einschließlich Gastbenutzer.
- Löschen Sie die Bewegung.

Fügen Sie daher nur bei Bedarf Miteigentümer für die Zusammenarbeit von Abläufen hinzu. In den meisten Fällen, wenn eine Bewegung gemeinsam genutzt werden muss, sollten Sie die Bewegung mit der Berechtigung "Nur ausführen" freigeben, was die Benutzer darauf beschränkt, den Verlauf der Bewegung zu sehen oder Änderungen an der Bewegung vorzunehmen. Auf diese Weise können die Benutzer auch festlegen, ob der Ablauf die Verbindungen des Benutzers, der den Ablauf aufruft, oder die eingebauten Verbindungen des Benutzers/Kontos, der den Ablauf erstellt, verwendet. Weitere

Informationen finden Sie [hier](#)



Konfigurieren von Sicherheitsrollen für einen angemessenen Zugriff

Abgesehen davon, dass Benutzer zu Miteigentümern gemacht werden, können bestimmte Benutzer aufgrund ihrer Sicherheitsrollen in Workflow-Tabellen oder ihrer Rolle als Umgebungsadministratoren Zugang zu Abläufen erhalten.

Im Allgemeinen können Benutzer mit Vollzugriff auf Workflow-Tabellen jeden Ablauf bearbeiten und jeden Laufverlauf einsehen. Ebenso haben Umgebungsadministratoren immer vollen Zugriff auf die Bearbeitung von Workflows und die Anzeige aller Daten, die durch sie fließen.

Gemeinsame Nutzung von Flüssen vs. Senden einer Kopie

Verwendung von Auditing in Dataverse und Purview

Durch die Aktivierung von Audits für Workflows und Verbindungsreferenztabellen lässt sich besser nachvollziehen, wann und von wem ein Power Automate-Flow erstellt/geändert/gelöscht oder freigegeben wird (gilt für alle Flows, die innerhalb von Lösungen erstellt werden). Darüber hinaus kann Microsoft Purview genutzt werden, um Cloud-Flow-Vorgänge zu verfolgen und festzustellen, wer sie durchgeführt hat.

Weitere Informationen finden Sie unter [Power Automate-Auditprotokolle anzeigen - Power Platform | Microsoft Learn](#)

Category	Event	Description
Flows	Created flow	The time when a flow is created.
Flows	Edited flow	Any updates made to the flow.
Flows	Deleted flow	When the flow is deleted.
Flow permissions	Edited permissions	Every time a user's permissions to a flow changes, for example, when a user is added as co-owner.
Flow permissions	Deleted permissions	Every time a user's permissions to the flow is removed.
Trials	Started a paid trial	When a user starts a paid trial.
Trials	Renewed a paid trial	When a user renews a paid trial.

Verwendung von Richtlinien für Datenverlust

"Richtlinien zur Verhinderung von Datenverlusten (Data Loss Prevention, DLP) sind wichtig für die Verwaltung und Kontrolle der gemeinsamen Nutzung und Bewegung von Daten in Power Apps, Power Automate und anderen Power Platform-Komponenten. Diese Richtlinien helfen Unternehmen, Datenverluste zu verhindern und die Einhaltung gesetzlicher Vorschriften und interner Richtlinien zu gewährleisten.

Einige der besten Praktiken, die bei der Erstellung von DLP berücksichtigt werden sollten, sind.

- Blockieren/Isolieren Sie nicht-geschäftliche Verbindungen, insbesondere in Standardumgebungen.
- Erwägen Sie die Blockierung von risikoreichen Operationen wie HTTP, HTTP mit AAD und SharePoint URL usw. in Umgebungen mit vielen Herstellern.
- Und schließlich sollten Sie die Endpunktfilterung nutzen, um sicherzustellen, dass nur die erwarteten Endpunkte erreicht werden.

Weitere Informationen finden Sie unter [Verwalten von Richtlinien zum Schutz vor Datenverlust \(DLP\) - Power Platform | Microsoft Learn](#)

CMK (Customer Managed Keys) für Cloud-Flows

Customer Managed Keys (CMK) für Cloud-Flows in Power Automate ist eine Sicherheitsfunktion, die es Unternehmen ermöglicht, ihre eigenen Verschlüsselungsschlüssel zur Verschlüsselung und zum Schutz von Daten innerhalb ihrer Flows zu verwenden. Diese

bietet eine bessere Kontrolle über die Verschlüsselungsschlüssel und erfüllt spezifische Compliance-Anforderungen für Datensicherheit und Datenschutz. Hier finden Sie einen Überblick darüber, wie CMK in Power Automate funktioniert und wie Sie es implementieren:

Was sind Customer Managed Keys (CMK)?

Customer Managed Keys (CMK) ermöglichen es Unternehmen, ihre eigenen Verschlüsselungsschlüssel zu verwalten, in der Regel mit einem Schlüsselverwaltungsdienst wie Azure Key Vault. Durch die Verwendung von CMK können Unternehmen:

- Sie erhalten eine bessere Kontrolle über die zum Schutz ihrer Daten verwendeten Verschlüsselungsschlüssel.
- Erfüllen Sie gesetzliche und behördliche Anforderungen, die eine vom Kunden kontrollierte Verschlüsselung vorschreiben.
- Entziehen Sie den Zugriff auf Daten durch Deaktivieren oder Löschen von Schlüsseln.

So funktioniert CMK in Power Automate

1. **Verschlüsselung im Ruhezustand:** CMK ermöglicht es Ihnen, die von Power Automate gespeicherten Daten im Ruhezustand mit Ihren eigenen Schlüsseln zu verschlüsseln.
2. **Azure Key Vault:** Die Schlüssel werden in der Regel in Azure Key Vault gespeichert und verwaltet, das sichere Schlüsselverwaltungsfunktionen bietet.
3. **Schlüsselrotation und -verwaltung:** Sie haben die Möglichkeit, die Schlüssel zu rotieren, zu widerrufen und zu prüfen, was eine bessere Kontrolle über den Lebenszyklus der Verschlüsselung ermöglicht.

Weitere Informationen finden Sie in folgendem [learn-Dokument](#) Weitere Details zur [Verwaltung von CMK](#)

Bewährte ALM-Verfahren für Power Automate

ALM (Application Lifecycle Management) ermöglicht es Unternehmen, Artefakte innerhalb einer Lösung sicher über Umgebungen wie Entwicklung, Test und Produktion hinweg zu bewegen.

- **Nicht verwaltete Lösungen** sind vollständig editierbar und ermöglichen es Entwicklern, die Komponenten innerhalb einer Lösung zu ändern und anzupassen.
- **Verwaltete** Lösungen hingegen sind versiegelt und erlauben keine Änderungen, sobald sie in den Zielumgebungen bereitgestellt wurden.

Bei der Verlagerung von Lösungen von einer Umgebung in eine andere sollten einige der wichtigsten Bereitstellungspraktiken beachtet werden:

Bewährte Verfahren für Lösungen

- Lösungen können als logische Grenze verwendet werden, um verwandte Abläufe zusammenzuhalten. Ziehen Sie in Betracht, übergeordnete und untergeordnete Ströme in einer Lösung zusammenzuhalten, die möglicherweise dieselbe Verbindungsreferenz verwenden müssen.
- Umgebungsvariablen in PowerApps ermöglichen es Ihnen, Konfigurationen in verschiedenen

Umgebungen (z. B. Entwicklung, Staging, Produktion) dynamisch zu ändern, ohne den Code der Anwendung zu modifizieren. Diese Variablen dienen als Platzhalter für Daten, die extern festgelegt oder aktualisiert werden können, was die Verwaltung und Bereitstellung der Anwendung erleichtert.

- Azure Key Vault, ein sicherer Speicherdienst, ergänzt dies, indem er sensible Informationen wie Geheimnisse, Schlüssel und Zertifikate sicher speichert. Die Integration von Azure Key Vault mit Power Automate gewährleistet, dass sensible Daten sicher gespeichert und abgerufen werden. Durch diese Integration kann Power Automate direkt auf die Geheimnisse von Azure Key Vault zugreifen, wodurch das Risiko einer Aufdeckung verringert und die allgemeine Sicherheitslage der Anwendung verbessert wird.

Bewährte Praktiken für den Einsatz

- Um ein sicheres ALM zu gewährleisten, sollten Sie sicherstellen, dass keine nicht verwalteten Anpassungen in den Test- und Produktionsumgebungen zugelassen werden, indem Sie nur verwaltete Lösungen einsetzen.
- Alle geschäftskritischen Abläufe sollten nur in einer Entwicklungsumgebung bearbeitet und dann in die Test- und dann in die Produktionsumgebung verschoben werden. Stellen Sie in den Produktionsumgebungen sicher, dass die Abläufe im Besitz von Service Principals sind.

Darüber hinaus sollten alle Änderungen an Produktionsumgebungen stets die ALM-Prozesse durchlaufen.

Pipelines für die Bereitstellung verwenden

Pipelines bieten eine einfache Möglichkeit, Lösungen von der Entwicklungs- über die Entwicklungs- bis hin zur Produktivumgebung zu verschieben. Pipelines bieten auch die Möglichkeit, Lösungen über Service Principals bereitzustellen und ermöglichen ein Rollback von Lösungen. Darüber hinaus können Pipelines für Genehmigungen, für Github usw. erweitert werden.

Verstehen, wann Flows mit Power Apps integriert werden sollten

Power Platform verfügt über eine Reihe von Funktionen, mit denen Entwickler Geschäftslogik für ihre Power Apps erstellen können. Im Folgenden finden Sie die wichtigsten Methoden sowie eine Anleitung, welche Methode Sie verwenden sollten:

Leistungsstarke Anwendungen mit Power Fx

Power Fx ist die funktionale Programmiersprache mit niedrigem Code, die von Excel und Power Platform gemeinsam genutzt wird. In Kombination mit Power Platform-Konnektoren können Entwickler sowohl auf Daten zugreifen als auch benutzerdefinierte Anwendungslogik erstellen. Power Fx unterstützt die folgenden Merkmale:

- **Live** - Canvas Power Apps werden genauso "neu berechnet" wie Excel-Tabellenkalkulationen. Während der Endbenutzer mit der App interagiert, stellt Power Fx im Hintergrund Datenanforderungen und hält den Status der App mit Dataverse auf dem neuesten Stand.
- **Delegation** - Power Fx "delegiert" automatisch, was es kann, an den Server. Funktionen wie Filter(), Lookup() und Search() ermöglichen einer Anwendung den Zugriff auf Daten durch serverseitige Filterung, so dass nur so viele Daten in die Anwendung gebracht werden, wie für das Erlebnis und die Funktionslogik erforderlich sind. Wo eine Delegation nicht möglich ist, werden die Funktionen im lokalen JavaScript-Kontext (d. h. im lokalen Browser) ausgeführt.
- **Optimiert für Dataverse** - Dataverse ist der systemeigene Datenspeicherdienst für Power Apps und als solcher gibt es weniger Schichten, die bei der Verwendung von Power Fx in Verbindung mit Dataverse involviert sind (was niedrige Latenzzeiten bedeutet). Darüber hinaus unterstützt Power Fx die verschiedenen Many-to-One- und One-to-Many-Beziehungen, die in Dataverse verfügbar sind.
- **Offline** - Dataverse-Daten können mit einem mobilen Player für den Einsatz im Feld offline genommen

werden. Änderungen auf beiden Seiten werden synchronisiert, wenn die Verbindung wiederhergestellt wird. Die in der App definierte Geschäftslogik wird auch im Offline-Modus ausgeführt.

Leistung automatisieren

Ein Low-Code-Workflow-Service, der auf dem Power Platform Connector Ecosystem aufbaut. Power Automate bietet bei der Erstellung von Power Apps die folgenden Funktionen:

- **Asynchron** - Ein Power Automate Cloud-Flow ist von Natur aus asynchron. Das heißt, wenn ein Ablauf initiiert wird, nutzt er ein Warteschlangensystem, um die verschiedenen Teilaufgaben zu verwalten. Die asynchrone Natur von Power Automate bedeutet, dass es sich gut für länger laufende komplexe Logiksequenzen eignet.
- **Detaillierte Protokollierung** - Alle Abläufe erstellen eine Aufzeichnung darüber, was bei ihrer Ausführung passiert ist, den so genannten "Laufverlauf". Dies ermöglicht die Rückverfolgbarkeit und stellt sicher, dass es eine Aufzeichnung darüber gibt, was passiert ist und warum.
- **Mehrere Konnektoren** - Obwohl es möglich ist, Power Apps Logik mit mehreren Konnektoren zu erstellen, wird die Leistung aufgrund der Live-Natur von Power Apps mit zunehmender Anzahl von Konnektoren abnehmen. Diese komplexen Szenarien mit mehreren Konnektoren eignen sich hervorragend für die Nutzung von Power Automate, das diese Fälle von der Live-Ausführung in der App auslagern kann.

Dataverse-Plugins mit Power Fx

In C# geschrieben, sind Dataverse-Plugins eine gängige Methode, mit der professionelle Entwickler seit Jahren Geschäftslogik ausdrücken. Als neue Low-Code-Option [kann Power Fx nun zur Erstellung von Plugins verwendet werden](#), die jetzt in der Vorschau zu sehen sind und in einigen Monaten verfügbar sein werden. Es bringt Delegations- und Dataverse-Optimierungen sowie diese Eigenschaften mit sich:

- **In der Transaktion** - Power Fx-Plugins werden in der Dataverse-Transaktion ausgeführt, die die Datenbank aktualisiert. Selbst wenn bei der Aktualisierung einer sekundären Tabelle ein Problem auftritt, werden alle Änderungen rückgängig gemacht, so dass die Datenbank immer in einem konsistenten Zustand ist.
- **Fehlerberichterstattung** - Die Rückmeldung eines Fehlers an den Endbenutzer in einer Anwendung oder einem Ablauf ist so einfach wie der Aufruf der Funktion Error(). Der Endbenutzer sieht den Fehler als Ergebnis seiner Aktion, er kann sofort korrigierend eingreifen und den Vorgang erneut versuchen.
- **Gemeinsamer Drosselpunkt** - Das Plugin ist unumgänglich. Gemeinsame Aktualisierungsgeschäftslogik kann an einer Stelle für alle Anwendungen, Abläufe und andere Power Platform-Endpunkte geschrieben und gepflegt werden.
- **Effizienz** - Plugins können mehrere Connector- und Datenbankaufrufe nacheinander durchführen, ohne den Speicher- und Netzwerk-Overhead jedes einzelnen Schritts, wenn sie in einem Power Automate-Ablauf wären.
- **Erhöhte Sicherheit** - Die Unterstützung von Dataverse für virtuelle Netzwerke ermöglicht es Ihnen, Ihre ausgehenden Verbindungen zu Ressourcen innerhalb Ihres privaten Netzwerks zu schützen. Auf diese Weise können Sie Ihren ausgehenden Datenverkehr von der Power Platform entsprechend Ihrer Netzwerkrichtlinie sicher verwalten. Darüber hinaus können die Daten von den Clients isoliert und geschützt werden, indem der Zugriff in ein serverseitiges Plugin verpackt wird.

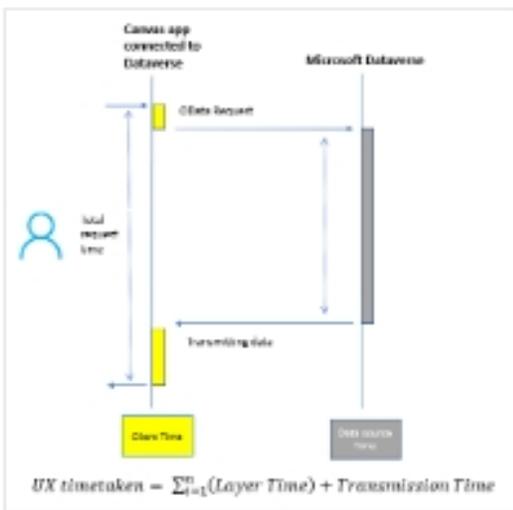
Wann Sie Power Fx im Vergleich zu Power Automate in Power Apps verwenden sollten

Power Fx in der App sollte der Standardmechanismus für die Erstellung der Power App-Geschäftslogik sein. Wie bei jedem Tool gibt es jedoch Sweet Spots und Sollbruchstellen, an denen es sinnvoll ist, andere Tools in der Toolbox zu nutzen.

Anwendungsfälle mit niedriger Latenzzeit

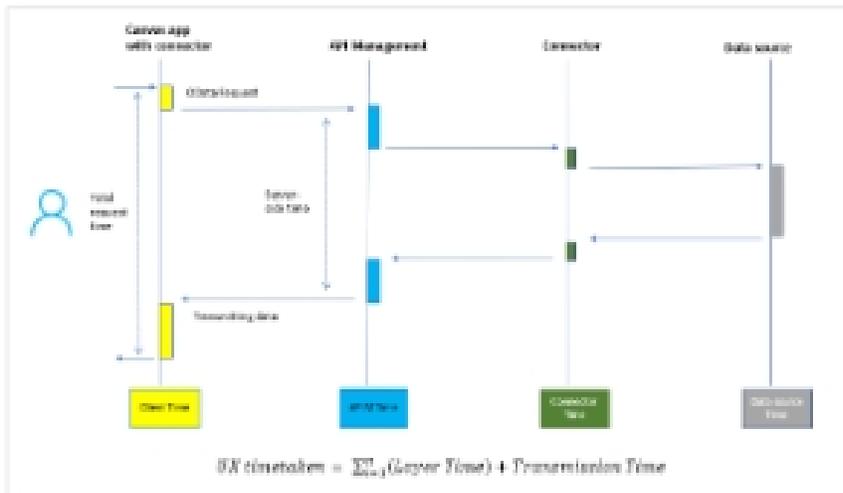
Wenn **eine niedrige Latenzzeit** in einer Power App im Mittelpunkt steht, bedeutet die Live-Natur einer Power Fx-Funktion, dass Sie die **beste Möglichkeit** haben, **Geschäftslogik mit niedriger Latenzzeit über Power Fx**

bereitzustellen. Das Erreichen einer niedrigen Latenzzeit hängt wesentlich von der Komplexität der Aufgabe und der Größe des betreffenden Datensatzes ab. Es ist daher wichtig zu betonen, dass Sie mit Power Fx zwar eine niedrige Latenz erreichen können, dass aber die Art der Aufgabe und die Art des Designs einen erheblichen Einfluss auf die erzielte Leistung haben. Weitere Anleitungen finden Sie [hier](#). Für gängige Datenzugriffsszenarien ist Dataverse in Kombination mit Power Fx der schnellste Ansatz.



Komplexe Sequenzen & Multiverbinder

Für **komplexe Abfolgen** von Aktionen über mehrere Konnektoren hinweg **ist Power Automate ein asynchroner Mechanismus, der die Verarbeitung von der Power App entlastet**. Darüber hinaus ermöglicht die asynchrone Natur von Power Automate der Power App, einen Workflow zu initiieren und das Erlebnis fortzusetzen, ohne auf eine Antwort zu warten.



<hier noch mehr hinzufügen>

Zentralisierte Geschäftslogik

Wenn die Geschäftslogik Änderungen an der Datenbank einschränkt, z. B. die Validierung vor dem Hinzufügen eines Datensatzes, dann **sind Dataverse-Plugins mit Power Fx die beste Lösung**. Keine andere Lösung ist Teil der Dataverse-Transaktion und kann einen Fehler an den Endbenutzer zurückmelden, der die ursprüngliche Änderung initiiert hat, so dass dieser Korrekturen vornehmen und es erneut versuchen kann, während die Datenbank durchgehend konsistent bleibt. Plugins bieten einen einheitlichen Choke-Point, der sicherstellt, dass für alle Anwendungen (Canvas und Model-driven), Flows und andere Endpunkte dieselbe Geschäftslogik durchgesetzt wird.

Ähnlich wie Power Automate bieten auch Dataverse-Plugins mit Power Fx einen zentralen Ort für die Definition benutzerdefinierter

Aktionen, die nicht unbedingt die Datenbank verändern. Dies erleichtert die gemeinsame Nutzung von Power Fx-basierter Geschäftslogik über alle Endpunkte der Power Platform.

Überwachung und Alarmierung

Die regelmäßige Überwachung der Power Automate-Abläufe ist für Unternehmen unerlässlich, um einen reibungslosen und effizienten Betrieb ihrer Arbeitsabläufe zu gewährleisten. Diese Praxis ermöglicht es Unternehmen, aufkommende Probleme oder Engpässe, die den Automatisierungsprozess behindern könnten, sofort zu erkennen und zu beheben. Durch proaktive Überwachung können Unternehmen die notwendigen Maßnahmen zur Verbesserung der Gesamtleistung ergreifen. Darüber hinaus hilft die Überwachung bei der Erkennung und Entschärfung potenzieller Sicherheitsschwachstellen oder Compliance-Probleme und gewährleistet so die sichere Verwaltung sensibler Daten.

Darüber hinaus bietet die Überwachung wertvolle Einblicke in Nutzungsmuster und Leistungsmetriken, die es Unternehmen ermöglichen, ihre Arbeitsabläufe zu optimieren und so die Produktivität und Kosteneffizienz zu steigern.

Automatisierungszentrum

Das Automation Center in Power Automate ist ein zentraler Knotenpunkt für die Verwaltung und Überwachung aller Ihrer Automatisierungsmaßnahmen. Es bietet eine umfassende Schnittstelle zum Erstellen, Überwachen und Optimieren Ihrer automatisierten Arbeitsabläufe in der gesamten Organisation. Diese Funktion trägt dazu bei, Abläufe zu rationalisieren, die Produktivität zu steigern und sicherzustellen, dass alle Automatisierungsaufgaben effizient ausgeführt werden.

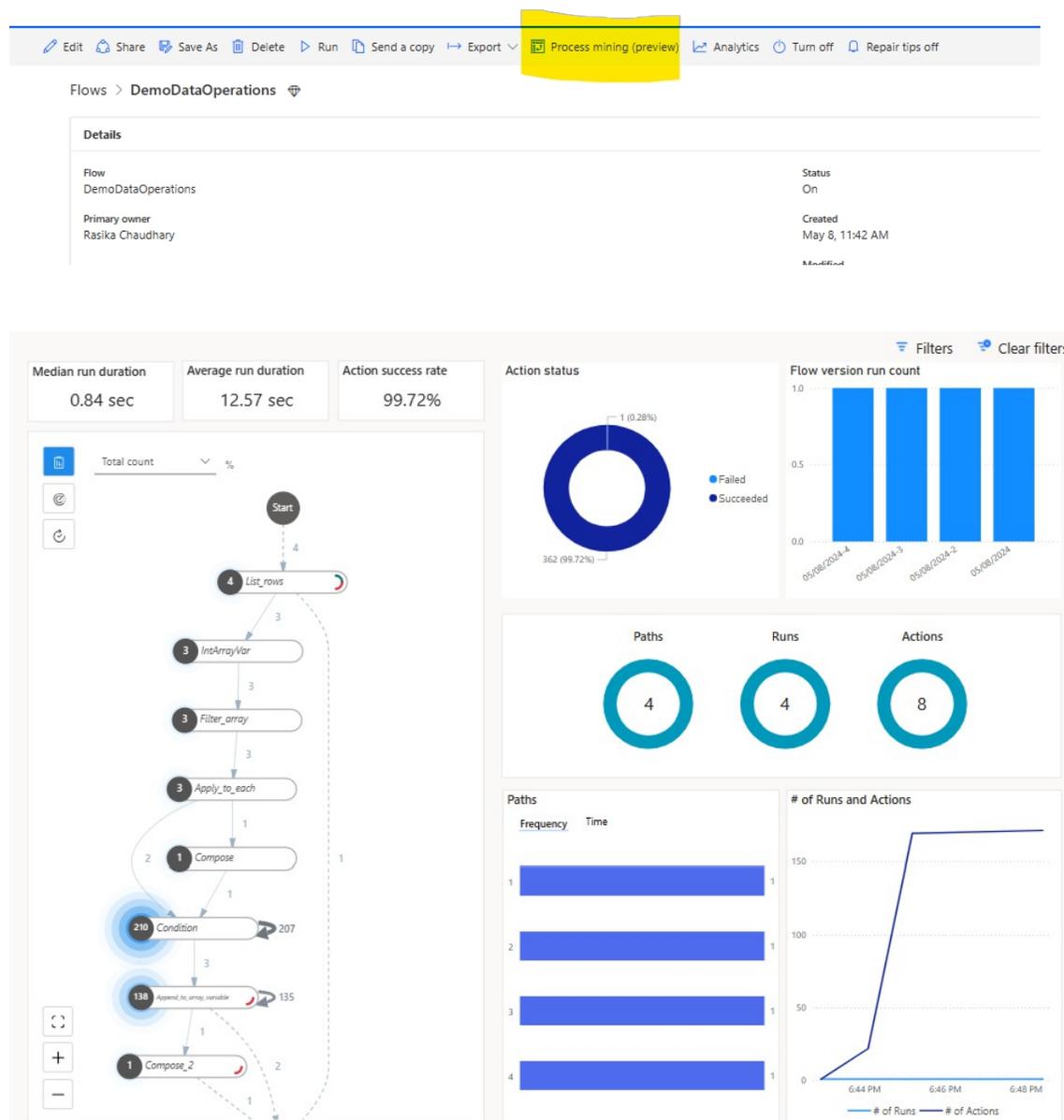
Hier sind einige der Bilder, die das Automation Center zeigen wird:

Visual	Description
Recommendations	List of automation health, compliance, best practice insights, and actionable recommendations.
Top-level flows	Number of top-level flows that had one or more runs based on selected filters. Gives an overall automation health indication and helps identify which top-level runs are failing the most.
Average runs per day	Average number of flow runs per day based on selected filters.
Average run duration	Average flow run duration based on selected filters.
Flow runs error rate	Percentage of errors that occurred during flow execution based on selected filters.
Flow runs by status	Overview of top-level runs status, can be used as to correlate with other factors such as triggering type, run modes, or machines.
Flow runs error trends	Tracks usage and reliability trends of top-level runs over time.
Flow runs by trigger type	Shows top-level flow runs by trigger type.
Top flow runs	Quickly identify critical and regularly failing automations, in order to improve health, resiliency, and exception handling.
Top error codes	Identifies most common errors during flow runs.
Top cloud flows with failed desktop flow runs	Shows which cloud flows are causing the most desktop flow failures and might need to be modified to reduce desktop flow failures.

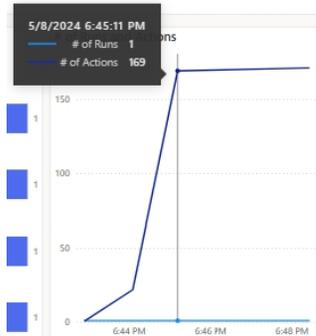
Damit Automation Center funktioniert, muss der Verlauf der Durchläufe in Dataverse gespeichert werden. Weitere Details zu Automation Center finden Sie [hier](#).

Leverage Process Mining (Vorschau)

Mit der Process-Mining-Funktion können Sie nützliche Erkenntnisse gewinnen und Ihre Cloud-Flows verbessern. Die Benutzer können sehen, wie Ihre Abläufe ablaufen, herausfinden, wo sie langsamer werden oder verbessert werden können, und alle Leistungsänderungen verfolgen. Wenn Sie die Process Mining-Funktion nutzen, um den Verlauf Ihrer Abläufe mit Process Mining-Methoden zu untersuchen, können Sie diese Erkenntnisse direkt von der Seite mit den Ablaufdetails aus gewinnen. Weitere Informationen zu den Voraussetzungen und bekannten Einschränkungen finden Sie [hier](#)



Der Abschnitt "Läufe und Aktionen" enthält auch Informationen über die Gesamtzahl der Aktionen, die für einen Flusslauf erforderlich waren. Dies ist wichtig, um zu verstehen, ob Ihr Fluss möglicherweise an die Drosselungsgrenzen für tägliche Aktionsbursts stößt.



Power Automate Analysen

Power Automate Analytics bezieht sich auf eine Reihe von Tools und Funktionen zur Analyse und Gewinnung von Erkenntnissen über die Leistung, Nutzung und Effektivität Ihrer Power Automate-Workflows.

Power Automate-Analyseberichte sind sowohl auf Tenant-Ebene als auch auf Ebene der einzelnen Cloud-Flows verfügbar.

Cloud-Flow-Analytik

Power Automate bietet integrierte Analysefunktionen, mit denen Sie Ablaufläufe überwachen, den Verlauf der Ausführung verfolgen und die Ablaufleistung analysieren können. Nutzen Sie Analyse-Dashboards und -Berichte, um Einblicke in die Ausführungsmuster von Abläufen zu erhalten und Engpässe zu erkennen.

Microsoft Cloud Flow Analytics ist ein von Microsoft angebotener Dienst, der Einblicke und Analysen zur Nutzung und Leistung von Microsoft Power Automate bietet. Sie können auf Cloud Flow Analytics zugreifen über die Menüschaltfläche "Analytics" auf jeder Microsoft Flow-Detailseite.

Darunter werden folgende Details angezeigt:

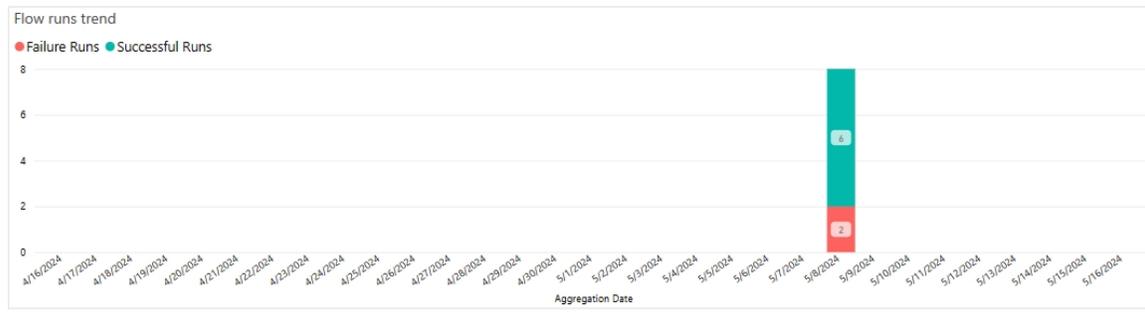
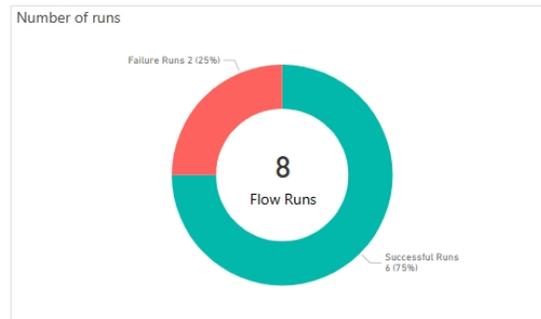
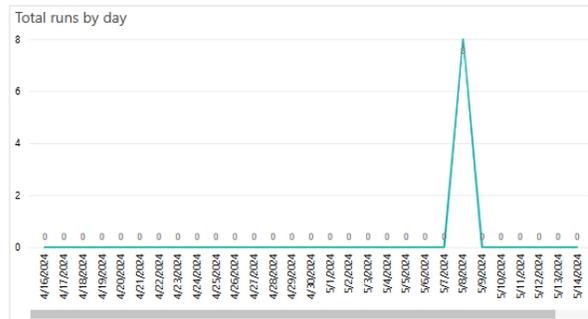
- Flow-Nutzung (einschließlich Anzahl der Aktionsanfragen aus den Cloud-Flow-Läufen)
- Detaillierte Flussfehleranalyse (Art des Fehlers, Anzahl und letztes Auftreten)
- Vorgeschichte von 30 Tagen

Flows > DemoDataOperations

Details	Status	Connection Reference
Flow DemoDataOperations	On	Microsoft
	Created	

Actions **Usage** Errors

Select range Last 30 days



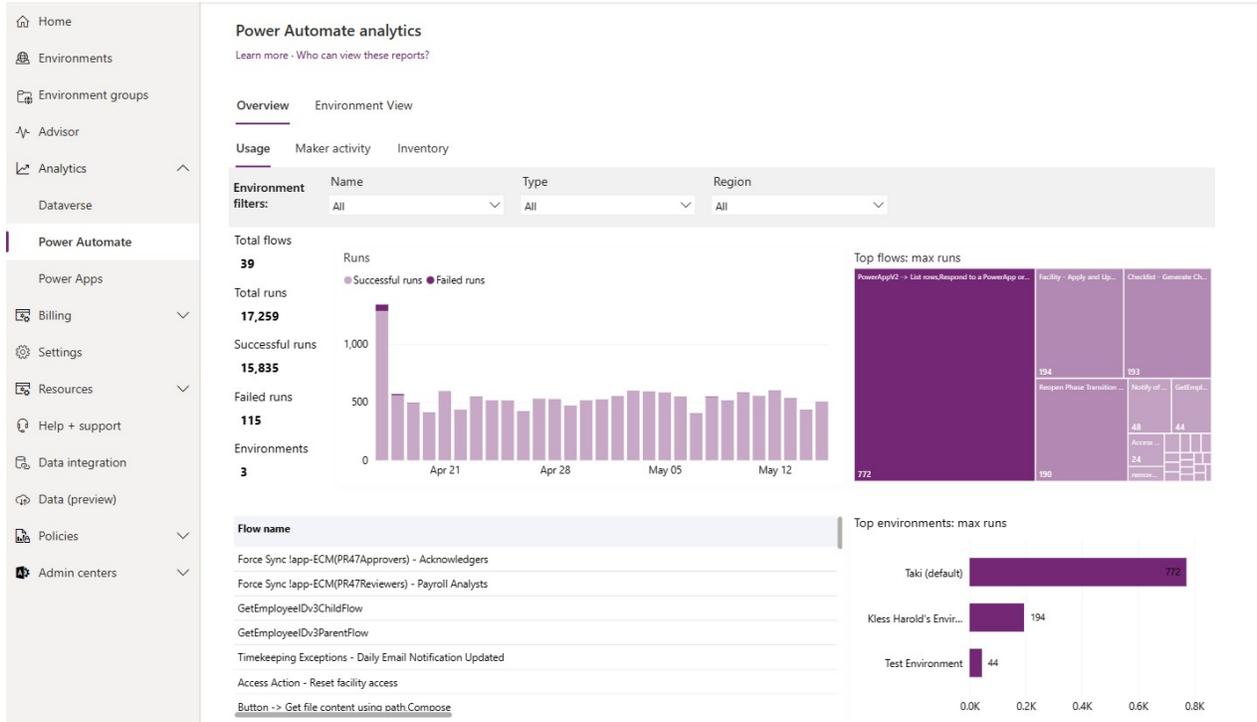
Power Automate Admin-Analytik

Umgebungsadministratoren können über das Microsoft [Power Platform Admin Center](#) auf Analysen für Power Automate zugreifen. Die Berichte bieten Einblicke in Läufe, Nutzung, Fehler, Arten von erstellten Abläufen, gemeinsam genutzte Abläufe und Details zu Konnektoren, die mit den verschiedenen Ablauftypen wie automatisierten Abläufen, Tastenabläufen, geplanten Abläufen, Genehmigungsabläufen und Geschäftsprozessabläufen verbunden sind.

Deckt folgende Details ab:

- Berichte über die Nutzung des Stroms auf der Grundlage von Umgebungen, Umgebungstyp und Region
- Flussinventar, einschließlich Flussstatus
- Maker-Aktivität (aber nur die Benutzer-ID)
- Anzahl der fehlgeschlagenen Läufe (keine Information, welcher Fluss fehlgeschlagen ist)
- 28 Tage Vorgeschichte

Weitere Einzelheiten zu Admin-Analysen finden Sie [hier](#)



Power Automate Management Anschlüsse

Der Power Automate-Dienst bietet auch Möglichkeiten, einige der Verwaltungsaktivitäten für Flows über 2 Konnektoren zu automatisieren

- Power Automate Management-Anschluss
- Power Automate für Administratoren

Power Automate Management-Anschluss

Dieser Konnektor bietet Aktionen für die Verwaltung von Abläufen, wie z. B. das Erstellen, Aktualisieren und Löschen von Abläufen, sowie das Abrufen von Ablaufmetadaten und Laufprotokollen. Er ermöglicht Ihnen die programmatische Interaktion mit Abläufen innerhalb Ihrer Power Automate-Umgebung. Admins können einige dieser Aktionen ausführen, die mit dem Zusatz "As Admin" versehen sind. Weitere Details zu diesem Connector finden Sie [hier](#)



Power Automate Management

Power Automate Management connector enables interaction with Power Automate Management service. For example: creating, editing, and... [Read more](#)

Cancel Flow Run	(i)
Create Connection	(i)
Create Flow	(i)
Delete Flow	(i)
Get Connector	(i)
Get Flow	(i)
Get Flow as Admin	(i)
List Callback URL	(i)
List Connectors	(i)
List Flow Owners	(i)
List Flow Run-Only Users	(i)
List Flows as Admin (V2)	(i)
List My Connections	(i)
List My Environments	(i)
List My Flows	(i)
Modify Flow Owners	(i)
Modify Flow Owners as Admin	(i)
Modify Run-Only Users	(i)
Restore Deleted Flow as Admin	(i)

Power Automate für Administratoren

Der Power Automate for Admins-Konnektor hingegen wird hauptsächlich von Administratoren verwendet, die damit die Möglichkeit haben, Abläufe zu aktivieren/deaktivieren, den Eigentümer des Ablaufs zu ändern usw. Weitere Einzelheiten zu diesem Konnektor finden Sie [hier](#)



Power Automate for Admins
Power Automate Management Connector for Administrators

Disable Flow as Admin	ⓘ
Edit Flow Owner Role as Admin	ⓘ
Enable Flow as Admin	ⓘ
Get Flow Owner Role as Admin	ⓘ
Get Flow User Details	ⓘ
Remove Flow Approval	ⓘ
Remove Flow User Details	ⓘ
Remove Flow as Admin	ⓘ

Power Automate App Insights

Power Automate App Insights ist eine der jüngsten Ergänzungen der Telemetriedaten, auf die Benutzer zugreifen können. Kunden können nun Azure App Insights nutzen, um ihre Workflows, die auf Power Automate laufen, zu diagnostizieren und zu überwachen. Darüber hinaus können Benutzer diese Telemetriedaten nutzen, um Dashboards, sofort einsatzbereite und benutzerdefinierte Warnmeldungen, Leistungsdiagnosen und benutzerdefinierte Analysen über Protokollanalysen zu erstellen.

Für Power Automate können die Ablaufläufe, Auslöser und Aktionsdaten mit App Insights auf Umgebungsebene verknüpft werden. Allerdings können sich Daten aus mehreren Umgebungen bei derselben App Insights-Ressource anmelden.

Die Telemetriedaten der Cloud-Flow-Läufe werden in der Tabelle der **Anfragen** gespeichert, die Daten der Auslöser- und Aktionsebene in der Tabelle der **Abhängigkeiten**.

Weitere Einzelheiten finden Sie in diesem [Artikel](#)

Um Daten an App Insights zu übertragen, wählen Sie Datenexport unter Analytics und konfigurieren Sie wie folgt:

New data exports to Azure Application Insights

Export to Azure Application Insights

Name the export package *

Test connection

Create an Azure Application Insights instance to begin exporting analytics data. [Go to the Azure Portal.](#)

Dataverse diagnostics and performances
Data about APIs, form load diagnostics and performance.

Power Automate
Data about Cloud flow runs, triggers and actions

- Cloud flow runs
- Cloud flow triggers
- Cloud flow actions

Dynamics Customer Service

Um die Ausführung von Cloud-Flows zu überwachen, sehen Sie sich den Abschnitt Metriken an und wenden Sie Folgendes an:

Search < + New chart Refresh Diagnose Share Feedback

Monitoring

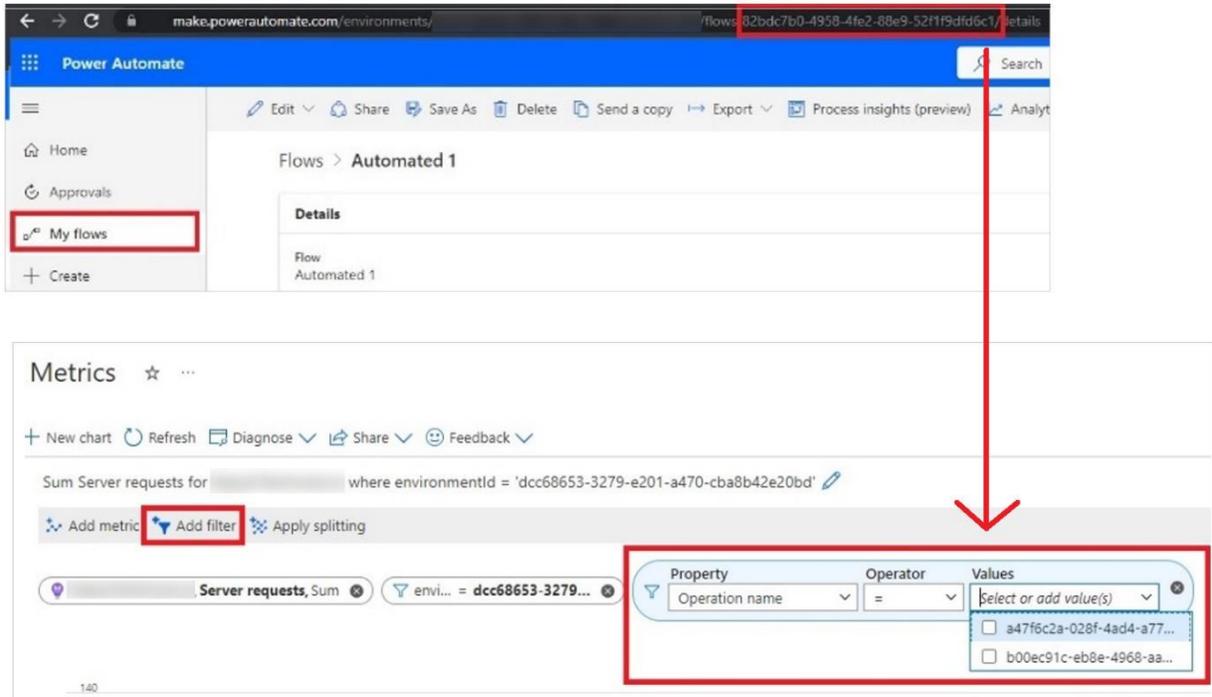
- Alerts
- Metrics**
- Diagnostic settings
- Diagnose and solve problems

Sum Server requests for

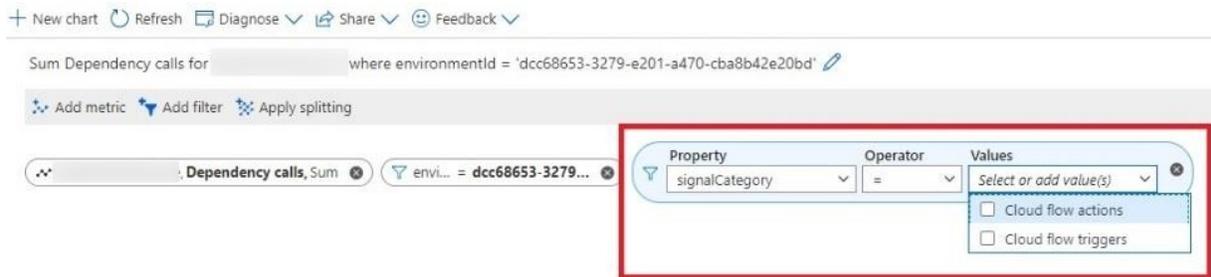
Add metric Add filter Apply splitting

Scope	Metric Namespace	Metric	Aggregation
	Log-based metrics	Server requests	Sum

Sie können außerdem Umgebungs- und Durchflussmengenfilter hinzufügen



Für die Überwachung von Auslösern und Aktionen filtern Sie unter Metriken nach Umgebung und fügen dann die Signalkategorie für Eigenschaftsfilter hinzu und wählen Cloud-Flow-Auslöser und Cloud-Flow-Aktionen



Verlauf des Flusslaufs in Dataverse

Diese Vorschaufunktion (ab Juni 2024) wird es ermöglichen, die Details der Flow-Runs in Dataverse in der FlowRun-Tabelle zu speichern. Dies wird das Automatisierungszentrum in die Lage versetzen, aussagekräftige Daten über die Ausführung von Cloud-Flows zu generieren.

Standardmäßig werden die Ausführungsdaten für einen Zeitraum von 28 Tagen gespeichert, aber der Standardzeitraum kann über das Verwaltungszentrum geändert werden.

Hier finden Sie einige zusätzliche Informationen zu dieser Funktion:

[Bekannte Einschränkungen für die Aufbewahrung der Historie von Datenflussläufen im Dataverse FAQ](#)

