Benutzerhandbuch

AWS App Studio

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS App Studio: Benutzerhandbuch

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS App Studio? ... 1
Verwenden Sie App Studio zum ersten Mal? ... 1
Konzepte ... 2
Rolle des Administrators ... 2
Anwendung (App) ... 3
Automatisierung ... 3
Automatisierungsaktionen ... 3
Rolle „Builder“ ... 3
Komponente ... 3
Konnektor ... 4
Entwicklungsumgebung ... 4
Entität ... 4
Instance ... 4
Seite ... 5
Auslöser ... 5
Wie funktioniert App Studio ... 6
Verbinden Sie Ihre Anwendung mit anderen Diensten ... 7
Konfiguration des Datenmodells Ihrer Anwendung ... 8
Die Benutzeroberfläche Ihrer Anwendung erstellen ... 9
Implementierung der Logik oder des Verhaltens Ihrer Anwendung ... 11
Der Entwicklungszyklus Ihrer Anwendung ... 13
Weitere Informationen ... 14
App Studio einrichten und sich bei App Studio anmelden ... 15
Erstmaliges Erstellen und Einrichten einer App Studio-Instanz ... 15
Eröffnen Sie ein AWS Konto ... 15
Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen ... 16
Erstellen Sie eine App Studio-Instanz im AWS Management Console ... 16
Eine Einladung zur Teilnahme an App Studio annehmen ... 22
Erste Schritte ... 23
Tutorial: Generieren Sie eine App mithilfe von KI ... 23
Voraussetzungen ... 24
Schritt 1: Erstellen einer -Anwendung ... 25
Schritt 2: Erkunden Sie Ihre neue Anwendung ... 25
Schritt 3: Zeigen Sie eine Vorschau Ihrer Anwendung an ... 28

Nächste Schritte ... 28
Tutorial: Beginne mit dem Erstellen von einer leeren App ... 29
Voraussetzungen ... 32
Schritt 1: Erstellen einer -Anwendung ... 32
Schritt 2: Erstellen Sie eine Entität, um die Daten Ihrer App zu definieren ... 33
Schritt 3: Entwerfen Sie die Benutzeroberfläche (UI) und die Logik ... 36
Schritt 4: Zeigen Sie eine Vorschau der Anwendung an ... 39
Schritt 5: Veröffentlichen Sie die Anwendung in der Testumgebung ... 40
Nächste Schritte ... 40
Dokumentation für Administratoren ... 42
Verwaltung des Benutzerzugriffs mit Gruppen und Rollen ... 42
Rollen und Berechtigungen ... 42
Gruppen anzeigen ... 43
Benutzer oder Gruppen hinzufügen ... 43
Die Rolle einer Gruppe ändern ... 45
Benutzer oder Gruppen entfernen ... 46
Stellen Sie über Konnektoren eine Connect zu anderen Diensten her ... 46
Connect zu AWS Diensten her ... 47
Connect zu Diensten von Drittanbietern her ... 95
Konnektoren anzeigen, bearbeiten und löschen ... 104
Löschen einer App Studio-Instanz ... 105
Builder-Dokumentation ... 107
Tutorials ... 107
Erstellen Sie eine Textzusammenfassungs-App mit Amazon Bedrock ... 107
Interaktion mit Amazon S3 ... 116
Aufrufen von Lambda-Funktionen ... 127
Erstellen Sie Ihre App mit generativer KI ... 130
Generieren Sie Ihre App ... 130
Ihre App erstellen oder bearbeiten ... 130
Generieren Sie Ihre Datenmodelle ... 131
Generieren von Beispieldaten ... 131
Aktionen für AWS Dienste konfigurieren ... 131
Spöttische Antworten ... 131
KI beim Bauen um Hilfe bitten ... 132
Anwendungen erstellen, bearbeiten und löschen ... 132
Erstellen einer Anwendung ... 132

Anwendungen importieren ... 134
Anwendungen duplizieren ... 140
Eine Anwendung bearbeiten oder erstellen ... 141
Bearbeiten Sie eine zuvor veröffentlichte App-Version ... 141
Eine Anwendung umbenennen ... 142
Löschen einer Anwendung ... 143
Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen ... 143
Eine Vorschau von Anwendungen anzeigen ... 144
Anwendungen veröffentlichen ... 145
Gemeinsame Nutzung veröffentlichter Anwendungen ... 150
Rollback zu einer zuvor veröffentlichten Version ... 151
Anwendungen exportieren ... 152
Seiten und Komponenten: Erstellen Sie die Benutzeroberfläche Ihrer App ... 153
Seiten verwalten ... 154
Verwaltung von Komponenten ... 156
Konfiguration der rollenbasierten Sichtbarkeit von Seiten ... 158
Seiten in der App-Navigation ordnen und organisieren ... 160
Ändern Sie die Farben in Ihrer App mit App-Themen ... 161
Referenz zu den Komponenten ... 162
Automatisierungen und Aktionen: Definieren Sie die Geschäftslogik Ihrer App ... 212
Konzepte für Automatisierungen ... 213
Automatisierungen erstellen, bearbeiten und löschen ... 215
Automatisierungsaktionen hinzufügen, bearbeiten und löschen ... 217
Referenz zu Automation-Aktionen ... 218
Entitäten und Datenaktionen: Konfigurieren Sie das Datenmodell Ihrer App ... 240
Bewährte Methoden beim Entwerfen von Datenmodellen ... 240
Eine Entität erstellen ... 242
Konfiguration einer Entität ... 245
Eine Entität löschen ... 254
Verwaltete Datenentitäten ... 254
Seiten- und Automatisierungsparameter ... 256
Seitenparameter ... 256
Automatisierungsparameter ... 258
Wird JavaScript zum Schreiben von Ausdrücken verwendet ... 264
Basissyntax ... 265
Interpolation ... 265

Verkettung ... 266
Datum und Uhrzeit ... 266
Codeblöcke ... 267
Globale Variablen und Funktionen ... 267
Referenzieren oder Aktualisieren von Werten von UI-Komponenten ... 267
Mit Tabellendaten arbeiten ... 269
Zugreifen auf Automatisierungen ... 270
Datenabhängigkeiten und zeitliche Überlegungen ... 273
Beispiel: Bestelldetails und Kundeninformationen ... 274
Bewährte Methoden für Datenabhängigkeit und Timing ... 274
Eine App mit mehreren Benutzern erstellen ... 275
Laden Sie Entwickler ein, eine App zu bearbeiten ... 276
Es wird versucht, eine App zu bearbeiten, die gerade von einem anderen Benutzer bearbeitet wird ... 276
Aktualisierung der Inhaltssicherheitseinstellungen Ihrer App ... 277
Problembehandlung und Debuggen ... 280
Einrichtung, Berechtigungen und Onboarding ... 280
Das App Studio-Setup ist fehlgeschlagen, als die Option Eine Kontoinstanz für mich erstellen ausgewählt wurde ... 280
Nach der Einrichtung kann nicht auf App Studio zugegriffen werden ... 281
Sie sind sich nicht sicher, welchen Benutzernamen oder welches Passwort Sie verwenden sollen, wenn Sie sich bei App Studio anmelden ... 281
Beim Einrichten von App Studio erhalte ich einen Systemfehler ... 281
Ich kann die URL meiner App Studio-Instanz nicht finden ... 282
Ich kann Gruppen oder Rollen in App Studio nicht ändern ... 282
Wie kann ich App Studio verlassen ... 280
Problembehandlung und Debuggen von Apps ... 282
Der AI-Builder-Assistent ... 283
Im App Studio ... 283
Apps in der Vorschau anzeigen ... 284
In der Testumgebung ... 285
Logs in verwenden CloudWatch ... 287
Konnektoren ... 289
Apps veröffentlichen und teilen ... 293
Ich sehe im Dialogfeld „Teilen“ keine neu erstellten App-Rollen ... 293
Ich habe keine E-Mail erhalten, als die Veröffentlichung meiner App abgeschlossen war ... 293

Die Endbenutzer meiner App können nicht auf die veröffentlichte App zugreifen 294
Sicherheit 295
Sicherheitsüberlegungen und Abhilfemaßnahmen 296
Sicherheitsüberlegungen 296
Empfehlungen zur Minderung von Sicherheitsrisiken 297
Datenschutz 297
Datenverschlüsselung 298
Verschlüsselung während der Übertragung 299
Schlüsselverwaltung 299
Datenschutz für den Datenverkehr zwischen Netzwerken 299
App Studio und Identity and Access Management 300
Identitätsbasierte Richtlinien 301
Ressourcenbasierte Richtlinien 302
Richtlinienaktionen 303
Richtlinienressourcen 304
Bedingungsschlüssel für die Richtlinie 304
ACLs 304
ABAC 305
Temporäre Anmeldeinformationen 305
Prinzipalberechtigungen 305
Servicerollen 306
Service-verknüpfte Rollen 306
AWS verwaltete Richtlinien 306
Service-verknüpfte Rollen 311
Beispiele für identitätsbasierte Richtlinien 313
Compliance-Validierung 317
Ausfallsicherheit 319
Sicherheit der Infrastruktur 319
Konfigurations- und Schwachstellenanalyse 320
Serviceübergreifende Confused-Deputy-Prävention 320
Regionsübergreifender Datentransfer 321
Unterstützte Browser 323
Unterstützte und empfohlene Browser für die Erstellung von Anwendungen 323
Unterstützte und empfohlene Browser für Endbenutzer von Anwendungen 323
Aktualisieren Sie die Browsereinstellungen, um Apps in App Studio zu erstellen 324
Kontingente 325

Dokumentverlauf 326
..... cccxl

Was ist AWS App Studio?

AWS App Studio ist ein generativer KI-gestützter Dienst, der natürliche Sprache verwendet, um Sie bei der Erstellung von Unternehmensanwendungen zu unterstützen. App Studio eröffnet Technikern ohne Kenntnisse in der Softwareentwicklung, wie IT-Projektmanagern, Dateningenieuren und Unternehmensarchitekten, die Anwendungsentwicklung. Mit App Studio können Sie schnell sichere und vollständig verwaltete Anwendungen erstellen AWS, ohne dass Sie über betriebliches Fachwissen verfügen müssen.

Entwickler können App Studio verwenden, um Apps zur Modernisierung interner Geschäftsprozesse zu erstellen und bereitzustellen. Einige Anwendungsfälle sind Inventarverwaltung und -verfolgung, Bearbeitung von Ansprüchen und komplexe Genehmigungen zur Verbesserung der Mitarbeiterproduktivität und der Kundenergebnisse.

Themen

- Verwenden Sie App Studio zum ersten Mal?

Verwenden Sie App Studio zum ersten Mal?

Wenn Sie App Studio zum ersten Mal verwenden, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- Für Benutzer mit der Administratorrolle, die App Studio einrichten, Benutzer und Zugriff verwalten und Connectors mit anderen Diensten AWS oder Diensten von Drittanbietern konfigurieren, finden Sie unter AWS App Studio-Konzepte undApp Studio einrichten und sich bei AWS App Studio anmelden.
- Entwickler, die Anwendungen erstellen und entwickeln werden, finden Sie unter AWS App Studio-Konzepte undErste Schritte mit AWS App Studio.

AWS App Studio-Konzepte

Machen Sie sich mit den wichtigsten Konzepten von App Studio vertraut, um die Erstellung von Anwendungen und die Automatisierung von Prozessen für Ihr Team zu beschleunigen. Zu diesen Konzepten gehören Begriffe, die überall in App Studio sowohl für Administratoren als auch für Entwickler verwendet werden.

Themen

- Rolle des Administrators
- Anwendung (App)
- Automatisierung
- Automatisierungsaktionen
- Rolle „Builder“
- Komponente
- Konnektor
- Entwicklungsumgebung
- Entität
- Instance
- Seite
- Auslöser

Rolle des Administrators

Admin ist eine Rolle, die einer Gruppe in App Studio zugewiesen werden kann. Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.

Nur Benutzer mit der Administratorrolle haben Zugriff auf den Admin-Hub, der Tools zur Verwaltung von Rollen, Datenquellen und Anwendungen enthält.

Anwendung (App)

Eine Anwendung (App) ist ein einzelnes Softwareprogramm, das für Endbenutzer entwickelt wurde, um bestimmte Aufgaben zu erledigen. Apps in App Studio umfassen Ressourcen wie UI-Seiten und Komponenten, Automatisierungen und Datenquellen, mit denen Benutzer interagieren können.

Automatisierung

Mit Automatisierungen definieren Sie die Geschäftslogik Ihrer Anwendung. Die Hauptkomponenten einer Automatisierung sind: Auslöser, die die Automatisierung starten, eine Abfolge von einer oder mehreren Aktionen, Eingabeparameter, mit denen Daten an die Automatisierung übergeben werden, und eine Ausgabe.

Automatisierungsaktionen

Eine Automatisierungsaktion, allgemein als Aktion bezeichnet, ist ein einzelner Logikschritt, aus dem eine Automatisierung besteht. Jede Aktion führt eine bestimmte Aufgabe aus, sei es das Senden einer E-Mail, das Erstellen eines Datensatzes, das Aufrufen einer Lambda-Funktion oder das Aufrufen. APIs Aktionen werden zu Automatisierungen aus der Aktionsbibliothek hinzugefügt und können in bedingte Anweisungen oder Schleifen gruppiert werden.

Rolle „Builder“

Builder ist eine Rolle, die einer Gruppe in App Studio zugewiesen werden kann. Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder Anwendungen anderer Builder verwalten.

Benutzer mit der Builder-Rolle haben Zugriff auf den Builder Hub, der Details zu Ressourcen wie den Anwendungen, auf die der Builder Zugriff hat, sowie hilfreiche Informationen wie Lernressourcen enthält.

Komponente

Komponenten sind einzelne funktionale Elemente innerhalb der Benutzeroberfläche Ihrer Anwendung. Komponenten sind in Seiten enthalten, und einige Komponenten können als Container für andere Komponenten dienen. Komponenten kombinieren Benutzeroberflächenelemente mit der Geschäftslogik, die das Benutzeroberflächenelement ausführen soll. Ein Komponententyp ist

beispielsweise ein Formular, in das Benutzer Informationen in Felder eingeben können und diese Informationen nach dem Absenden als Datenbankeintrag hinzugefügt werden.

Konnektor

Ein Connector ist eine Verbindung zwischen App Studio und anderen AWS Diensten wie AWS Lambda Amazon Redshift oder Diensten von Drittanbietern. Sobald ein Connector erstellt und konfiguriert ist, können Builder ihn und die Ressourcen, die er mit App Studio verbindet, in ihren Anwendungen verwenden.

Nur Benutzer mit der Administratorrolle können Connectors erstellen, verwalten oder löschen.

Entwicklungsumgebung

Die Entwicklungsumgebung ist ein visuelles Tool zum Erstellen von Anwendungen. Diese Umgebung umfasst die folgenden Registerkarten zum Erstellen von Apps:

- Seiten: Hier entwerfen Entwickler ihre Anwendungen mit Seiten und Komponenten.
- Automatisierungen: Hier entwerfen Entwickler die Geschäftslogik ihrer Anwendung mithilfe von Automatisierungen.
- Daten: Hier entwerfen Entwickler das Datenmodell ihrer Anwendung mithilfe von Entitäten.

Die Entwicklungsumgebung enthält auch eine Debug-Konsole und ein KI-Chatfenster, um beim Erstellen kontextuelle Hilfe zu erhalten. Entwickler können in der Entwicklungsumgebung eine Vorschau ihrer laufenden Anwendungen anzeigen.

Entität

Entitäten sind Datentabellen in App Studio. Entitäten interagieren direkt mit Tabellen in Datenquellen. Entitäten umfassen Felder zur Beschreibung der darin enthaltenen Daten, Abfragen zum Suchen und Zurückgeben von Daten sowie Zuordnungen, um die Felder der Entität mit den Spalten einer Datenquelle zu verbinden.

Instance

Eine Instanz ist ein logischer Container für all Ihre App Studio-Ressourcen. Sie repräsentiert Sie, Ihr Unternehmen, Ihr Team oder Ihre Organisation und enthält alle Ihre App Studio-Ressourcen

wie Anwendungen, Konnektoren und Rollenzuweisungen für Benutzer und Gruppen. Größere Organisationen oder Unternehmen verfügen in der Regel über mehrere App Studio-Instanzen, z. B. eine Sandbox-, Test- und Produktionsinstanz. Sie erstellen eine Instanz im Rahmen der Einrichtung von App Studio.

Seite

Seiten sind Container für Komponenten, die die Benutzeroberfläche einer Anwendung in App Studio bilden. Jede Seite stellt einen Bildschirm der Benutzeroberfläche (UI) Ihrer Anwendung dar, mit dem Ihre Benutzer interagieren werden. Seiten werden auf der Registerkarte Seiten des Anwendungsstudios erstellt und bearbeitet.

Auslöser

Ein Trigger bestimmt, wann und unter welchen Bedingungen eine Automatisierung ausgeführt wird. Einige Beispiele für Trigger sind `On click` für Schaltflächen und `On select` für Texteingaben. Der Typ der Komponente bestimmt die Liste der verfügbaren Trigger für diese Komponente. Trigger werden den Komponenten hinzugefügt und im Application Studio konfiguriert.

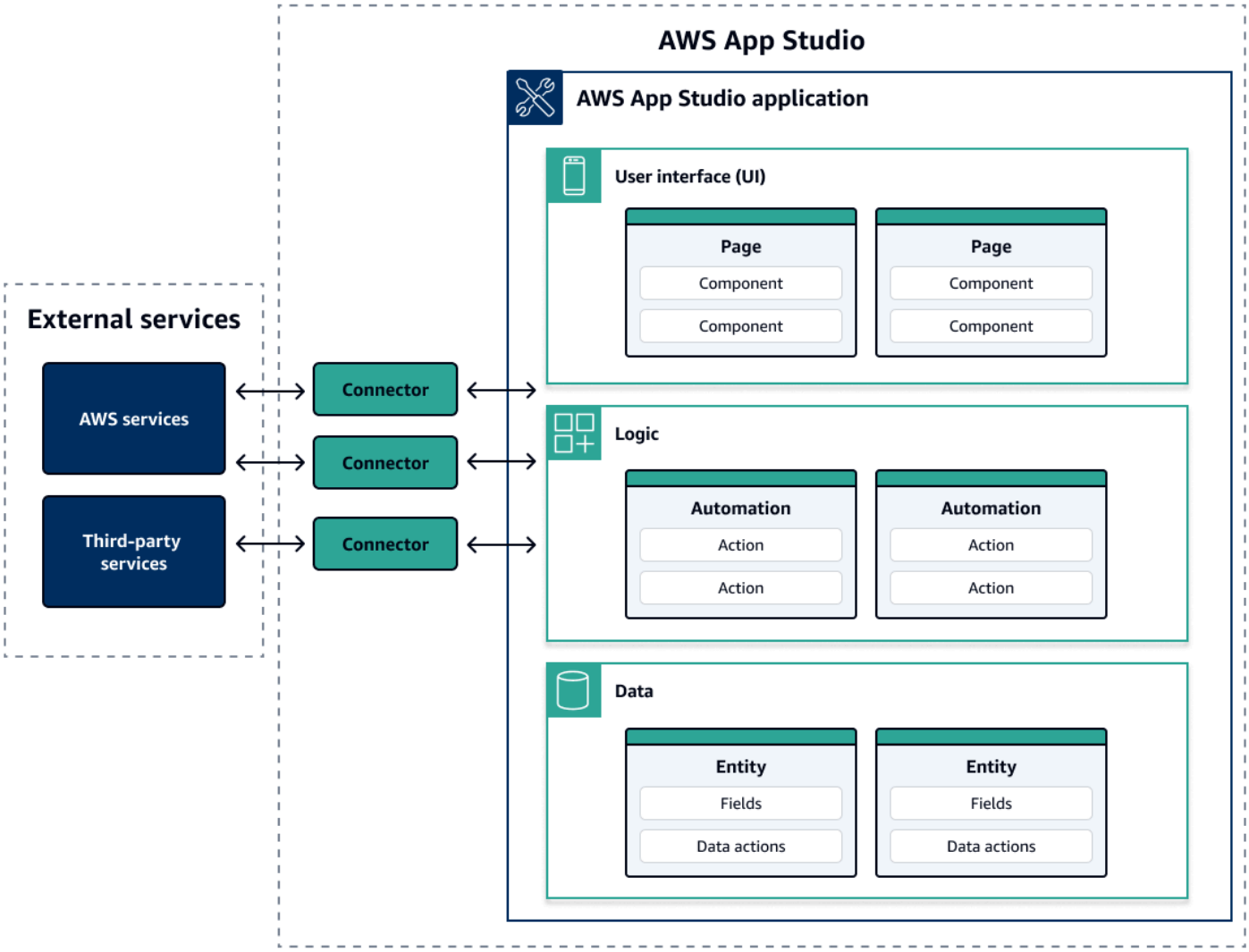
So funktioniert AWS App Studio

Es gibt einige wichtige Konzepte, die Sie verstehen sollten, wenn Sie AWS App Studio zum Erstellen von Anwendungen verwenden. Dieses Thema behandelt die Grundlagen der folgenden Konzepte oder Ressourcen:

- Verwenden von Konnektoren, um eine Verbindung zu anderen Diensten herzustellen und deren Ressourcen oder API-Aufrufe in Ihrer Anwendung zu verwenden. Sie können beispielsweise Konnektoren verwenden, um Daten zu speichern und darauf zuzugreifen oder Benachrichtigungen von Ihrer App aus zu senden.
- Verwenden Sie Entitäten, um das Datenmodell Ihrer Anwendung zu konfigurieren, das Ihre Anwendung mit Ihrer externen Datenquelle verbindet.
- Verwenden von Seiten und Komponenten zum Erstellen der Benutzeroberfläche (UI) Ihrer Anwendung.
- Verwenden Sie Automatisierungen und Aktionen, um die Logik oder das Verhalten Ihrer Anwendung zu implementieren.
- Der Lebenszyklus der Anwendungsentwicklung in App Studio: Erstellen, Testen und Veröffentlichen.

Weitere Informationen zu App Studio-Konzepten finden Sie unterAWS App Studio-Konzepte.

Die folgende Abbildung zeigt ein einfaches Diagramm, wie App Studio und seine Ressourcen organisiert sind.



Innerhalb einer App in App Studio interagieren Seiten, Automatisierungen und Entitäten miteinander. Sie verwenden Konnektoren, um diese Ressourcen mit externen Diensten wie Daten-, Speicher- oder Benachrichtigungsanbietern zu verbinden. Um erfolgreich eine App zu entwickeln, ist es wichtig zu verstehen, wie all diese Konzepte und Ressourcen miteinander interagieren.

Verbinden Sie Ihre Anwendung mit anderen Diensten

Einer der größten Vorteile der Verwendung von App Studio zur Erstellung von Anwendungen ist die einfache Integration Ihrer App in andere Dienste. In App Studio stellen Sie eine Verbindung zu anderen Diensten her, indem Sie Konnektoren verwenden, die für den Dienst und die Ressourcen oder API-Aufrufe, die Sie mit Ihrer Anwendung verwenden möchten, spezifisch sind.

Sie erstellen Connectors auf App Studio-Instanzebene und nicht in einzelnen Apps. Nachdem Sie Konnektoren erstellt haben, können Sie sie je nach verbundenem Dienst und Anwendung in verschiedenen Teilen von Anwendungen verwenden.

Im Folgenden finden Sie Beispiele für Funktionen in Anwendungen, die Konnektoren verwenden, um eine Verbindung zu anderen Diensten herzustellen:

- Der häufigste Anwendungsfall, der in fast allen Anwendungen verwendet wird, ist das Speichern und Zugreifen auf Daten, die in der Anwendung verwendet werden, indem eine Verbindung zu AWS Datendiensten wie Amazon Redshift, Amazon DynamoDB oder Amazon Aurora hergestellt wird.
- Eine Anwendung, die das Hochladen und Anzeigen von Bildern, z. B. Quittungen, ermöglicht, kann Amazon S3 verwenden, um die Bilddateien zu speichern und darauf zuzugreifen.
- Eine Textzusammenfassungs-App kann eine Texteingabe an Amazon Bedrock senden und die zurückgesendete Zusammenfassung anzeigen.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können. Beim Erstellen von Konnektoren müssen Sie die richtigen Anmeldeinformationen und Informationen zu den Ressourcen oder API-Aufrufen angeben, die Sie verwenden möchten.

Konfiguration des Datenmodells Ihrer Anwendung

Die Daten Ihrer Anwendung sind die Informationen, die der Anwendung zugrunde liegen. In App Studio erstellen und verwenden Sie Entitäten, die die verschiedenen Datentypen repräsentieren, die Sie speichern und mit denen Sie arbeiten. In einer Tracking-Anwendung für Kundenbesprechungen könnten Sie beispielsweise drei Entitäten haben, die Kundengespräche, die Tagesordnungen und die Teilnehmer repräsentieren.

Entitäten enthalten Felder mit Typen wie Ganzzahl oder Zeichenfolge, die die gespeicherten Daten beschreiben. Obwohl Sie Entitäten verwenden, um Ihr Datenmodell zu definieren, müssen Sie Ihre Entität mit einem externen Datenspeicherdienst wie Amazon Redshift oder Amazon DynamoDB verbinden, um die Daten zu speichern. Sie können sich eine Entität als Vermittler zwischen Ihrer App Studio-Anwendung und den Daten im externen Service vorstellen.

Sie können Datenaktionen verwenden, um mit den Daten in Ihrer Anwendung aus Komponenten und Automatisierungen zu interagieren. Die beiden am häufigsten zu verwendenden Datenaktionen sind eine `getAll` Aktion und eine `getByID` Aktion. Ihre Anwendung könnte beispielsweise die `getAll` Datenaktion verwenden, um eine Tabelle mit Ihren Daten zu füllen, und eine `getByID` Aktion, um eine Detailkomponente mit weiteren Informationen zu einer bestimmten Dateneingabe zu füllen.

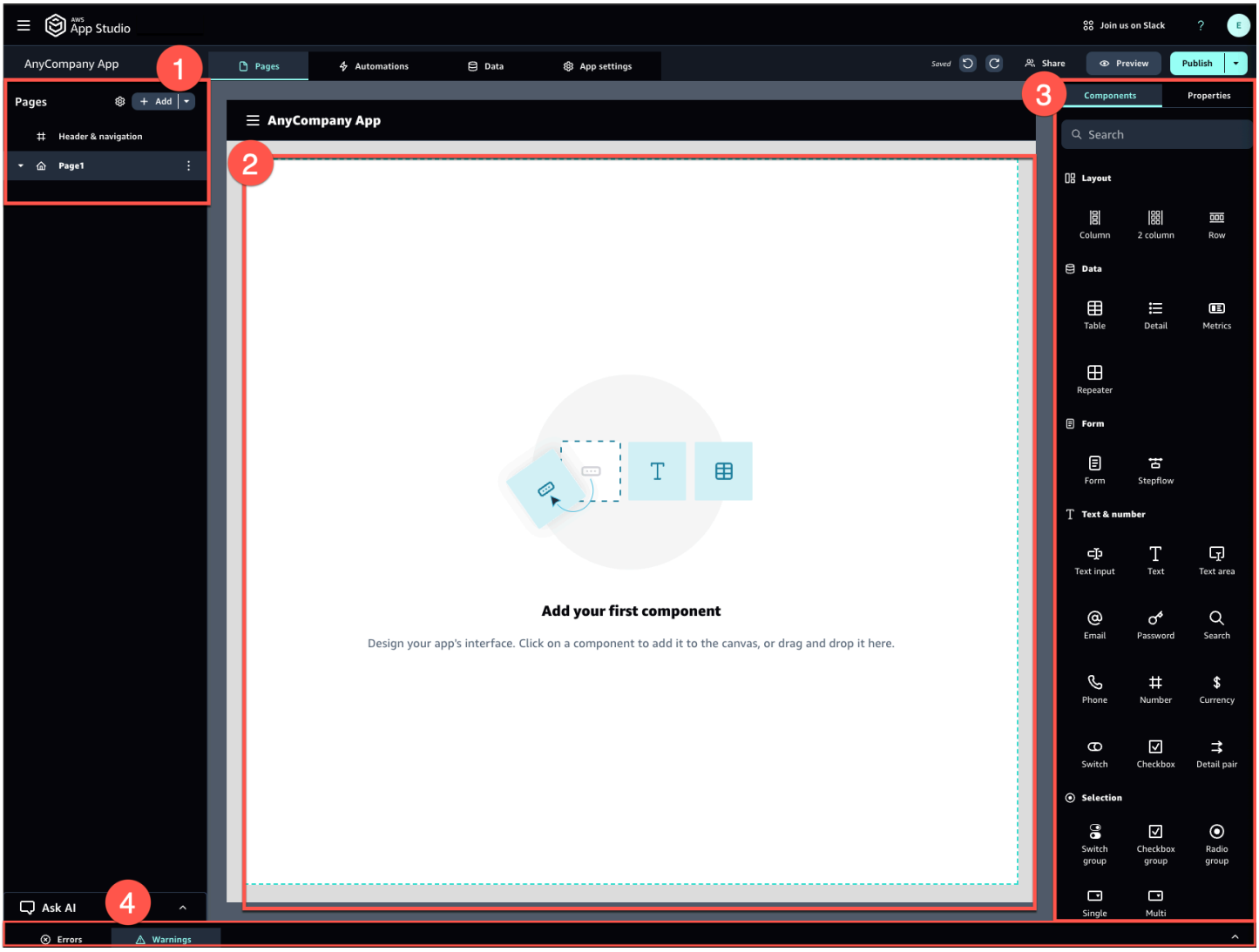
Sie können Ihrer Entität auch Beispieldaten hinzufügen, um Ihre Anwendung einfacher zu testen, ohne externe Dienste aufrufen zu müssen.

Die Benutzeroberfläche Ihrer Anwendung erstellen

In App Studio erstellen Sie die Benutzeroberfläche Ihrer Anwendung mit Seiten und Komponenten. Seiten sind einzelne Bildschirme Ihrer Anwendung und sind Container für Komponenten. Komponenten sind die Bausteine der Benutzeroberfläche Ihrer Anwendung. Es gibt viele Arten von Komponenten, z. B. Tabellen, Formulare, Bildbetrachter und Schaltflächen.

Die folgende Abbildung zeigt die Registerkarte „Seiten“ des Anwendungsstudios, auf der Sie Seiten und Komponenten zu Ihrer Anwendung hinzufügen oder konfigurieren können. Die folgenden Schlüsselbereiche sind hervorgehoben und nummeriert:

1. Das Seiten-Panel auf der linken Seite. Hier verwalten Sie Seiten, die Kopfzeile der Anwendung und die Navigationseinstellungen. Sie können alle Seiten und Komponenten Ihrer Anwendung anzeigen.
2. Die Leinwand, auf der die Komponenten der aktuellen Seite angezeigt werden. Sie können die Komponenten auf der Arbeitsfläche auswählen, um ihre Eigenschaften zu konfigurieren.
3. Das Komponenten - oder Eigenschaftenfenster auf der rechten Seite. Wenn nichts ausgewählt ist, wird das Bedienfeld „Komponenten“ angezeigt, in dem die Liste der Komponenten angezeigt wird, die Ihrer Seite hinzugefügt werden können. Wenn Sie eine Seite oder Komponente auswählen, wird das Eigenschaftenfenster angezeigt, in dem Sie die Seite oder Komponente konfigurieren.
4. Die unteren Bereiche „Fehler“ und „Warnungen“. In diesen Bereichen werden alle Fehler oder Warnungen in Ihrer Anwendung angezeigt, die am häufigsten auf Konfigurationsprobleme zurückzuführen sind. Sie können das Fenster auswählen, um es zu erweitern und die Meldungen zu sehen.



Beispielsweise können Anwendungen, bei denen Benutzer Informationen eingeben müssen, die folgenden Seiten und Komponenten enthalten:

- Eine Eingabeseite, die eine Formularkomponente enthält, die Benutzer zum Ausfüllen und Senden von Informationen verwenden.
- Eine Listenansichtsseite, die eine Tabellenkomponente mit Informationen zu jeder Eingabe enthält.
- Eine Detailansichtsseite, die eine Detailkomponente mit weiteren Informationen zu jeder Eingabe enthält.

Komponenten können statische Informationen oder Daten enthalten, z. B. ein Formular mit definierten Feldern. Sie können mithilfe von Automatisierungen auch dynamische Informationen einbeziehen, z.

B. einen Bildbetrachter, der ein Bild aus einem Amazon S3 S3-Bucket abruft und es dem Benutzer anzeigt.

Es ist wichtig, das Konzept der Seitenparameter zu verstehen. Sie verwenden Seitenparameter, um Informationen von einer Seite zur anderen zu senden. Ein gängiges Beispiel für einen Anwendungsfall für Seitenparameter ist das Suchen und Filtern, bei dem der Suchbegriff von einer Seite an die Tabelle oder Liste mit Elementen gesendet wird, nach denen auf einer anderen Seite gefiltert werden soll. Ein weiteres Anwendungsbeispiel ist das Anzeigen von Artikeldetails, bei dem die Element-ID an eine detaillierte Viewer-Seite gesendet wird.

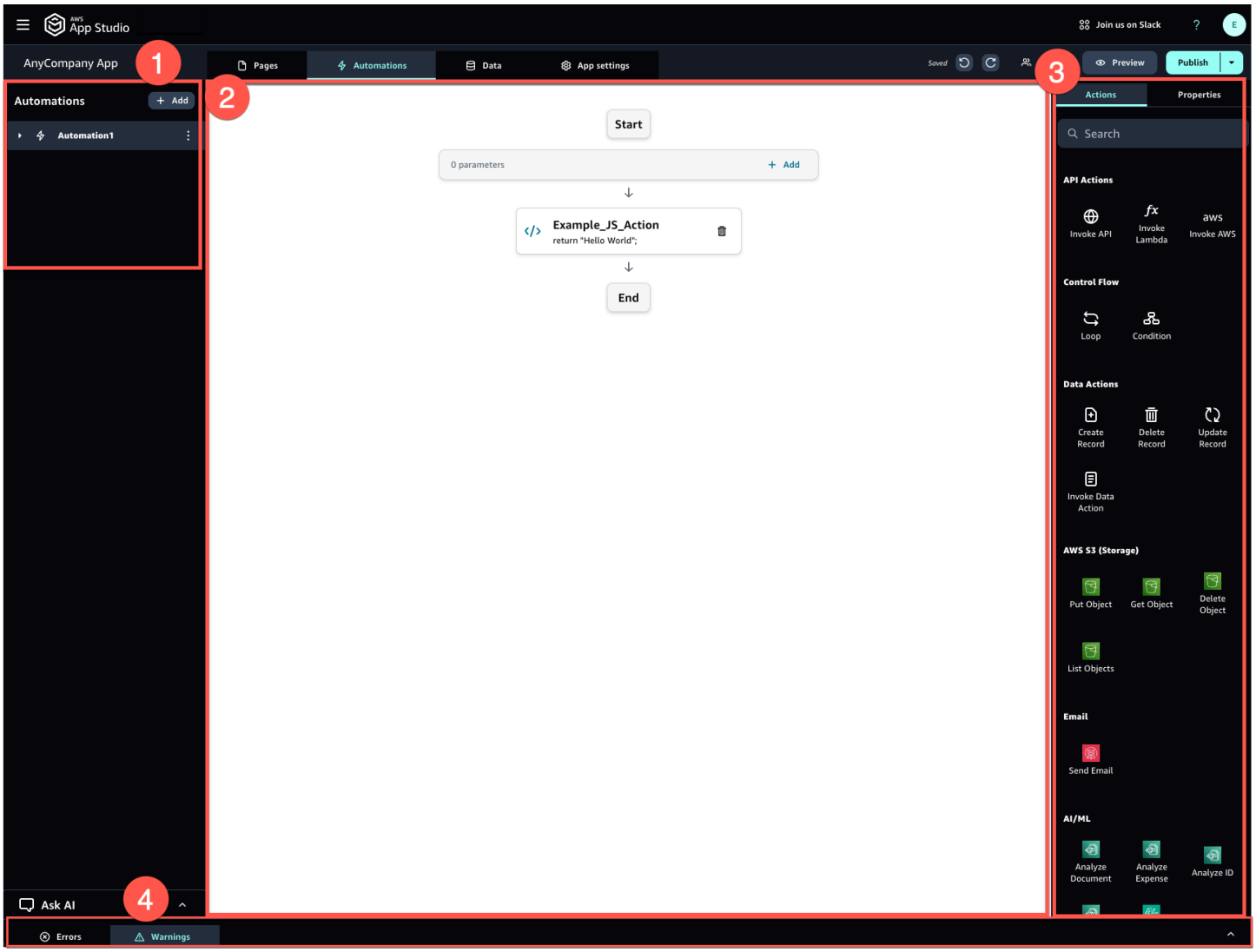
Implementierung der Logik oder des Verhaltens Ihrer Anwendung

Sie können sich die Logik oder das Verhalten Ihrer Anwendung als Funktionalität der Anwendung vorstellen. Sie können definieren, was passiert, wenn ein Benutzer eine Schaltfläche auswählt, Informationen übermittelt, zu einer neuen Seite navigiert oder auf andere Weise interagiert.
In App Studio definieren Sie die Logik Ihrer Anwendung mit Automatisierungen und Aktionen. Automatisierungen sind Container für Aktionen, die die Bausteine der Funktionalität von Automatisierungen bilden.

Die folgende Abbildung zeigt die Registerkarte „Automatisierungen“ des Anwendungsstudios, auf der Sie Automatisierungen und deren Aktionen in Ihrer Anwendung hinzufügen oder konfigurieren können. Die folgenden Schlüsselbereiche sind hervorgehoben und nummeriert:

- Das Automationsfenster auf der linken Seite. Hier verwalten Sie Automatisierungen. Sie können alle Automatisierungen und Aktionen Ihrer Anwendung einsehen.
- Die Leinwand, auf der die aktuelle Automatisierung angezeigt wird. Es zeigt die konfigurierten Automatisierungsparameter (die später in diesem Abschnitt erklärt werden) und Aktionen an. Sie können die Komponenten auf der Arbeitsfläche auswählen, um ihre Eigenschaften zu konfigurieren.
- Die Bereiche Aktionen und Eigenschaften auf der rechten Seite. Wenn nichts ausgewählt ist, wird das Bedienfeld „Aktionen“ angezeigt. Es zeigt die Liste der Aktionen an, die zu Ihrer Automatisierung hinzugefügt werden können. Wenn Sie eine Automatisierung auswählen, können Sie ihre Eigenschaften anzeigen und konfigurieren, z. B. die Eingabe und Ausgabe der Automatisierung. Wenn Sie eine Aktion auswählen, können Sie die Eigenschaften der Aktion anzeigen und konfigurieren.

- Die unteren Fenster mit Fehlern und Warnungen. In diesem Bereich werden alle Fehler oder Warnungen in Ihrer Anwendung angezeigt (am häufigsten aufgrund von Konfigurationsproblemen). Sie können das Fenster auswählen, um es zu erweitern und die Meldungen zu sehen.



Automatisierungen können einfach (wie das Hinzufügen von Zahlen und das Zurückgeben des Ergebnisses) oder leistungsfähiger sein (z. B. das Senden einer Eingabe an einen anderen Dienst und die Rückgabe des Ergebnisses). Die Hauptkomponenten einer Automatisierung sind wie folgt:

- Ein Trigger, der definiert, wann die Automatisierung ausgeführt wird. Ein Beispiel ist, wenn der Benutzer eine Taste in der Benutzeroberfläche drückt.
- Eine Automatisierungseingabe, die Informationen an die Automatisierung sendet. Sie definieren Automatisierungseingaben mit Automatisierungsparametern. Wenn Sie beispielsweise Amazon

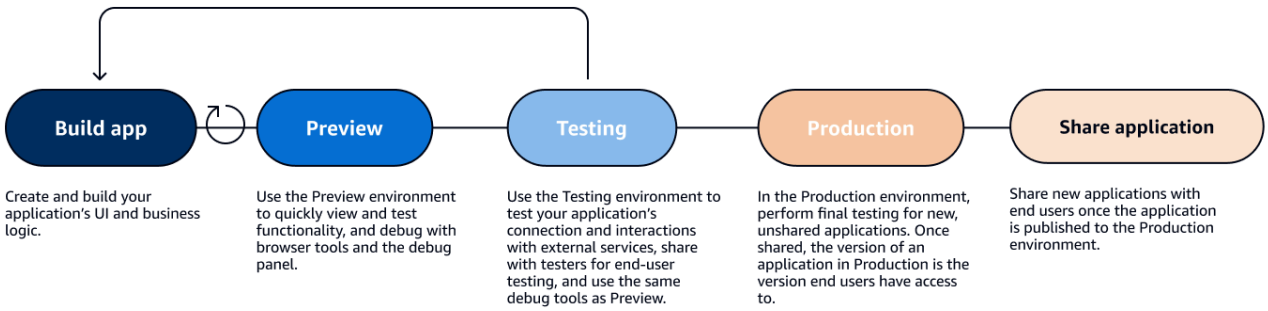
Bedrock verwenden möchten, um eine Textzusammenfassung an den Benutzer zurückzugeben, konfigurieren Sie den Text so, dass er als Automatisierungsparameter zusammengefasst wird.

- Aktionen, die die Bausteine der Funktionalität einer Automatisierung bilden. Sie können sich jede Aktion als einen Schritt in der Automatisierung vorstellen. Aktionen können aufrufen APIs, benutzerdefinierte Funktionen aufrufen JavaScript, Datensätze erstellen und andere Funktionen ausführen. Sie können Aktionen auch in Schleifen oder Bedingungen gruppieren, um die Funktionalität weiter anzupassen. Sie können mit einer Aktion auch andere Automatisierungen aufrufen.
- Eine Automatisierungsausgabe, die Sie in Komponenten oder sogar anderen Automatisierungen verwenden können. Die Automatisierungsausgabe könnte beispielsweise Text sein, der in einer Textkomponente angezeigt wird, ein Bild, das in einer Bildbetrachterkomponente angezeigt werden soll, oder die Eingabe für eine andere Automatisierung.

Der Entwicklungszyklus Ihrer Anwendung

Der Entwicklungszyklus Ihrer Anwendung umfasst die folgenden Phasen: Erstellen, Testen und Veröffentlichen. Er wird als Zyklus bezeichnet, da Sie bei der Erstellung und Bearbeitung Ihrer Anwendung wahrscheinlich zwischen diesen Phasen hin und her iterieren werden.

Die folgende Abbildung zeigt einen vereinfachten Zeitplan für den Lebenszyklus der Anwendungsentwicklung in App Studio:



App Studio bietet verschiedene Tools zur Unterstützung des Lebenszyklus Ihrer Anwendung. Zu diesen Tools gehören die folgenden drei unterschiedlichen Umgebungen, die im vorherigen Diagramm dargestellt sind:

- Die Vorschauumgebung, in der Sie eine Vorschau Ihrer Anwendung anzeigen können, um zu sehen, wie sie für Endbenutzer aussieht, und bestimmte Funktionen testen können. Verwenden Sie die Vorschauumgebung, um Ihre Anwendung schnell zu testen und zu iterieren, ohne sie

veröffentlichen zu müssen. Anwendungen in der Vorschauumgebung kommunizieren oder übertragen keine Daten mit externen Diensten. Das bedeutet, dass Sie Interaktionen und Funktionen, die auf externen Diensten basieren, in der Preview-Umgebung nicht testen können.

- Die Testumgebung, in der Sie die Verbindung und Interaktionen Ihrer Anwendung mit externen Diensten testen können. In dieser Umgebung können Sie auch Endbenutzertests durchführen, indem Sie die in der Testumgebung veröffentlichte Version an Gruppen von Testern weitergeben.
- Die Produktionsumgebung, in der Sie letzte Tests neuer Apps durchführen können, bevor Sie sie mit Endbenutzern teilen. Nachdem die Apps gemeinsam genutzt wurden, ist die Version der Anwendung, die in der Produktionsumgebung veröffentlicht wird, die Version, die Endbenutzer ansehen und verwenden werden.

Weitere Informationen

Da Sie nun mit den Grundlagen der Anwendungsentwicklung in App Studio vertraut sind, können Sie entweder damit beginnen, eine eigene Anwendung zu erstellen, oder sich eingehender mit Konzepten und Ressourcen befassen.

Um mit der Entwicklung zu beginnen, empfehlen wir Ihnen, eines der Tutorials für die ersten Schritte auszuprobieren:

- Im Folgenden erfahren Tutorial: Generieren Sie eine App mit KI Sie, wie Sie den AI Builder-Assistenten verwenden können, um sich einen Vorsprung bei der Erstellung einer App zu verschaffen.
- Im Folgenden erfahren Tutorial: Beginne mit dem Erstellen von einer leeren App Sie, wie Sie eine App von Grund auf neu erstellen und gleichzeitig die Grundlagen erlernen.

Weitere Informationen zu den in diesem Thema erwähnten Ressourcen oder Konzepten finden Sie in den folgenden Themen:

- Connect App Studio über Konnektoren mit anderen Diensten
- Entitäten und Datenaktionen: Konfigurieren Sie das Datenmodell Ihrer App
- Seiten und Komponenten: Erstellen Sie die Benutzeroberfläche Ihrer App
- Automatisierungen und Aktionen: Definieren Sie die Geschäftslogik Ihrer App
- Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen

App Studio einrichten und sich bei AWS App Studio anmelden

Die Einrichtung von AWS App Studio ist je nach Ihrer Rolle unterschiedlich:

- Erstmalige Einrichtung als Administrator AWS oder Unternehmensadministrator: Um App Studio zum ersten Mal als Administrator einzurichten, erstellen Sie ein AWS Konto, falls Sie noch keines haben, erstellen die App Studio-Instanz und konfigurieren den Benutzerzugriff mithilfe von IAM Identity Center-Gruppen. Nachdem die Instanz erstellt wurde, kann jeder Benutzer mit der Administratorrolle in App Studio weitere Einrichtungsaufgaben ausführen, z. B. Konnektoren konfigurieren, um andere Dienste (wie Datenquellen) mit Ihrer App Studio-Instanz zu verbinden. Informationen zur erstmaligen Einrichtung finden Sie unter. Erstmaliges Erstellen und Einrichten einer App Studio-Instanz
- Erste Schritte als Builder: Wenn Sie eine Einladung erhalten, App Studio als Builder beizutreten, müssen Sie die Einladung annehmen und Ihre IAM Identity Center-Benutzeranmeldedaten aktivieren, indem Sie ein Passwort angeben. Danach können Sie sich bei App Studio anmelden und mit der Erstellung von Anwendungen beginnen. Informationen zum Annehmen einer Einladung und zum Beitritt zu einer App Studio-Instanz finden Sie unterEine Einladung zur Teilnahme an App Studio annehmen.

Erstmaliges Erstellen und Einrichten einer App Studio-Instanz

Eröffnen Sie ein AWS Konto

Für die Einrichtung von App Studio ist ein AWS Konto erforderlich. Für die Nutzung von App Studio ist nur ein AWS Konto erforderlich. Entwickler und Administratoren benötigen kein AWS Konto, um App Studio zu verwenden, da der Zugriff mit AWS IAM Identity Center verwaltet wird.

Um ein zu erstellen AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscodes auf der Telefontastatur.

Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontoserstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen

Wenn Sie zum ersten Mal ein AWS Konto erstellen, beginnen Sie mit einem Standardsatz von Anmeldeinformationen mit vollständigem Zugriff auf alle AWS Ressourcen in Ihrem Konto. Diese Identität wird als Root-Benutzer des AWS Kontos bezeichnet. Für die Erstellung von AWS Rollen und Ressourcen zur Verwendung mit App Studio empfehlen wir dringend, nicht den Root-Benutzer des AWS Kontos zu verwenden. Stattdessen empfehlen wir, dass Sie einen Administratorbenutzer erstellen und verwenden.

Verwenden Sie die folgenden Themen, um einen Administratorbenutzer für die Verwaltung von AWS Rollen und Ressourcen für die Verwendung mit App Studio zu erstellen.

- Informationen zu einem einzelnen, eigenständigen AWS Konto finden Sie unter Erstellen Sie Ihren ersten IAM-Benutzer im IAM-Benutzerhandbuch. Sie können einen beliebigen Benutzernamen angeben, für den jedoch die `AdministratorAccess` Berechtigungsrichtlinie erforderlich ist.
- Informationen zu mehreren AWS Konten AWS Organizations, die über verwaltet werden, finden Sie unter Einrichten des AWS-Kontozugriffs für einen IAM Identity AWS IAM Identity Center Center-Administratorbenutzer im Benutzerhandbuch.

Erstellen Sie eine App Studio-Instanz im AWS Management Console

Um App Studio zu verwenden, müssen Sie auf der App Studio-Landingpage im eine Instanz erstellen AWS Management Console. Es gibt zwei Methoden, mit denen eine App Studio-Instanz erstellt werden kann:

1. Einfache Erstellung: Mit dieser vereinfachten Methode richten Sie nur einen Benutzer ein, der im Rahmen der Einrichtung auf App Studio zugreifen und es verwenden kann. Sie sollten diese Methode verwenden, wenn Sie App Studio für Ihre Organisation oder Ihr Team evaluieren oder wenn Sie planen, App Studio nur selbst zu verwenden. Nach der Einrichtung können Sie App

Studio weitere Benutzer oder Gruppen hinzufügen. Beachten Sie, dass Sie diese Methode nicht verwenden können, wenn Sie über eine Organisationsinstanz von IAM Identity Center verfügen.

2. Standarderstellung: Mit dieser Methode fügen Sie Benutzer oder Gruppen hinzu und weisen ihnen im Rahmen der Einrichtung Rollen in App Studio zu. Sie sollten diese Methode verwenden, wenn Sie bei der Einrichtung mehr als einen Benutzer zu App Studio hinzufügen möchten.

> **Note**
>
> Sie können in allen AWS Regionen nur eine Instanz von App Studio erstellen. Wenn Sie über eine bestehende Instanz verfügen, müssen Sie diese löschen, bevor Sie eine weitere erstellen können. Weitere Informationen finden Sie unter Löschen einer App Studio-Instanz.

Easy create

Um eine App Studio-Instanz AWS Management Console mit Easy Create zu erstellen

1. Öffnen Sie die App Studio-Konsole unter https://console.aws.amazon.com/appstudio/.
2. Navigieren Sie zu der AWS Region, in der Sie eine App Studio-Instanz erstellen möchten.
3. Wählen Sie Erste Schritte.
4. Wählen Sie Einfach erstellen und dann Weiter.
5. Die nächsten Schritte zur Einrichtung von App Studio hängen davon ab, ob Sie eine IAM Identity Center-Kontoinstanz haben oder nicht. Weitere Informationen zu IAM Identity Center-Instanzen, einschließlich der verschiedenen Typen und wie Sie herausfinden können, welchen Typ Sie haben, finden Sie unter Organisations- und Kontoinstanzen von IAM Identity Center verwalten im AWS IAM Identity Center-Benutzerhandbuch.
   - Wenn Sie eine Kontoinstanz von IAM Identity Center haben:
     a. Überprüfen Sie unter Kontoberechtigungen die erforderlichen Berechtigungen für die Aktivierung von App Studio. Wenn Ihr Konto nicht über die erforderlichen Berechtigungen verfügt, können Sie App Studio nicht aktivieren. Sie müssen entweder die erforderlichen Berechtigungen zu Ihrem Konto hinzufügen lassen oder zu einem Konto wechseln, das über diese Berechtigungen verfügt.
     b. Suchen Sie unter Benutzer hinzufügen nach der E-Mail-Adresse des Benutzers in Ihrer IAM Identity Center-Kontoinstanz, der auf App Studio zugreift, und wählen Sie sie aus. Dieser Benutzer wird die Administratorrolle in der App Studio-Instanz

haben. Wenn Sie den Benutzer, dem Sie Zugriff auf App Studio gewähren möchten, nicht sehen, müssen Sie ihn möglicherweise zu Ihrer IAM Identity Center-Instanz hinzufügen.

- Wenn Sie keine Kontoinstanz von IAM Identity Center haben:

> **Note**
>
> Bei der Einrichtung von App Studio wird automatisch eine IAM Identity Center-Kontoinstanz mit dem Benutzer erstellt, den Sie während des Einrichtungsvorgangs konfigurieren. Nach Abschluss der Einrichtung können Sie Benutzer und Gruppen in der IAM Identity Center-Konsole unter hinzufügen oder verwalten. https://console.aws.amazon.com/singlesignon/

a. Überprüfen Sie unter Kontoberechtigungen die erforderlichen Berechtigungen für die Aktivierung von App Studio. Wenn Ihr Konto nicht über die erforderlichen Berechtigungen verfügt, können Sie App Studio nicht aktivieren. Sie müssen entweder die erforderlichen Berechtigungen zu Ihrem Konto hinzufügen lassen oder zu einem Konto wechseln, das über diese Berechtigungen verfügt.

b. Geben Sie unter Benutzer hinzufügen eine E-Mail-Adresse, einen Vornamen, einen Nachnamen und einen Benutzernamen für den Benutzer ein, der auf App Studio zugreift. Dieser Benutzer wird die Administratorrolle in der App Studio-Instanz haben.

6. Überprüfen Sie unter Dienstzugriff und Rollen die Servicerollen und die mit dem Dienst verknüpften Rollen, die automatisch erstellt werden, wenn Sie App Studio einrichten, um dem Dienst die erforderlichen Berechtigungen zu gewähren. Wählen Sie Berechtigungen anzeigen, um die genauen Berechtigungen anzuzeigen, die für Servicerollen gewährt wurden, oder Richtlinie anzeigen, um die Berechtigungsrichtlinie anzuzeigen, die der dienstbezogenen Rolle zugeordnet ist.
7. Bestätigen Sie im Feld Bestätigung die Aussagen, indem Sie die entsprechenden Kontrollkästchen aktivieren.
8. Wählen Sie Einrichten, um Ihre Instanz zu erstellen.

> **Note**
>
> Um Ihrer App Studio-Instanz nach der Einrichtung weitere Benutzer oder Gruppen hinzuzufügen, müssen Sie sie zu Ihrer IAM Identity Center-Instanz hinzufügen.

Standard create

Um eine App Studio-Instanz AWS Management Console mit der Standardmethode zu erstellen

1. Öffnen Sie die App Studio-Konsole unter https://console.aws.amazon.com/appstudio/.
2. Navigieren Sie zu der AWS Region, in der Sie eine App Studio-Instanz erstellen möchten.
3. Wählen Sie Erste Schritte.
4. Wählen Sie Standard erstellen und dann Weiter.
5. Die Schritte zur Einrichtung von App Studio hängen davon ab, ob Sie eine IAM Identity Center-Instanz haben oder nicht, und vom Instanztyp. Weitere Informationen zu IAM Identity Center-Instanzen, einschließlich der verschiedenen Typen und wie Sie herausfinden können, welchen Typ Sie haben, finden Sie unter Organisations- und Kontoinstanzen von IAM Identity Center verwalten im AWS IAM Identity Center-Benutzerhandbuch.
   - Wenn Sie über eine Organisationsinstanz von IAM Identity Center verfügen:
     - Wählen Sie unter Zugriff auf App Studio mit Single Sign-On konfigurieren bestehende IAM Identity Center-Gruppen aus, um ihnen Zugriff auf App Studio zu gewähren. App Studio-Gruppen werden auf der Grundlage der angegebenen Konfiguration erstellt. Mitglieder von Gruppen, die zu Admin-Gruppen hinzugefügt wurden, haben die Admin-Rolle, und Mitglieder von Gruppen, die zu Builder-Gruppen hinzugefügt wurden, haben die Builder-Rolle in App Studio. Die Rollen sind wie folgt definiert:
       - Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.
       - Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder Anwendungen anderer Builder verwalten.

- Wenn Sie eine Kontoinstanz der IAM Identity Center-Instanz haben:

  a. Überprüfen Sie unter Kontoberechtigungen die erforderlichen Berechtigungen für die Aktivierung von App Studio. Wenn Ihr Konto nicht über die erforderlichen Berechtigungen verfügt, können Sie App Studio nicht aktivieren. Sie müssen entweder die erforderlichen Berechtigungen zu Ihrem Konto hinzufügen lassen oder zu einem Konto wechseln, das über diese Berechtigungen verfügt.

  b. Wählen Sie unter Zugriff auf App Studio mit Single Sign-On konfigurieren im IAM Identity Center-Konto die Option Bestehende Kontoinstanz verwenden

  c. Wählen Sie unter AWS Region die Region aus, in der sich Ihre IAM Identity Center-Kontoinstanz befindet.

  d. Wählen Sie bestehende IAM Identity Center-Gruppen aus, um ihnen Zugriff auf App Studio zu gewähren. App Studio-Gruppen werden auf der Grundlage der angegebenen Konfiguration erstellt. Mitglieder von Gruppen, die zu Admin-Gruppen hinzugefügt wurden, haben die Admin-Rolle, und Mitglieder von Gruppen, die zu Builder-Gruppen hinzugefügt wurden, haben die Builder-Rolle in App Studio. Die Rollen sind wie folgt definiert:

     - Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.
     - Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder Anwendungen anderer Builder verwalten.

- Wenn Sie keine IAM Identity Center-Instanz haben:

  > **Note**
  >
  > Bei der Einrichtung von App Studio wird automatisch eine IAM Identity Center-Kontoinstanz mit den Gruppen erstellt, die Sie während des Einrichtungsvorgangs konfigurieren. Nach Abschluss der Einrichtung können Sie Benutzer und Gruppen in der IAM Identity Center-Konsole unter hinzufügen oder verwalten. https://console.aws.amazon.com/singlesignon/

a. Überprüfen Sie unter Kontoberechtigungen die erforderlichen Berechtigungen für die Aktivierung von App Studio. Wenn Ihr Konto nicht über die erforderlichen Berechtigungen verfügt, können Sie App Studio nicht aktivieren. Sie müssen entweder die erforderlichen Berechtigungen zu Ihrem Konto hinzufügen lassen oder zu einem Konto wechseln, das über diese Berechtigungen verfügt.

b. Wählen Sie unter Zugriff auf App Studio mit Single Sign-On konfigurieren im IAM Identity Center-Konto die Option Kontoinstanz für mich erstellen aus.

c. Geben Sie unter Benutzer und Gruppen erstellen und zu App Studio hinzufügen einen Namen ein und fügen Sie Benutzer zu einer Admin- und Builder-Gruppe hinzu. Benutzer, die der Admin-Gruppe hinzugefügt werden, haben die Administratorrolle in App Studio, und Benutzer, die der Builder-Gruppe hinzugefügt werden, haben die Builder-Rolle. Die Rollen sind wie folgt definiert:

   - Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.
   - Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder Anwendungen anderer Builder verwalten.

> **Important**
>
> Sie müssen sich selbst als Benutzer der Admin-Gruppe hinzufügen, um App Studio einzurichten und nach der Einrichtung Administratorzugriff zu haben.

6. Sehen Sie sich unter Dienstzugriff und Rollen die Servicerollen und die dienstbezogenen Rollen an, die automatisch erstellt werden, wenn Sie App Studio einrichten, um dem Dienst die erforderlichen Berechtigungen zu gewähren. Wählen Sie Berechtigungen anzeigen, um die genauen Berechtigungen anzuzeigen, die für Servicerollen gewährt wurden, oder Richtlinie anzeigen, um die Berechtigungsrichtlinie anzuzeigen, die der dienstbezogenen Rolle zugeordnet ist.
7. Bestätigen Sie unter Bestätigung die Aussagen, indem Sie die entsprechenden Kontrollkästchen aktivieren.

8. Wählen Sie Einrichten, um Ihre Instanz zu erstellen.

Eine Einladung zur Teilnahme an App Studio annehmen

Der Zugriff auf App Studio wird vom IAM Identity Center verwaltet. Das bedeutet, dass jeder Benutzer, der App Studio verwenden möchte, einen Benutzer in IAM Identity Center konfigurieren und zu einer Gruppe gehören muss, die von einem Administrator zu App Studio hinzugefügt wurde. Wenn ein Administrator Sie einlädt, IAM Identity Center beizutreten, erhalten Sie eine E-Mail, in der Sie aufgefordert werden, die Einladung anzunehmen und Ihre Benutzeranmeldedaten zu aktivieren. Nachdem sie aktiviert wurden, können Sie sich mit diesen Anmeldedaten bei App Studio anmelden.

Um eine Einladung zu IAM Identity Center anzunehmen, um auf App Studio zuzugreifen

1. Wenn Sie eine Einladungs-E-Mail erhalten, folgen Sie den Schritten, um ein Passwort einzugeben und Ihre Benutzeranmeldedaten in IAM Identity Center zu aktivieren. Weitere Informationen finden Sie unter Annahme der Einladung zum Beitritt zu IAM Identity Center.
2. Nachdem Sie Ihre Benutzeranmeldedaten aktiviert haben, verwenden Sie sie, um sich bei Ihrer App Studio-Instanz anzumelden.

Erste Schritte mit AWS App Studio

Die folgenden Tutorials für die ersten Schritte führen Sie durch die Erstellung Ihrer ersten Anwendung in App Studio.

- Empfehlung: Informationen zur Verwendung generativer KI zur Beschreibung der App, die Sie erstellen möchten, und zur automatischen Erstellung der App und ihrer Ressourcen finden Sie unterTutorial: Generieren Sie eine App mit KI.
- Informationen dazu, wie Sie mit der Entwicklung von einer leeren App aus beginnen, finden Sie unterTutorial: Beginne mit dem Erstellen von einer leeren App.

Tutorial: Generieren Sie eine App mit KI

AWS App Studio enthält generative KI-Funktionen im gesamten Service, um die Anwendungserstellung zu beschleunigen. In diesem Tutorial erfahren Sie, wie Sie mithilfe von KI eine App generieren, indem Sie Ihre App in natürlicher Sprache beschreiben.

Die Verwendung von KI zur Generierung einer App ist eine hervorragende Möglichkeit, mit der Entwicklung zu beginnen, da viele der Ressourcen der App für Sie erstellt wurden. In der Regel ist es viel einfacher, mit der Entwicklung einer generierten App mit vorhandenen Ressourcen zu beginnen, als mit einer leeren App zu beginnen.

> **Note**
>
> Eine ähnliche exemplarische Vorgehensweise mit Bildern finden Sie im Blogbeitrag Erstellen von Unternehmensanwendungen mit natürlicher Sprache mithilfe von AWS App Studio (Vorschau). Der Blogbeitrag enthält auch Informationen zum Einrichten und Konfigurieren von Ressourcen für Administratoren. Sie können jedoch bei Bedarf direkt zum Abschnitt über das Erstellen von Anwendungen springen.

Wenn App Studio eine App mit KI generiert, erstellt es die App mit den folgenden Ressourcen, die auf die von Ihnen beschriebene App zugeschnitten sind:

- Seiten und Komponenten: Komponenten sind die Bausteine der Benutzeroberfläche Ihrer Anwendung. Sie stellen visuelle Elemente wie Tabellen, Formulare und Schaltflächen dar.

Jede Komponente hat ihre eigenen Eigenschaften, und Sie können eine Komponente an Ihre spezifischen Anforderungen anpassen. Seiten sind die Container für Komponenten.

- Automatisierungen: Sie verwenden Automatisierungen, um die Logik und die Workflows zu definieren, die das Verhalten Ihrer Anwendung steuern. Beispielsweise können Sie Automatisierungen verwenden, um Zeilen in einer Datentabelle zu erstellen, zu aktualisieren, zu lesen oder zu löschen oder um mit Objekten in einem Amazon S3 S3-Bucket zu interagieren. Sie können sie auch verwenden, um Aufgaben wie Datenvalidierung, Benachrichtigungen oder Integrationen mit anderen Systemen zu erledigen.
- Entitäten: Daten sind die Informationen, die Ihrer Anwendung zugrunde liegen. Die generierte App erstellt Entitäten, die Tabellen ähneln. Entitäten stellen die verschiedenen Arten von Daten dar, die Sie speichern und mit denen Sie arbeiten müssen, z. B. Kunden, Produkte oder Bestellungen. Sie können diese Datenmodelle mithilfe von App Studio-Connectors mit einer Vielzahl von Datenquellen APIs, einschließlich AWS Diensten und externen Datenquellen, verbinden.

Inhalt

- Voraussetzungen
- Schritt 1: Erstellen einer -Anwendung
- Schritt 2: Erkunden Sie Ihre neue Anwendung
  - Erkunden Sie Seiten und Komponenten
  - Erkunden Sie Automatisierungen und Aktionen
  - Untersuchen Sie Daten mit Entitäten
- Schritt 3: Zeigen Sie eine Vorschau Ihrer Anwendung an
- Nächste Schritte

Voraussetzungen

Bevor Sie beginnen, sollten Sie die folgenden Voraussetzungen überprüfen und erfüllen:

- Zugriff auf AWS App Studio. Weitere Informationen finden Sie unter App Studio einrichten und sich bei AWS App Studio anmelden.
- Optional: Lesen SieAWS App Studio-Konzepte, um sich mit wichtigen App Studio-Konzepten vertraut zu machen.

Schritt 1: Erstellen einer -Anwendung

Der erste Schritt beim Generieren einer App besteht darin, dem KI-Assistenten in App Studio die App zu beschreiben, die Sie erstellen möchten. Sie können die Anwendung, die generiert wird, überprüfen und vor der Generierung nach Bedarf iterieren.

Um Ihre App mithilfe von KI zu generieren

1. Melden Sie sich bei App Studio an.
2. Wählen Sie in der linken Navigationsleiste Builder Hub und dann + App erstellen aus.
3. Wählen Sie App mit KI generieren aus.
4. Geben Sie im Feld App-Name einen Namen für Ihre App ein.
5. Wählen Sie im Dialogfeld „Datenquellen auswählen“ die Option Überspringen aus.
6. Sie können mit der Definition der App beginnen, die Sie generieren möchten, indem Sie sie im Textfeld beschreiben oder indem Sie in einer Beispielaufforderung die Option Anpassen auswählen. Nachdem Sie Ihre App beschrieben haben, generiert App Studio die App-Anforderungen und Details, die Sie überprüfen können. Dazu gehören Anwendungsfälle, Benutzerabläufe und Datenmodelle.
7. Verwenden Sie das Textfeld, um nach Bedarf mit Ihrer App zu iterieren, bis Sie mit den Anforderungen und Details zufrieden sind.
8. Wenn Sie bereit sind, Ihre App zu generieren und mit der Erstellung zu beginnen, wählen Sie App generieren.
9. Optional können Sie sich ein kurzes Video ansehen, in dem detailliert beschrieben wird, wie Sie sich in Ihrer neuen App zurechtfinden.
10. Wählen Sie App bearbeiten, um die Entwicklungsumgebung für Ihre App aufzurufen.

Schritt 2: Erkunden Sie Ihre neue Anwendung

In der Entwicklungsumgebung finden Sie die folgenden Ressourcen:

- Eine Arbeitsfläche, die Sie verwenden, um Ihre Anwendung anzusehen oder zu bearbeiten. Die Arbeitsfläche ändert sich je nach der ausgewählten Ressource.
- Navigationsregisterkarten oben auf der Leinwand. Die Registerkarten werden in der folgenden Liste beschrieben:

  - Seiten: Hier verwenden Sie Seiten und Komponenten, um die Benutzeroberfläche Ihrer App zu entwerfen.
  - Automatisierungen: Hier verwenden Sie Aktionen in Automatisierungen, um die Geschäftslogik Ihrer App zu definieren.
  - Daten: Hier definierst du Entitäten, ihre Felder, Beispieldaten und Datenaktionen, um die Datenmodelle deiner App zu definieren.
  - App-Einstellungen: Hier definieren Sie Einstellungen für Ihre App, einschließlich App-Rollen, mit denen Sie die rollenbasierte Sichtbarkeit von Seiten für Ihre Endbenutzer definieren.
- Ein Navigationsmenü auf der linken Seite, das Ressourcen enthält, je nachdem, welche Registerkarte Sie gerade aufrufen.
- Ein Menü auf der rechten Seite, das Ressourcen und Eigenschaften für ausgewählte Ressourcen auf den Tabs Seiten und Automatisierungen auflistet.
- Eine Debug-Konsole, die Warnungen und Fehler am unteren Rand des Builders anzeigt. In Ihrer generierten App sind möglicherweise Fehler vorhanden. Dies ist wahrscheinlich auf Automatisierungen zurückzuführen, für die ein konfigurierter Connector erforderlich ist, um Aktionen wie das Senden einer E-Mail mit Amazon Simple Email Service auszuführen.
- Ein Ask AI-Chat-Fenster, in dem Sie kontextuelle Hilfe vom AI Builder-Assistenten erhalten.

Schauen wir uns die Tabs Seiten, Automatisierungen und Daten genauer an.

Erkunden Sie Seiten und Komponenten

Auf der Registerkarte Seiten werden Seiten und ihre Komponenten angezeigt, die für Sie generiert wurden.

Jede Seite stellt einen Bildschirm der Benutzeroberfläche (UI) Ihrer Anwendung dar, mit dem Ihre Benutzer interagieren werden. Auf diesen Seiten finden Sie verschiedene Komponenten (wie Tabellen, Formulare und Schaltflächen), mit denen Sie das gewünschte Layout und die gewünschte Funktionalität erstellen können.

Nehmen Sie sich etwas Zeit, um sich die Seiten und ihre Komponenten anzusehen, indem Sie das Navigationsmenü auf der linken Seite verwenden. Wenn Sie eine Seite oder Komponente auswählen, können Sie im Menü auf der rechten Seite Eigenschaften wählen.

Erkunden Sie Automatisierungen und Aktionen

Auf der Registerkarte Automatisierungen werden Automatisierungen und ihre Aktionen angezeigt, die für Sie generiert wurden.

Automatisierungen definieren die Geschäftslogik Ihrer App, z. B. das Erstellen, Anzeigen, Aktualisieren oder Löschen von Dateneinträgen, das Senden von E-Mails oder sogar das Aufrufen von APIs Lambda-Funktionen.

Nehmen Sie sich etwas Zeit, um sich die Automatisierungen anzusehen, indem Sie das Navigationsmenü auf der linken Seite verwenden. Wenn Sie eine Automatisierung auswählen, können Sie ihre Eigenschaften im Eigenschaftenmenü auf der rechten Seite einsehen. Eine Automatisierung enthält die folgenden Ressourcen:

- Automatisierungen bestehen aus einzelnen Aktionen, die die Bausteine der Geschäftslogik Ihrer App bilden. Sie können die Aktionen einer Automatisierung im Navigationsmenü auf der linken Seite oder auf der Arbeitsfläche einer ausgewählten Automatisierung anzeigen. Wenn Sie eine Aktion auswählen, können Sie ihre Eigenschaften im Eigenschaftenmenü auf der rechten Seite einsehen.
- Automatisierungsparameter geben an, wie Daten an eine Automatisierung übergeben werden. Parameter dienen als Platzhalter, die bei der Ausführung der Automatisierung durch tatsächliche Werte ersetzt werden. Auf diese Weise können Sie dieselbe Automatisierung jedes Mal mit unterschiedlichen Eingaben verwenden.
- In der Automatisierungsausgabe konfigurieren Sie das Ergebnis einer Automatisierung. Standardmäßig hat eine Automatisierung keine Ausgabe. Um das Ergebnis einer Automatisierung in Komponenten oder anderen Automatisierungen zu verwenden, müssen Sie sie hier definieren.

Weitere Informationen finden Sie unter Automationskonzepte.

Untersuchen Sie Daten mit Entitäten

Auf der Registerkarte Daten werden Entitäten angezeigt, die für Sie generiert wurden.

Entitäten stellen Tabellen dar, die die Daten Ihrer Anwendung enthalten, ähnlich wie Tabellen in einer Datenbank. Anstatt die Benutzeroberfläche (UI) und die Automatisierungen Ihrer Anwendung direkt mit Datenquellen zu verbinden, stellen sie zunächst eine Verbindung zu Entitäten her. Entitäten agieren als Vermittler zwischen Ihrer eigentlichen Datenquelle und Ihrer App Studio-App. Dies bietet einen zentralen Ort für die Verwaltung und den Zugriff auf Ihre Daten.

Nehmen Sie sich etwas Zeit, um sich die generierten Entitäten anzusehen, indem Sie sie im Navigationsmenü auf der linken Seite auswählen. Sie können die folgenden Details überprüfen:

- Auf der Registerkarte Konfiguration werden der Entitätsname und die zugehörigen Felder angezeigt, die die Spalten Ihrer Entität darstellen.
- Auf der Registerkarte Datenaktionen werden die Datenaktionen angezeigt, die mit Ihrer Entität generiert wurden. Komponenten und Automatisierungen können Datenaktionen verwenden, um Daten aus Ihrer Entität abzurufen.
- Auf der Registerkarte Beispieldaten werden Beispieldaten angezeigt, mit denen Sie Ihre App in der Entwicklungsumgebung testen können (die nicht mit externen Diensten kommuniziert). Weitere Informationen zu Umgebungen finden Sie unterAnwendungsumgebungen.
- Auf der Registerkarte Verbindung werden Informationen zu den externen Datenquellen angezeigt, mit denen die Entität verbunden ist. App Studio bietet eine verwaltete Datenspeicherlösung, die eine DynamoDB-Tabelle verwendet. Weitere Informationen finden Sie unter Verwaltete Datenentitäten in AWS App Studio.

Schritt 3: Zeigen Sie eine Vorschau Ihrer Anwendung an

Sie können eine Vorschau einer Anwendung in App Studio anzeigen, um zu sehen, wie sie Benutzern angezeigt wird. Sie können ihre Funktionalität auch testen, indem Sie sie verwenden und die Protokolle in einem Debug-Bereich überprüfen.

Die Anwendungsvorschauumgebung unterstützt nicht die Anzeige von Live-Daten oder die Verbindung mit externen Ressourcen über Konnektoren, z. B. Datenquellen. Stattdessen können Sie Beispieldaten und simulierte Ausgaben verwenden, um die Funktionalität zu testen.

Um eine Vorschau Ihrer App zum Testen anzuzeigen

1. Wählen Sie in der oberen rechten Ecke des App Builders die Option Vorschau aus.
2. Interagiere mit den Seiten deiner App.

Nächste Schritte

Nachdem Sie Ihre erste App erstellt haben, sind hier die nächsten Schritte:

- Eine weitere Anleitung für die ersten Schritte mit Bildern finden Sie im Blogbeitrag Erstellen von Unternehmensanwendungen mit natürlicher Sprache mithilfe von AWS App Studio (Vorschau).

- Apps verwenden Konnektoren, um Daten zu senden und zu empfangen oder um mit externen Diensten (sowohl AWS Diensten als auch Diensten von Drittanbietern) zu kommunizieren. Es ist notwendig, mehr über Konnektoren und deren Konfiguration zu erfahren, um Apps zu erstellen. Beachten Sie, dass Sie über die Administratorrolle verfügen müssen, um Konnektoren verwalten zu können. Weitere Informationen hierzu finden Sie unter Connect App Studio über Konnektoren mit anderen Diensten.
- Weitere Informationen zur Vorschau, Veröffentlichung und eventuellen Freigabe Ihrer App für Endbenutzer finden Sie unterAnwendungen in der Vorschau anzeigen, veröffentlichen und teilen.
- Erkunden und aktualisieren Sie die App, die Sie generiert haben, weiter, um praktische Erfahrungen zu sammeln.
- Weitere Informationen zum Erstellen von Apps finden Sie unterBuilder-Dokumentation. Insbesondere die folgenden Themen könnten hilfreich sein, um sich mit ihnen zu befassen:
  - Referenz zu Automation-Aktionen
  - Referenz zu den Komponenten
  - Interaktion mit Amazon Simple Storage Service mit Komponenten und Automatisierungen
  - Sicherheitsüberlegungen und Abhilfemaßnahmen

Tutorial: Beginne mit dem Erstellen von einer leeren App

In diesem Tutorial erstellen Sie mithilfe von AWS App Studio eine interne Anwendung für Anfragen zu Kundengesprächen. Sie erfahren, wie Sie Apps in App Studio erstellen, und konzentrieren sich dabei auf reale Anwendungsfälle und praktische Beispiele. Außerdem erfahren Sie mehr über die Definition von Datenstrukturen, das UI-Design und die Anwendungsbereitstellung.

> **Note**
>
> In diesem Tutorial wird beschrieben, wie Sie eine App von Grund auf neu erstellen, beginnend mit einer leeren App. In der Regel ist es viel schneller und einfacher, KI zur Generierung einer App und ihrer Ressourcen zu verwenden, indem eine Beschreibung der App bereitgestellt wird, die Sie erstellen möchten. Weitere Informationen finden Sie unter Tutorial: Generieren Sie eine App mit KI.

Der Schlüssel zum Verständnis der Anwendungsentwicklung mit App Studio liegt darin, die folgenden vier Kernkonzepte zu verstehen und zu verstehen, wie sie zusammenarbeiten: Komponenten, Automatisierungen, Daten und Konnektoren.

- Komponenten: Komponenten sind die Bausteine der Benutzeroberfläche Ihrer Anwendung. Sie stellen visuelle Elemente wie Tabellen, Formulare und Schaltflächen dar. Jede Komponente hat ihre eigenen Eigenschaften, die Sie an Ihre spezifischen Anforderungen anpassen können.
- Automatisierungen: Mit Automatisierungen können Sie die Logik und die Workflows definieren, die das Verhalten Ihrer Anwendung steuern. Sie können Automatisierungen verwenden, um Zeilen in einer Datentabelle zu erstellen, zu aktualisieren, zu lesen oder zu löschen oder um mit Objekten in einem Amazon S3 S3-Bucket zu interagieren. Sie können sie auch verwenden, um Aufgaben wie Datenvalidierung, Benachrichtigungen oder Integrationen mit anderen Systemen zu erledigen.
- Daten: Daten sind die Informationen, die Ihrer Anwendung zugrunde liegen. In App Studio können Sie Datenmodelle, sogenannte Entitäten, definieren. Entitäten stellen die verschiedenen Arten von Daten dar, die Sie speichern und mit denen Sie arbeiten müssen, z. B. Anfragen zu Kundengesprächen, Tagesordnungen oder Teilnehmer. Sie können diese Datenmodelle mithilfe von App Studio-Connectors mit einer Vielzahl von Datenquellen verbinden APIs, einschließlich AWS Diensten und externen Datenquellen.
- Konnektoren: App Studio bietet Verbindungen mit einer Vielzahl von Datenquellen, darunter AWS Dienste wie Aurora, DynamoDB und Amazon Redshift. Zu den Datenquellen gehören auch Dienste von Drittanbietern wie Salesforce oder viele andere, die OpenAPI oder generische API-Konnektoren verwenden. Sie können App Studio-Konnektoren verwenden, um Daten und Funktionen aus diesen unternehmensgerechten Diensten und externen Anwendungen einfach in Ihre Anwendungen zu integrieren.

Im weiteren Verlauf des Tutorials erfahren Sie, wie die wichtigsten Konzepte von Komponenten, Daten und Automatisierung zusammenwirken, um Ihre interne Anwendung für Anfragen zu Kundenbesprechungen zu erstellen.

Im Folgenden finden Sie allgemeine Schritte, die beschreiben, was Sie in diesem Tutorial tun werden:

1. Beginnen Sie mit Daten: Viele Anwendungen beginnen mit einem Datenmodell, daher beginnt dieses Tutorial auch mit Daten. Um die App für Anfragen zu Kundengesprächen zu erstellen, erstellen Sie zunächst eine `MeetingRequests` Entität. Diese Entität stellt die Datenstruktur zum Speichern aller relevanten Informationen zu Besprechungsanfragen dar, z. B. Kundenname, Besprechungsdatum, Tagesordnung und Teilnehmer. Dieses Datenmodell dient als Grundlage für

Ihre Anwendung und unterstützt die verschiedenen Komponenten und Automatisierungen, die Sie erstellen werden.

2. Erstellen Sie Ihre Benutzeroberfläche (UI): Nachdem das Datenmodell eingerichtet ist, führt Sie das Tutorial durch die Erstellung der Benutzeroberfläche (UI). In App Studio erstellen Sie die Benutzeroberfläche, indem Sie Seiten hinzufügen und ihnen Komponenten hinzufügen. Sie fügen Komponenten wie Tabellen, Detailansichten und Kalender zu einer Dashboard-Seite für Besprechungsanfragen hinzu. Diese Komponenten sind so konzipiert, dass sie die in der `MeetingRequests` Entität gespeicherten Daten anzeigen und mit ihnen interagieren. Auf diese Weise können Ihre Benutzer Kundengespräche anzeigen, verwalten und planen. Sie werden auch eine Seite zur Erstellung von Besprechungsanfragen erstellen. Diese Seite enthält eine Formularkomponente zum Sammeln von Daten und eine Button-Komponente zum Absenden von Daten.
3. Fügen Sie Geschäftslogik mit Automatisierungen hinzu: Um die Funktionalität Ihrer Anwendung zu verbessern, konfigurieren Sie einige der Komponenten so, dass Benutzerinteraktionen möglich sind. Einige Beispiele sind das Navigieren zu einer Seite oder das Erstellen eines neuen Datensatzes für eine Besprechungsanfrage in der `MeetingRequests` Entität.
4. Verbesserung durch Validierung und Ausdrücke: Um die Integrität und Genauigkeit Ihrer Daten zu gewährleisten, fügen Sie der Formularkomponente Validierungsregeln hinzu. Auf diese Weise können Sie sicherstellen, dass Benutzer bei der Erstellung neuer Datensätze für Besprechungsanfragen vollständige und gültige Informationen angeben. Außerdem verwenden Sie Ausdrücke, um auf Daten in Ihrer Anwendung zu verweisen und diese zu bearbeiten, sodass Sie dynamische und kontextbezogene Informationen auf der gesamten Benutzeroberfläche anzeigen können.
5. Vorschau und Test: Bevor Sie Ihre Anwendung bereitstellen, haben Sie die Möglichkeit, eine Vorschau anzuzeigen und sie gründlich zu testen. Auf diese Weise können Sie überprüfen, ob die Komponenten, Daten und Automatisierungen reibungslos zusammenarbeiten. Dies bietet Ihren Benutzern ein reibungsloses und intuitives Erlebnis.
6. Veröffentlichen Sie die Anwendung: Abschließend stellen Sie Ihre ausgefüllte interne Anwendung für Anfragen zu Kundengesprächen bereit und machen sie Ihren Benutzern zugänglich. Mit der Leistungsfähigkeit des Low-Code-Ansatzes in App Studio haben Sie eine maßgeschneiderte Anwendung erstellt, die die spezifischen Anforderungen Ihres Unternehmens erfüllt, ohne dass umfangreiche Programmierkenntnisse erforderlich sind.

Inhalt

- Voraussetzungen

- Schritt 1: Erstellen einer -Anwendung
- Schritt 2: Erstellen Sie eine Entität, um die Daten Ihrer App zu definieren
  - Erstellen Sie eine verwaltete Entität
  - Fügen Sie Ihrer Entität Felder hinzu
- Schritt 3: Entwerfen Sie die Benutzeroberfläche (UI) und die Logik
  - Fügen Sie eine Dashboard-Seite für Besprechungsanfragen hinzu
  - Fügen Sie eine Seite zur Erstellung einer Besprechungsanfrage hinzu
- Schritt 4: Zeigen Sie eine Vorschau der Anwendung an
- Schritt 5: Veröffentlichen Sie die Anwendung in der Testumgebung
- Nächste Schritte

Voraussetzungen

Bevor Sie beginnen, sollten Sie die folgenden Voraussetzungen überprüfen und erfüllen:

- Zugriff auf AWS App Studio. Weitere Informationen finden Sie unter App Studio einrichten und sich bei AWS App Studio anmelden.
- Optional: Lesen SieAWS App Studio-Konzepte, um sich mit wichtigen App Studio-Konzepten vertraut zu machen.
- Optional: Kenntnisse der grundlegenden Konzepte der Webentwicklung, wie z. B. der JavaScript Syntax.
- Optional: Vertrautheit mit AWS Diensten.

Schritt 1: Erstellen einer -Anwendung

1. Melden Sie sich bei App Studio an.
2. Wählen Sie in der linken Navigationsleiste Builder Hub und dann + App erstellen aus.
3. Wählen Sie Von vorne beginnen.
4. Geben Sie im Feld App-Name einen Namen für Ihre App ein, z. **`Customer Meeting Requests`** B.
5. Wenn Sie aufgefordert werden, Datenquellen oder einen Konnektor auszuwählen, wählen Sie für die Zwecke dieses Tutorials die Option Überspringen.
6. Wählen Sie Weiter aus, um fortzufahren.

7. (Optional): Sehen Sie sich das Video-Tutorial an, um einen schnellen Überblick über die Erstellung von Apps in App Studio zu erhalten.
8. Wählen Sie App bearbeiten, um zum App Studio App Builder zu gelangen.

Schritt 2: Erstellen Sie eine Entität, um die Daten Ihrer App zu definieren

Entitäten stellen Tabellen dar, die die Daten Ihrer Anwendung enthalten, ähnlich wie Tabellen in einer Datenbank. Anstatt dass die Benutzeroberfläche (UI) und die Automatisierungen Ihrer Anwendung eine direkte Verbindung zu Datenquellen herstellen, stellen sie zuerst eine Verbindung zu Entitäten her. Entitäten agieren als Vermittler zwischen Ihrer eigentlichen Datenquelle und Ihrer App Studio-App und bieten einen zentralen Ort für die Verwaltung und den Zugriff auf Ihre Daten.

Es gibt vier Möglichkeiten, eine Entität zu erstellen. Für dieses Tutorial verwenden Sie die von App Studio verwaltete Entität.

Erstellen Sie eine verwaltete Entität

Durch das Erstellen einer verwalteten Entität wird auch eine entsprechende DynamoDB-Tabelle erstellt, die App Studio verwaltet. Wenn Änderungen an der Entität in der App Studio-App vorgenommen werden, wird die DynamoDB-Tabelle automatisch aktualisiert. Mit dieser Option müssen Sie keine Datenquelle eines Drittanbieters manuell erstellen, verwalten oder eine Verbindung zu einer Datenquelle herstellen oder die Zuordnung von Entitätsfeldern zu Tabellenspalten festlegen.

Wenn Sie eine Entität erstellen, müssen Sie ein Primärschlüsselfeld definieren. Ein Primärschlüssel dient als eindeutiger Bezeichner für jeden Datensatz oder jede Zeile in der Entität. Dadurch wird sichergestellt, dass jeder Datensatz leicht identifiziert und ohne Mehrdeutigkeit abgerufen werden kann. Der Primärschlüssel besteht aus den folgenden Eigenschaften:

- Primärschlüsselname: Ein Name für das Primärschlüsselfeld der Entität.
- Primärschlüssel-Datentyp: Der Typ des Primärschlüsselfeldes. In App Studio sind die unterstützten Primärschlüsseltypen String für Text und Float für eine Zahl. Ein Textprimärschlüssel (like*`meetingName`*) hätte den Typ String und ein numerischer Primärschlüssel (like*`meetingId`*) hätte den Typ Float.

Der Primärschlüssel ist eine wichtige Komponente einer Entität, da er die Datenintegrität gewährleistet, Datenduplikationen verhindert und ein effizientes Abrufen und Abfragen von Daten ermöglicht.

Um eine verwaltete Entität zu erstellen

1. Wählen Sie in der oberen Menüleiste die Option Daten aus.
2. Wählen Sie + Entität erstellen.
3. Wählen Sie „Verwaltete App Studio-Entität erstellen“.
4. Geben Sie im Feld Entitätsname einen Namen für Ihre Entität ein. Geben Sie für dieses Tutorial **`MeetingRequests`** ein.
5. Geben Sie im Feld Primärschlüssel die Bezeichnung für den Primärschlüsselnamen ein, der der Primärschlüsselspalte in Ihrer Entität zugewiesen werden soll. Geben Sie für dieses Tutorial **`requestID`** ein.
6. Wählen Sie als Primärschlüssel-Datentyp die Option Float aus.
7. Klicken Sie auf Create entity (Entity erstellen).

Fügen Sie Ihrer Entität Felder hinzu

Für jedes Feld geben Sie den Anzeigenamen an. Dabei handelt es sich um die Bezeichnung, die für App-Benutzer sichtbar ist. Der Anzeigename kann Leerzeichen und Sonderzeichen enthalten, muss jedoch innerhalb der Entität eindeutig sein. Der Anzeigename dient als benutzerfreundliche Bezeichnung für das Feld und hilft Benutzern, seinen Zweck leicht zu identifizieren und zu verstehen.

Als Nächstes geben Sie den Systemnamen an, eine eindeutige Kennung, die intern von der Anwendung verwendet wird, um auf das Feld zu verweisen. Der Systemname sollte kurz sein und keine Leerzeichen oder Sonderzeichen enthalten. Der Systemname ermöglicht es der Anwendung, Änderungen an den Felddaten vorzunehmen. Er dient als eindeutiger Bezugspunkt für das Feld innerhalb der Anwendung.

Schließlich wählen Sie den Datentyp aus, der der Art von Daten, die Sie in dem Feld speichern möchten, am besten entspricht, z. B. Zeichenfolge (Text), Boolean (wahr/falsch), Datum, Dezimal, Float, Integer oder. DateTime Durch die Definition des geeigneten Datentyps wird die Datenintegrität gewährleistet und die korrekte Handhabung und Verarbeitung der Feldwerte ermöglicht. Wenn Sie beispielsweise Kundennamen in Ihrer Besprechungsanfrage speichern, würden Sie den `String` Datentyp für Textwerte auswählen.

Um Ihrer **`MeetingRequests`** Entität Felder hinzuzufügen

- Wählen Sie + Feld hinzufügen, um die folgenden vier Felder hinzuzufügen:

a. Fügen Sie ein Feld hinzu, das den Namen eines Kunden mit den folgenden Informationen darstellt:

   - Anzeigename: **`Customer name`**
   - Name des Systems: **`customerName`**
   - Datentyp: **`String`**

b. Fügen Sie ein Feld hinzu, das das Besprechungsdatum mit den folgenden Informationen darstellt:

   - Anzeigename: **`Meeting date`**
   - Name des Systems: **`meetingDate`**
   - Datentyp: **`DateTime`**

c. Fügen Sie ein Feld hinzu, das die Tagesordnung der Sitzung mit den folgenden Informationen darstellt:

   - Anzeigename: **`Agenda`**
   - Name des Systems: **`agenda`**
   - Datentyp: **`String`**

d. Fügen Sie ein Feld für die Besprechungsteilnehmer mit den folgenden Informationen hinzu:

   - Anzeigename: **`Attendees`**
   - Name des Systems: **`attendees`**
   - Datentyp: **`String`**

Sie können Ihrer Entität Beispieldaten hinzufügen, mit denen Sie Ihre Anwendung testen und in der Vorschau anzeigen können, bevor Sie sie veröffentlichen. Durch das Hinzufügen von bis zu 500 Zeilen mit Scheindaten können Sie reale Szenarien simulieren und untersuchen, wie Ihre Anwendung verschiedene Datentypen verarbeitet und anzeigt, ohne sich auf tatsächliche Daten verlassen oder eine Verbindung zu externen Diensten herstellen zu müssen. Auf diese Weise können Sie Probleme oder Inkonsistenzen frühzeitig im Entwicklungsprozess erkennen und beheben. Dadurch wird sichergestellt, dass Ihre Anwendung beim Umgang mit tatsächlichen Daten wie vorgesehen funktioniert.

Um Ihrer Entität Beispieldaten hinzuzufügen

1. Wählen Sie im Banner den Tab Beispieldaten aus.
2. Wählen Sie Weitere Beispieldaten generieren aus.
3. Wählen Sie Speichern.

Wählen Sie optional im Banner Verbindung aus, um die Details über den Connector und die für Sie erstellte DynamoDB-Tabelle zu überprüfen.

Schritt 3: Entwerfen Sie die Benutzeroberfläche (UI) und die Logik

Fügen Sie eine Dashboard-Seite für Besprechungsanfragen hinzu

In App Studio stellt jede Seite einen Bildschirm der Benutzeroberfläche (UI) Ihrer Anwendung dar, mit dem Ihre Benutzer interagieren werden. Auf diesen Seiten können Sie verschiedene Komponenten wie Tabellen, Formulare und Schaltflächen hinzufügen, um das gewünschte Layout und die gewünschte Funktionalität zu erstellen.

Neu erstellte Anwendungen verfügen über eine Standardseite, sodass Sie diese Seite umbenennen, anstatt eine neue hinzuzufügen, die Sie als einfache Dashboardseite für Besprechungsanfragen verwenden können.

Um die Standardseite umzubenennen

1. Wählen Sie im Navigationsmenü der oberen Leiste die Option Seiten aus.
2. Doppelklicken Sie im linken Bereich auf Page1, benennen Sie es um **`MeetingRequestsDashboard`** in und drücken Sie die Eingabetaste.

Fügen Sie nun der Seite eine Tabellenkomponente hinzu, die zur Anzeige von Besprechungsanfragen verwendet wird.

Um der Dashboard-Seite für Besprechungsanfragen eine Tabellenkomponente hinzuzufügen

1. Suchen Sie im rechten Bedienfeld „Komponenten“ die Tabellenkomponente und ziehen Sie sie auf die Arbeitsfläche.
2. Wählen Sie die Tabelle auf der Arbeitsfläche aus, um sie auszuwählen.
3. Aktualisieren Sie im Eigenschaftenbereich auf der rechten Seite die folgenden Einstellungen:

a. Wählen Sie das Stiftsymbol, in das Sie die Tabelle umbenennen möchten**`meetingRequestsTable`**.

b. Wählen Sie in der Dropdownliste Quelle die Option Entität aus.

c. Wählen Sie in der Dropdownliste Datenaktionen die Entität aus, die Sie erstellt haben (**`MeetingRequests`**), und wählen Sie + Datenaktionen hinzufügen aus.

4. Wenn Sie dazu aufgefordert werden, wählen Sie GetAll aus.

> **Note**
>
> Die Datenaktion „Alle abrufen“ ist eine bestimmte Art von Datenaktion, die alle Datensätze (Zeilen) von einer angegebenen Entität abruft. Wenn Sie beispielsweise die Datenaktion „Alle abrufen“ einer Tabellenkomponente zuordnen, wird die Tabelle automatisch mit allen Daten aus der verbundenen Entität aufgefüllt und jeder Datensatz wird als Zeile in der Tabelle angezeigt.

Fügen Sie eine Seite zur Erstellung einer Besprechungsanfrage hinzu

Erstellen Sie als Nächstes eine Seite, die ein Formular enthält, das Endbenutzer zum Erstellen von Besprechungsanfragen verwenden. Sie fügen außerdem eine Schaltfläche zum Absenden hinzu, die den Datensatz in der `MeetingRequests` Entität erstellt und den Endbenutzer dann zurück zur `MeetingRequestsDashboard` Seite navigiert.

Um eine Seite zur Erstellung einer Besprechungsanfrage hinzuzufügen

1. Wählen Sie im oberen Banner die Option Seiten aus.
2. Wählen Sie in der linken Seitenleiste + Hinzufügen.
3. Wählen Sie im Eigenschaftenfenster auf der rechten Seite das Stiftsymbol aus und benennen Sie die Seite in um. **`CreateMeetingRequest`**

Nachdem die Seite hinzugefügt wurde, fügen Sie der Seite ein Formular hinzu, das Endbenutzer zur Eingabe von Informationen verwenden, um eine Besprechungsanfrage in der `MeetingRequests` Entität zu erstellen. App Studio bietet eine Methode zum Generieren eines Formulars aus einer vorhandenen Entität. Dabei werden die Formularfelder anhand der Felder der Entität automatisch ausgefüllt und außerdem eine Schaltfläche zum Absenden generiert, mit der ein Datensatz in der Entität mit den Formulareingaben erstellt wird.

Um automatisch ein Formular aus einer Entität auf der Seite zur Erstellung von Besprechungsanfragen zu generieren

1. Suchen Sie im rechten Komponentenmenü nach der Komponente Formular und ziehen Sie sie auf die Arbeitsfläche.
2. Wählen Sie Formular generieren aus.
3. Wählen Sie in der Dropdownliste die `MeetingRequests` Entität aus.
4. Wählen Sie Generate (Erstellen).
5. Wählen Sie auf der Leinwand die Schaltfläche „Senden“, um sie auszuwählen.
6. Wählen Sie im Eigenschaftenbereich auf der rechten Seite im Bereich Trigger die Option + Hinzufügen.
7. Wählen Sie „Navigieren“.
8. Ändern Sie im Eigenschaftenbereich auf der rechten Seite den Namen der Aktion in einen aussagekräftigen Namen, z. B. **`Navigate to MeetingRequestsDashboard`**
9. Ändern Sie den Navigationstyp in Seite. Wählen Sie in der Dropdownliste Navigieren zu die Option**`MeetingRequestsDashboard`**.

Da wir nun über eine Seite und ein Formular zur Erstellung von Besprechungsanfragen verfügen, möchten wir die Navigation von der `MeetingRequestsDashboard` Seite aus zu dieser Seite vereinfachen, sodass Endbenutzer, die das Dashboard überprüfen, problemlos Besprechungsanfragen erstellen können. Gehen Sie wie folgt vor, um auf der `MeetingRequestsDashboard` Seite eine Schaltfläche zu erstellen, mit der Sie zu der `CreateMeetingRequest` Seite navigieren können.

So fügen Sie eine Schaltfläche hinzu, von der aus **`MeetingRequestsDashboard`** Sie navigieren können **`CreateMeetingRequest`**

1. Wählen Sie im oberen Banner die Option Seiten aus.
2. Wählen Sie die `MeetingRequestsDashboard` Seite aus.
3. Suchen Sie im rechten Bedienfeld „Komponenten“ nach der Button-Komponente, ziehen Sie sie auf die Arbeitsfläche und platzieren Sie sie über der Tabelle.
4. Wählen Sie die neu hinzugefügte Schaltfläche, um sie auszuwählen.
5. Aktualisieren Sie im Eigenschaftenbereich auf der rechten Seite die folgenden Einstellungen:

a. Wählen Sie das Stiftsymbol aus, in das Sie die Schaltfläche umbenennen möchten**`createMeetingRequestButton`**.

b. Bezeichnung der Schaltfläche:**`Create Meeting Request`**. Dies ist der Name, den Endbenutzer sehen werden.

c. Wählen Sie in der Dropdownliste Symbol die Option + Plus aus.

d. Erstellen Sie einen Trigger, der den Endbenutzer zur Seite navigiert: `MeetingRequestsDashboard`

   1. Wählen Sie im Bereich Trigger die Option + Hinzufügen aus.
   2. Wählen Sie unter Aktionstyp die Option Navigieren aus.
   3. Wählen Sie den Trigger aus, den Sie gerade erstellt haben, um ihn zu konfigurieren.
   4. Geben Sie im Feld Aktionsname einen beschreibenden Namen ein, z. B. **`NavigateToCreateMeetingRequest`**
   5. Wählen Sie in der Dropdownliste Navigationstyp die Option Seite aus.
   6. Wählen Sie im Drop-down-Menü Navigieren zu die `CreateMeetingRequest` Seite aus.

Schritt 4: Zeigen Sie eine Vorschau der Anwendung an

Sie können eine Vorschau einer Anwendung in App Studio anzeigen, um zu sehen, wie sie Benutzern angezeigt wird. Sie können ihre Funktionalität auch testen, indem Sie sie verwenden und die Protokolle in einem Debug-Bereich überprüfen.

Die Anwendungsvorschauumgebung unterstützt die Anzeige von Live-Daten nicht. Sie unterstützt auch nicht die Verbindung mit externen Ressourcen über Konnektoren, z. B. Datenquellen. Stattdessen können Sie Beispieldaten und simulierte Ausgaben verwenden, um die Funktionalität zu testen.

Um eine Vorschau Ihrer App zum Testen anzuzeigen

1. Wählen Sie in der oberen rechten Ecke des App Builders die Option Vorschau aus.
2. Interagiere mit der `MeetingRequestsDashboard` Seite und teste die Tabelle, das Formular und die Schaltflächen.

Schritt 5: Veröffentlichen Sie die Anwendung in der Testumgebung

Nachdem Sie Ihre Anwendung erstellt, konfiguriert und getestet haben, ist es an der Zeit, sie in der Testumgebung zu veröffentlichen, um abschließende Tests durchzuführen und sie dann für Benutzer freizugeben.

Um Ihre App in der Testumgebung zu veröffentlichen

1. Wählen Sie in der oberen rechten Ecke des App Builders die Option Veröffentlichen aus.
2. Fügen Sie eine Versionsbeschreibung für die Testumgebung hinzu.
3. Überprüfen Sie das Kontrollkästchen für das SLA und aktivieren Sie es.
4. Wählen Sie Starten. Die Veröffentlichung kann bis zu 15 Minuten dauern.
5. (Optional) Wenn Sie bereit sind, können Sie anderen Zugriff gewähren, indem Sie „Teilen“ wählen und den Anweisungen folgen.

> **Note**
>
> Um Apps zu teilen, muss ein Administrator Endbenutzergruppen erstellt haben.

Wählen Sie nach dem Testen erneut Veröffentlichen, um die Anwendung in die Produktionsumgebung hochzustufen. Weitere Informationen zu den verschiedenen Anwendungsumgebungen finden Sie unterAnwendungsumgebungen.

Nächste Schritte

Nachdem Sie Ihre erste App erstellt haben, gehen Sie wie folgt vor:

1. Erstellen Sie die Tutorial-App weiter. Nachdem Sie nun Daten, einige Seiten und eine Automatisierung konfiguriert haben, können Sie weitere Seiten und Komponenten hinzufügen, um mehr über das Erstellen von Apps zu erfahren.
2. Weitere Informationen zum Erstellen von Apps finden Sie unterBuilder-Dokumentation. Insbesondere die folgenden Themen könnten hilfreich sein, um sich mit ihnen zu befassen:
   - Referenz zu Automation-Aktionen
   - Referenz zu den Komponenten
   - Interaktion mit Amazon Simple Storage Service mit Komponenten und Automatisierungen
   - Sicherheitsüberlegungen und Abhilfemaßnahmen

Darüber hinaus enthalten die folgenden Themen weitere Informationen zu den im Tutorial behandelten Konzepten:

- Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen
- Eine Entität in einer App Studio-App erstellen

Dokumentation für Administratoren

Die folgenden Themen enthalten Informationen, die Benutzern helfen sollen, die Verbindungen und den Zugriff auf Dienste von Drittanbietern, Benutzer und Rollen in App Studio verwalten.

Themen

- Zugriff und Rollen in App Studio verwalten
- Connect App Studio über Konnektoren mit anderen Diensten
- Löschen einer App Studio-Instanz

Zugriff und Rollen in App Studio verwalten

Eine der Aufgaben von Administratoren in App Studio ist die Verwaltung von Zugriff, Rollen und Berechtigungen. Die folgenden Themen enthalten Informationen zu den Rollen in App Studio und dazu, wie Sie Benutzer hinzufügen, entfernen oder ihre Rolle ändern können.

Der Zugriff auf AWS App Studio wird mithilfe von IAM Identity Center-Gruppen verwaltet. Um Benutzer zu Ihrer App Studio-Instanz hinzuzufügen, müssen Sie entweder:

- Fügen Sie sie einer vorhandenen IAM Identity Center-Gruppe hinzu, die zu App Studio hinzugefügt wurde.
- Fügen Sie sie einer neuen oder vorhandenen IAM Identity Center-Gruppe hinzu, die nicht zu App Studio hinzugefügt wurde, und fügen Sie sie dann zu App Studio hinzu.

Da Rollen auf Gruppen angewendet werden, sollten die IAM Identity Center-Gruppen die Zugriffsrechte (oder Rollen) darstellen, die Sie Mitgliedern der Gruppe zuweisen möchten. Weitere Informationen zu IAM Identity Center, einschließlich Informationen zur Verwaltung von Benutzern und Gruppen, finden Sie im IAM Identity Center-Benutzerhandbuch.

Rollen und Berechtigungen

In App Studio gibt es drei Rollen. Die folgende Liste enthält jede Rolle und ihre Beschreibung.

- Admin: Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber

hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.

- Builder: Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder die Anwendungen anderer Builder verwalten.
- App-Benutzer: App-Benutzer können auf veröffentlichte Apps zugreifen und diese verwenden, aber nicht auf Ihre App Studio-Instanz zugreifen, um Apps zu erstellen oder Ressourcen zu verwalten.

In App Studio werden Rollen Gruppen zugewiesen, daher wird jedem Mitglied einer hinzugefügten IAM Identity Center-Gruppe die Rolle zugewiesen, die der Gruppe zugewiesen ist.

Gruppen anzeigen

Führen Sie die folgenden Schritte aus, um die Gruppen anzuzeigen, die zu Ihrer App Studio-Instanz hinzugefügt wurden.

> **Note**
>
> Sie müssen ein Administrator sein, um Gruppen in Ihrer App Studio-Instanz anzeigen zu können.

Um Gruppen anzuzeigen, die zu Ihrer App Studio-Instanz hinzugefügt wurden

- Wählen Sie im Navigationsbereich im Abschnitt Verwalten die Option Rollen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Gruppen sowie die jeder Gruppe zugewiesenen Rollen angezeigt werden.

  Informationen zum Verwalten von Gruppen finden Sie unter Benutzer oder Gruppen hinzufügenDie Rolle einer Gruppe ändern, oderBenutzer oder Gruppen aus App Studio entfernen.

Benutzer oder Gruppen hinzufügen

Um Benutzer zu App Studio hinzuzufügen, müssen Sie sie zu einer IAM Identity Center-Gruppe hinzufügen und diese Gruppe zu App Studio hinzufügen. Führen Sie die folgenden Schritte aus, um

Benutzer zu App Studio hinzuzufügen, indem Sie IAM Identity Center-Gruppen hinzufügen und ihnen eine Rolle zuweisen.

> **Note**
>
> Sie müssen ein Administrator sein, um Benutzer zu Ihrer App Studio-Instanz hinzufügen zu können.

Um Benutzer oder Gruppen zu Ihrer App Studio-Instanz hinzuzufügen

1. Um Benutzer zu Ihrer App Studio-Instanz hinzuzufügen, müssen Sie sie entweder zu einer vorhandenen IAM Identity Center-Gruppe hinzufügen, die zu App Studio hinzugefügt wurde, oder Sie müssen eine neue IAM Identity Center-Gruppe erstellen, der Sie den neuen Benutzer hinzufügen und die neue Gruppe zu App Studio hinzufügen.

   Informationen zur Verwaltung von IAM Identity Center-Benutzern und -Gruppen finden Sie unter Identitäten in IAM Identity Center verwalten im Benutzerhandbuch.AWS IAM Identity Center
2. Wenn Sie Benutzer zu einer bestehenden IAM Identity Center-Gruppe hinzugefügt haben, die bereits zu App Studio hinzugefügt wurde, kann der neue Benutzer mit den angegebenen Berechtigungen auf App Studio zugreifen, nachdem er die Einrichtung seiner IAM Identity Center-Berechtigungen abgeschlossen hat. Wenn Sie eine neue IAM Identity Center-Gruppe erstellt haben, führen Sie die folgenden Schritte aus, um die Gruppe zu App Studio hinzuzufügen und den Mitgliedern der Gruppe eine Rolle zuzuweisen.
3. Wählen Sie im Navigationsbereich im Abschnitt Verwalten die Option Rollen aus.
4. Wählen Sie auf der Seite Rollen die Option + Gruppe hinzufügen aus. Dadurch wird ein Dialogfeld Gruppen hinzufügen geöffnet, in dem Sie Informationen über die Gruppe eingeben können.
5. Geben Sie im Dialogfeld Gruppen hinzufügen die folgenden Informationen ein:

   - Wählen Sie in der Dropdownliste die vorhandene IAM Identity Center-Gruppe aus.
   - Wählen Sie eine Rolle für die Gruppe aus.
     - Administrator: Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.

- Builder: Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder die Anwendungen anderer Builder verwalten.
- App-Benutzer: App-Benutzer können auf veröffentlichte Apps zugreifen und diese verwenden, aber nicht auf Ihre App Studio-Instanz zugreifen, um Apps zu erstellen oder Ressourcen zu verwalten.

6. Wählen Sie Zuweisen, um die Gruppe zu App Studio hinzuzufügen und ihren Mitgliedern die konfigurierte Rolle zuzuweisen.

Die Rolle einer Gruppe ändern

Gehen Sie wie folgt vor, um die einer Gruppe in App Studio zugewiesene Rolle zu ändern. Wenn Sie die Rolle einer Gruppe ändern, ändert sich die Rolle jedes Mitglieds in dieser Gruppe.

> **Note**
>
> Sie müssen ein Administrator sein, um die Rolle einer Gruppe in App Studio ändern zu können.

Um die Rolle einer Gruppe zu ändern

1. Wählen Sie im Navigationsbereich im Abschnitt Verwalten die Option Rollen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Gruppen sowie die jeder Gruppe zugewiesenen Rollen angezeigt werden.
2. Wählen Sie das Ellipsensymbol (...) und wählen Sie Rolle ändern.
3. Wählen Sie im Dialogfeld Rolle ändern eine neue Rolle für die Gruppe aus:

   - Administrator: Administratoren können Benutzer und Gruppen in App Studio verwalten, Konnektoren hinzufügen und verwalten sowie von Buildern erstellte Anwendungen verwalten. Darüber hinaus verfügen Benutzer mit der Administratorrolle über alle in der Builder-Rolle enthaltenen Berechtigungen.
   - Builder: Builder können Anwendungen erstellen und erstellen. Builder können keine Benutzer oder Gruppen verwalten, Connector-Instanzen hinzufügen oder bearbeiten oder die Anwendungen anderer Builder verwalten.

- App-Benutzer: App-Benutzer können auf veröffentlichte Apps zugreifen und diese verwenden, aber nicht auf Ihre App Studio-Instanz zugreifen, um Apps zu erstellen oder Ressourcen zu verwalten.

4. Wählen Sie Ändern, Rolle der Gruppe ändern aus.

Benutzer oder Gruppen aus App Studio entfernen

Sie können eine IAM Identity Center-Gruppe nicht aus App Studio entfernen. Wenn Sie die folgenden Anweisungen ausführen, wird stattdessen die Rolle der Gruppe auf App-Benutzer herabgestuft. Mitglieder der Gruppe können weiterhin auf veröffentlichte App Studio-Apps zugreifen.

Um jeglichen Zugriff auf App Studio und seine Apps zu entfernen, müssen Sie entweder die IAM Identity Center-Gruppe oder die Benutzer in der AWS IAM Identity Center Konsole löschen. Informationen zur Verwaltung von IAM Identity Center-Benutzern und -Gruppen finden Sie im Benutzerhandbuch unter Identitäten in IAM Identity Center verwalten.AWS IAM Identity Center

> **Note**
>
> Sie müssen ein Administrator sein, um den Zugriff einer Gruppe in App Studio herabstufen zu können.

Um eine Gruppe zu entfernen

1. Wählen Sie im Navigationsbereich im Abschnitt Verwalten die Option Rollen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Gruppen sowie die jeder Gruppe zugewiesenen Rollen angezeigt werden.
2. Wählen Sie das Ellipsensymbol (...) und wählen Sie Rolle widerrufen.
3. Wählen Sie im Dialogfeld „Rolle widerrufen“ die Option „Widerrufen“, um die Rolle der Gruppe auf „App-Benutzer“ herabzustufen.

Connect App Studio über Konnektoren mit anderen Diensten

Ein Connector ist eine Verbindung zwischen App Studio und anderen AWS Diensten wie AWS Lambda Amazon Redshift oder Diensten von Drittanbietern. Sobald ein Connector erstellt und konfiguriert ist, können Builder ihn und die Ressourcen, die er mit App Studio verbindet, in ihren Anwendungen verwenden.

Nur Benutzer mit der Administratorrolle können Connectors erstellen, verwalten oder löschen.

Themen

- Connect zu AWS Diensten her
- Connect zu Diensten von Drittanbietern her
- Konnektoren anzeigen, bearbeiten und löschen

Connect zu AWS Diensten her

Themen

- Connect zu Amazon Redshift her
- Connect zu Amazon DynamoDB her
- Verbinden mit AWS Lambda
- Connect zu Amazon Simple Storage Service (Amazon S3) her
- Connect zu Amazon Aurora her
- Connect zu Amazon Bedrock her
- Connect zu Amazon Simple Email Service her
- Stellen Sie mithilfe des Connectors für andere AWS Dienste eine Verbindung zu Diensten her AWS
- Verwenden Sie verschlüsselte Datenquellen mit CMKs

Connect zu Amazon Redshift her

Um App Studio mit Amazon Redshift zu verbinden, damit Builder auf Amazon Redshift Redshift-Ressourcen zugreifen und diese in Anwendungen verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: Amazon Redshift Redshift-Ressourcen erstellen und konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon Redshift Redshift-Berechtigungen
3. Schritt 3: Amazon Redshift Redshift-Connector erstellen

Schritt 1: Amazon Redshift Redshift-Ressourcen erstellen und konfigurieren

Gehen Sie wie folgt vor, um Amazon Redshift Redshift-Ressourcen für die Verwendung mit App Studio zu erstellen und zu konfigurieren.

So richten Sie Amazon Redshift für die Verwendung mit App Studio ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Redshift Redshift-Konsole unter https://console.aws.amazon.com/redshiftv2/.

   Wir empfehlen, den in Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen erstellten Administratorbenutzer zu verwenden.
2. Erstellen Sie ein Redshift Serverless Data Warehouse oder einen bereitgestellten Cluster. Weitere Informationen finden Sie unter Erstellen eines Data Warehouse mit Redshift Serverless oder Erstellen eines Clusters im Amazon Redshift Redshift-Benutzerhandbuch.
3. Sobald die Bereitstellung abgeschlossen ist, wählen Sie Daten abfragen, um den Abfrage-Editor zu öffnen. Connect zu Ihrer Datenbank her.
4. Ändern Sie die folgenden Einstellungen:

   1. Stellen Sie den Schalter Isolierte Sitzung auf`OFF`. Dies ist erforderlich, damit Sie Datenänderungen sehen können, die von anderen Benutzern vorgenommen wurden, z. B. von einer laufenden App Studio-Anwendung aus.
   2. Wählen Sie das Zahnradsymbol. Klicken Sie auf Account settings (Kontoeinstellungen). Erhöhen Sie die maximale Anzahl gleichzeitiger Verbindungen auf`10`. Dies ist die Obergrenze für die Anzahl der Abfrage-Editor-Sitzungen, die eine Verbindung zu einer Amazon Redshift Redshift-Datenbank herstellen können. Sie gilt nicht für andere Clients wie App Studio-Anwendungen.
5. Erstellen Sie Ihre Datentabellen unter dem `public` Schema. `INSERT`Talle Anfangsdaten in diesen Tabellen.
6. Führen Sie die folgenden Befehle im Abfrage-Editor aus:

   Der folgende Befehl erstellt einen Datenbankbenutzer und verbindet ihn mit einer IAM-Rolle namens*`AppBuilderDataAccessRole`*, die von App Studio verwendet wird. Sie werden die IAM-Rolle in einem späteren Schritt erstellen, und der Name hier muss mit dem Namen übereinstimmen, der dieser Rolle zugewiesen wurde.

   ```
   CREATE USER "IAMR:AppBuilderDataAccessRole" WITH PASSWORD DISABLE;
   ```

Der folgende Befehl gewährt App Studio alle Berechtigungen für alle Tabellen.

> **Note**
>
> Um optimale Sicherheitsvorkehrungen zu treffen, sollten Sie die Berechtigungen hier auf die minimal erforderlichen Berechtigungen für die entsprechenden Tabellen beschränken. Weitere Informationen zu dem GRANT Befehl finden Sie unter GRANT im Amazon Redshift Database Developer Guide.

```
GRANT ALL ON ALL TABLES IN SCHEMA public to "IAMR:AppBuilderDataAccessRole";
```

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon Redshift Redshift-Berechtigungen

Um Amazon Redshift Redshift-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Richtlinie und -Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Richtlinie steuert den Umfang der Daten, die Builder verwenden können, und steuert, welche Operationen für diese Daten ausgeführt werden können, z. B. Erstellen, Lesen, Aktualisieren oder Löschen. Die IAM-Richtlinie wird dann an eine IAM-Rolle angehängt, die von App Studio verwendet wird.

Wir empfehlen, mindestens eine IAM-Rolle pro Dienst und Richtlinie zu erstellen. Wenn Builder beispielsweise zwei Anwendungen erstellen, die von unterschiedlichen Tabellen in Amazon Redshift unterstützt werden, sollte ein Administrator zwei IAM-Richtlinien und -Rollen erstellen, eine für jede der Tabellen in Amazon Redshift.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit den entsprechenden Amazon Redshift Redshift-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

Um eine IAM-Richtlinie mit entsprechenden Amazon Redshift Redshift-Berechtigungen zu erstellen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Richtlinien zu erstellen. Wir empfehlen, den in erstellten Administratorbenutzer zu verwenden. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie ein JSON-Richtliniendokument ein oder fügen Sie es ein. Die folgenden Registerkarten enthalten Beispielrichtlinien für bereitgestelltes und serverloses Amazon Redshift.

> **Note**
>
> Die folgenden Richtlinien gelten für alle Amazon Redshift Redshift-Ressourcen, die den Platzhalter () * verwenden. Aus Sicherheitsgründen sollten Sie den Platzhalter durch den Amazon-Ressourcennamen (ARN) der Ressourcen ersetzen, die Sie mit App Studio verwenden möchten.

Provisioned

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "ProvisionedRedshiftForAppStudio",
      "Effect": "Allow",
         "Action": [
            "redshift:DescribeClusters",
            "redshift:GetClusterCredentialsWithIAM",
            "redshift-data:ListDatabases",
            "redshift-data:ListTables",
            "redshift-data:DescribeTable",
            "redshift-data:DescribeStatement",
            "redshift-data:ExecuteStatement",
            "redshift-data:GetStatementResult"
         ],
         "Resource": "*"
      }
```

```
   ]
}
```

Serverless

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "ServerlessRedshiftForAppStudio",
      "Effect": "Allow",
         "Action": [
            "redshift-serverless:ListNamespaces",
            "redshift-serverless:GetCredentials",
            "redshift-serverless:ListWorkgroups",
            "redshift-data:ListDatabases",
            "redshift-data:ListTables",
            "redshift-data:DescribeTable",
            "redshift-data:DescribeStatement",
            "redshift-data:ExecuteStatement",
            "redshift-data:GetStatementResult"
         ],
         "Resource": "*"
      }
   ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen wie **`RedshiftServerlessForAppStudio`** oder **`RedshiftProvisionedForAppStudio`** und eine Beschreibung (optional) ein.
8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon Redshift Redshift-Ressourcen zu gewähren

Erstellen Sie nun eine IAM-Rolle, die die zuvor erstellte Richtlinie verwendet. App Studio verwendet diese Richtlinie, um Zugriff auf die konfigurierten Amazon Redshift Redshift-Ressourcen zu erhalten.

So erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon Redshift Redshift-Ressourcen zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den in erstellten Administratorbenutzer zu verwenden. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich die Option Rollen aus
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
5. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
```

```
    ]
}
```

Wählen Sie Weiter aus.

6. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie im vorherigen Schritt erstellt haben (**`RedshiftServerlessForAppStudio`**oder**`RedshiftProvisionedForAppStudio`**), und wählen Sie sie aus. Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

7. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.

   > **Important**
   >
   > Der Rollenname hier muss mit dem Rollennamen übereinstimmen, der im `GRANT` Befehl in Schritt 1: Amazon Redshift Redshift-Ressourcen erstellen und konfigurieren (*`AppBuilderDataAccessRole`*) verwendet wurde.

8. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

9. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den Amazon Redshift-Connector in App Studio erstellen.

Schritt 3: Amazon Redshift Redshift-Connector erstellen

Nachdem Sie Ihre Amazon Redshift-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit Amazon Redshift zu verbinden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können.

So erstellen Sie einen Connector für Amazon Redshift

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie den Amazon Redshift Redshift-Connector.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon Redshift Redshift-Ressourcen zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre Amazon Redshift Redshift-Ressourcen befinden.
   - Berechnungstyp: Wählen Sie aus, ob Sie Amazon Redshift Serverless oder einen bereitgestellten Cluster verwenden.
   - Auswahl eines Clusters oder einer Arbeitsgruppe: Wenn Bereitgestellt ausgewählt ist, wählen Sie den Cluster aus, den Sie mit App Studio verbinden möchten. Wenn Serverless ausgewählt ist, wählen Sie die Arbeitsgruppe aus.
   - Datenbankauswahl: Wählen Sie die Datenbank aus, die Sie mit App Studio verbinden möchten.
   - Verfügbare Tabellen: Wählen Sie die Tabellen aus, die Sie mit App Studio verbinden möchten.
6. Wählen Sie Weiter aus. Überprüfen Sie die Verbindungsinformationen und wählen Sie Erstellen.
7. Der neu erstellte Connector wird in der Connectorliste angezeigt.

Connect zu Amazon DynamoDB her

Um App Studio mit DynamoDB zu verbinden, damit Builder auf DynamoDB-Ressourcen in Anwendungen zugreifen und diese verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: DynamoDB-Ressourcen erstellen und konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden DynamoDB-Berechtigungen
3. DynamoDB-Konnektor erstellen

Schritt 1: DynamoDB-Ressourcen erstellen und konfigurieren

Gehen Sie wie folgt vor, um DynamoDB-Ressourcen für die Verwendung mit App Studio zu erstellen und zu konfigurieren.

So richten Sie DynamoDB für die Verwendung mit App Studio ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die DynamoDB-Konsole unter. https://console.aws.amazon.com/dynamodb/

   Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Tables (Tabellen) aus.
3. Wählen Sie Create table (Tabelle erstellen) aus.
4. Geben Sie einen Namen und Schlüssel für Ihre Tabelle ein.
5. Wählen Sie Create table (Tabelle erstellen) aus.
6. Nachdem Ihre Tabelle erstellt wurde, fügen Sie ihr einige Elemente hinzu, sodass sie angezeigt werden, sobald die Tabelle mit App Studio verbunden ist.

   a. Wählen Sie Ihre Tabelle aus, wählen Sie Aktionen und dann Elemente erkunden aus.

   b. Wählen Sie unter Zurückgesendete Artikel die Option Artikel erstellen aus.

   c. (Optional): Wählen Sie Neues Attribut hinzufügen aus, um Ihrer Tabelle weitere Attribute hinzuzufügen.

   d. Geben Sie Werte für jedes Attribut ein und wählen Sie Element erstellen.

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden DynamoDB-Berechtigungen

Um DynamoDB-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Richtlinie und -Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Richtlinie steuert den Umfang der Daten, die Builder verwenden können, und steuert, welche Operationen für diese Daten ausgeführt werden können, z. B. Erstellen, Lesen, Aktualisieren oder Löschen. Die IAM-Richtlinie wird dann an eine IAM-Rolle angehängt, die von App Studio verwendet wird.

Wir empfehlen, mindestens eine IAM-Rolle pro Dienst und Richtlinie zu erstellen. Wenn Builder beispielsweise zwei Anwendungen erstellen, die von denselben Tabellen in DynamoDB unterstützt werden, eine, die nur Lesezugriff benötigt, und eine, die Lesen, Erstellen, Aktualisieren und Löschen erfordert, sollte ein Administrator zwei IAM-Rollen erstellen, eine mit Nur-Lese-Rechten und eine mit vollen CRUD-Berechtigungen für die entsprechenden Tabellen in DynamoDB.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit entsprechenden DynamoDB-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

So erstellen Sie eine IAM-Richtlinie mit entsprechenden DynamoDB-Berechtigungen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Richtlinien zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie ein JSON-Richtliniendokument ein oder fügen Sie es ein. Die folgenden Registerkarten enthalten Beispielrichtlinien für schreibgeschützten und vollständigen Zugriff auf DynamoDB-Tabellen sowie Beispiele für Richtlinien, die AWS KMS Berechtigungen für DynamoDB-Tabellen enthalten, die mit einem AWS KMS vom Kunden verwalteten Schlüssel (CMK) verschlüsselt sind.

> **Note**
>
> Die folgenden Richtlinien gelten für alle DynamoDB-Ressourcen, die den Platzhalter () verwenden. * Aus Sicherheitsgründen sollten Sie den Platzhalter durch den Amazon-Ressourcennamen (ARN) der Ressourcen ersetzen, die Sie mit App Studio verwenden möchten.

Read only

Die folgende Richtlinie gewährt Lesezugriff auf die konfigurierten DynamoDB-Ressourcen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "ReadOnlyDDBForAppStudio",
      "Effect": "Allow",
         "Action": [
            "dynamodb:ListTables",
            "dynamodb:DescribeTable",
            "dynamodb:PartiQLSelect"
         ],
         "Resource": "*"
      }
   ]
}
```

Full access

Die folgende Richtlinie gewährt Erstellungs-, Lese-, Aktualisierungs- und Löschzugriff auf die konfigurierten DynamoDB-Ressourcen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "FullAccessDDBForAppStudio",
      "Effect": "Allow",
         "Action": [
```

```
            "dynamodb:ListTables",
            "dynamodb:DescribeTable",
            "dynamodb:PartiQLSelect",
            "dynamodb:PartiQLInsert",
            "dynamodb:PartiQLUpdate",
            "dynamodb:PartiQLDelete"
         ],
         "Resource": "*"
      }
   ]
}
```

Read only - KMS encrypted

Die folgende Richtlinie gewährt Lesezugriff auf die konfigurierten verschlüsselten DynamoDB-Ressourcen, indem sie Berechtigungen bereitstellt AWS KMS . Sie müssen den ARN durch den ARN des AWS KMS Schlüssels ersetzen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "ReadOnlyDDBForAppStudio",
      "Effect": "Allow",
         "Action": [
            "dynamodb:ListTables",
            "dynamodb:DescribeTable",
            "dynamodb:PartiQLSelect"
         ],
         "Resource": "*"
      },
      {
      "Sid": "KMSPermissionsForEncryptedTable",
      "Effect": "Allow",
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey"
         ],
         "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },

   ]
```

```
}
```

Full access - KMS encrypted

Die folgende Richtlinie gewährt Lesezugriff auf die konfigurierten verschlüsselten DynamoDB-Ressourcen, indem sie Berechtigungen bereitstellt AWS KMS . Sie müssen den ARN durch den ARN des AWS KMS Schlüssels ersetzen.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
      "Sid": "ReadOnlyDDBForAppStudio",
      "Effect": "Allow",
         "Action": [
           "dynamodb:ListTables",
           "dynamodb:DescribeTable",
           "dynamodb:PartiQLSelect",
           "dynamodb:PartiQLInsert",
           "dynamodb:PartiQLUpdate",
           "dynamodb:PartiQLDelete"
         ],
         "Resource": "*"
      },
      {
      "Sid": "KMSPermissionsForEncryptedTable",
      "Effect": "Allow",
         "Action": [
            "kms:Decrypt",
            "kms:DescribeKey"
         ],
         "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      },

   ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen wie **`ReadOnlyDDBForAppStudio`** oder **`FullAccessDDBForAppStudio`** und eine Beschreibung (optional) ein.

8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf DynamoDB-Ressourcen zu gewähren

Erstellen Sie nun eine IAM-Rolle, die die zuvor erstellte Richtlinie verwendet. App Studio verwendet diese Richtlinie, um Zugriff auf die konfigurierten DynamoDB-Ressourcen zu erhalten.

So erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf DynamoDB-Ressourcen zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie im vorherigen Schritt erstellt haben (**`ReadOnlyDDBForAppStudio`**oder**`FullAccessDDBForAppStudio`**), und wählen Sie sie aus. Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.
7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

8. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den DynamoDB-Connector in App Studio erstellen.

DynamoDB-Konnektor erstellen

Nachdem Sie Ihre DynamoDB-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit DynamoDB zu verbinden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können.

So erstellen Sie einen Konnektor für DynamoDB

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie Amazon DynamoDB aus der Liste der Konnektortypen aus.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren DynamoDB-Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren DynamoDB-Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf DynamoDB-Ressourcen zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre DynamoDB-Ressourcen befinden.
   - Verfügbare Tabellen: Wählen Sie die Tabellen aus, die Sie mit App Studio verbinden möchten.
6. Wählen Sie Weiter aus. Überprüfen Sie die Verbindungsinformationen und wählen Sie Erstellen.
7. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Verbinden mit AWS Lambda

Um App Studio mit Lambda zu verbinden, damit Builder auf Lambda-Ressourcen in Anwendungen zugreifen und diese verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: Lambda-Funktionen erstellen und konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Lambda-Ressourcen zu gewähren
3. Schritt 3: Lambda-Konnektor erstellen

Schritt 1: Lambda-Funktionen erstellen und konfigurieren

Wenn Sie noch keine Lambda-Funktionen haben, müssen Sie diese zuerst erstellen. Weitere Informationen zum Erstellen von Lambda-Funktionen finden Sie im AWS Lambda Developer Guide.

Schritt 2: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Lambda-Ressourcen zu gewähren

Um Lambda-Ressourcen mit App Studio verwenden zu können, müssen Administratoren eine IAM-Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Rolle steuert die Ressourcen oder Betriebsabläufe, auf die Anwendungen von Lambda aus zugreifen können.

Wir empfehlen, mindestens eine IAM-Rolle pro Service und Richtlinie zu erstellen.

So erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Lambda-Ressourcen zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach den Richtlinien, die die entsprechenden Berechtigungen für die Rolle gewähren, und wählen Sie sie aus. Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt. Für Lambda könnten Sie erwägen, die `AWSLambdaRole` Richtlinie hinzuzufügen, die Berechtigungen zum Aufrufen von Lambda-Funktionen gewährt.

   Weitere Informationen zur Verwendung von IAM-Richtlinien mit Lambda, einschließlich einer Liste der verwalteten Richtlinien und ihrer Beschreibungen, finden Sie unter Identity and Access Management for AWS Lambda im AWS Lambda Developer Guide.

   Wählen Sie Weiter aus.

6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung an.
7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

8. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den Lambda-Connector in App Studio erstellen.

Schritt 3: Lambda-Konnektor erstellen

Nachdem Sie Ihre Lambda-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit Lambda zu verbinden.

> **Note**
>
> Sie benötigen die Admin-Rolle in App Studio, um Connectors erstellen zu können.

Um einen Konnektor für Lambda zu erstellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie Andere AWS Dienste aus der Liste der Konnektortypen aus.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Lambda-Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Lambda-Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Lambda-Ressourcen zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Service: Wählen Sie Lambda.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre Lambda-Ressourcen befinden.
6. Wählen Sie Erstellen aus.
7. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Connect zu Amazon Simple Storage Service (Amazon S3) her

Gehen Sie wie folgt vor, um App Studio mit Amazon S3 zu verbinden, damit Builder auf Amazon S3 S3-Ressourcen in Anwendungen zugreifen und diese verwenden können:

1. Schritt 1: Amazon S3 S3-Ressourcen erstellen und konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon S3 S3-Berechtigungen
3. Schritt 3: Amazon S3 S3-Connector erstellen

Nachdem Sie die Schritte abgeschlossen und den Connector mit den entsprechenden Berechtigungen erstellt haben, können Builder den Connector verwenden, um Apps zu erstellen, die mit Amazon S3 S3-Ressourcen interagieren. Weitere Informationen zur Interaktion mit Amazon S3 in App Studio-Apps finden Sie unterInteraktion mit Amazon Simple Storage Service mit Komponenten und Automatisierungen.

Schritt 1: Amazon S3 S3-Ressourcen erstellen und konfigurieren

Abhängig von den Anforderungen Ihrer App und Ihren vorhandenen Ressourcen müssen Sie möglicherweise einen Amazon S3 S3-Bucket erstellen, in den Apps schreiben und aus dem sie lesen können. Informationen zum Erstellen von Amazon S3-Ressourcen, einschließlich Buckets, finden Sie unter Erste Schritte mit Amazon S3 im Amazon Simple Storage Service-Benutzerhandbuch.

Um die S3-Upload Komponente in Ihren Apps zu verwenden, müssen Sie allen Amazon S3 S3-Buckets, in die Sie hochladen möchten, eine CORS-Konfiguration (Cross-Origin Resource Sharing) hinzufügen. Die CORS-Konfiguration gibt App Studio die Erlaubnis, Objekte in den Bucket zu verschieben. Das folgende Verfahren beschreibt, wie Sie mithilfe der Konsole eine CORS-Konfiguration zu einem Amazon S3 S3-Bucket hinzufügen. Weitere Informationen zu CORS und seiner Konfiguration finden Sie unter Using Cross-Origin Resource Sharing (CORS) im Amazon Simple Storage Service-Benutzerhandbuch.

Um eine CORS-Konfiguration zu einem Amazon S3 S3-Bucket in der Konsole hinzuzufügen

1. Navigieren Sie zu Ihrem Bucket im https://console.aws.amazon.com/s3/.
2. Wählen Sie die Registerkarte Berechtigungen.
3. Wählen Sie unter Cross-Origin Resource Sharing (CORS) die Option Bearbeiten aus.
4. Fügen Sie das folgende Snippet hinzu:

```
[
    {
        "AllowedHeaders": [
            "*"
        ],
        "AllowedMethods": [
            "PUT",
            "POST"
        ],
        "AllowedOrigins": [
            "*"
        ],
```

```
        "ExposeHeaders": []
    }
]
```

5. Wählen Sie Änderungen speichern aus.

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon S3 S3-Berechtigungen

Um Amazon S3 S3-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Richtlinie und -Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Richtlinie steuert den Umfang der Daten, die Builder verwenden können, und steuert, welche Operationen für diese Daten ausgeführt werden können, z. B. Erstellen, Lesen, Aktualisieren oder Löschen. Die IAM-Richtlinie wird dann an eine IAM-Rolle angehängt, die von App Studio verwendet wird.

Wir empfehlen, mindestens eine IAM-Rolle pro Dienst und Richtlinie zu erstellen. Wenn Builder beispielsweise zwei Anwendungen erstellen, die von unterschiedlichen Buckets in Amazon S3 unterstützt werden, sollte ein Administrator zwei IAM-Richtlinien und -Rollen erstellen, eine für jeden der Buckets.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit entsprechenden Amazon S3 S3-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

Um eine IAM-Richtlinie mit entsprechenden Amazon S3 S3-Berechtigungen zu erstellen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Richtlinien zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie ein JSON-Richtliniendokument ein oder fügen Sie es ein. Die folgenden Tabs enthalten Beispielrichtlinien für schreibgeschützten und vollen Zugriff auf Amazon S3 S3-Ressourcen.

> **Note**
>
> Die folgenden Richtlinien gelten für alle Amazon S3 S3-Ressourcen, die den Platzhalter (*) verwenden. Aus Sicherheitsgründen sollten Sie den Platzhalter durch den Amazon-Ressourcennamen (ARN) der Ressourcen, wie Buckets oder Ordner, ersetzen, die Sie mit App Studio verwenden möchten.

Read only

Die folgende Richtlinie gewährt nur Lesezugriff (Abrufen und Auflisten) auf die konfigurierten Amazon S3 S3-Buckets oder -Ordner.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "S3ReadOnlyForAppStudio",
         "Effect": "Allow",
         "Action": [
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource": "*"
      }
   ]
}
```

Full access

Die folgende Richtlinie gewährt vollen Zugriff (Put, Get, List und Delete) auf die konfigurierten Amazon S3 S3-Buckets oder -Ordner.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "S3FullAccessForAppStudio",
         "Effect": "Allow",
         "Action": [
```

```
            "s3:PutObject",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:DeleteObject"
         ],
         "Resource": "*"
      }
   ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen, z. B.**`AWSAppStudioS3FullAccess`**, und eine Beschreibung (optional) ein.
8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon S3 S3-Ressourcen zu gewähren

Um Amazon S3 S3-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Rolle steuert den Umfang der Daten, die Builder verwenden können, und steuert, welche Operationen für diese Daten aufgerufen werden können, z. B. Erstellen, Lesen, Aktualisieren oder Löschen.

Wir empfehlen, mindestens eine IAM-Rolle pro Service und Richtlinie zu erstellen.

Um eine IAM-Rolle zu erstellen, um App Studio Zugriff auf Amazon S3 S3-Ressourcen zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

- Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
- *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie im vorherigen Schritt erstellt haben (**S3ReadOnlyForAppStudio**oder**S3FullAccessForAppStudio**), und wählen Sie sie aus. Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.

7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:
   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`
8. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, um im nächsten Schritt den Amazon S3-Connector in App Studio zu erstellen.

Schritt 3: Amazon S3 S3-Connector erstellen

Nachdem Sie Ihre Amazon S3-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit Amazon S3 zu verbinden.

> **Note**
>
> Sie benötigen die Admin-Rolle in App Studio, um Connectors erstellen zu können.

Um einen Connector für Amazon S3 zu erstellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie den Amazon S3 S3-Connector.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Amazon S3 S3-Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Amazon S3 S3-Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon S3 S3-Ressourcen zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.

- Region: Wählen Sie die AWS Region aus, in der sich Ihre Amazon S3 S3-Ressourcen befinden.

6. Wählen Sie Erstellen aus.
7. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Connect zu Amazon Aurora her

Um App Studio mit Aurora zu verbinden, damit Builder auf Aurora-Ressourcen in Anwendungen zugreifen und diese nutzen können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: Aurora-Ressourcen erstellen und konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Aurora-Berechtigungen
3. Schritt 3: Aurora-Connector in App Studio erstellen

App Studio unterstützt die folgenden Aurora-Versionen:

- Aurora MySQL Serverlos V1:5.72
- Aurora PostgreSQL Serverless V1:11.18, 13.9
- Aurora MySQL Serverless V2:13.11 oder höher, 14.8 oder höher und 15.3 oder höher
- Aurora PostgreSQL Serverless V2:13.11 oder höher, 14.8 oder höher und 15.3 oder höher

Schritt 1: Aurora-Ressourcen erstellen und konfigurieren

Um Aurora-Datenbanken mit App Studio verwenden zu können, müssen Sie sie zunächst erstellen und entsprechend konfigurieren. App Studio unterstützt zwei Aurora-Datenbanktypen: Aurora PostgreSQL und Aurora MySQL. Einen Vergleich der Typen finden Sie unter Was ist der Unterschied zwischen MySQL und PostgreSQL? . Wählen Sie die entsprechende Registerkarte und folgen Sie den Anweisungen, um Aurora für die Verwendung mit App Studio-Apps einzurichten.

Aurora PostgreSQL

Gehen Sie wie folgt vor, um einen Aurora PostgreSQL-Datenbankcluster für die Verwendung mit App Studio zu erstellen und zu konfigurieren.

So richten Sie Aurora für die Verwendung mit App Studio ein

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
2. Wählen Sie Datenbank erstellen aus.
3. Wählen Sie Aurora (PostgreSQL-kompatibel).
4. Wählen Sie unter Verfügbare Versionen eine Version aus, die größer oder gleich Version `13.1114.8`, und ist. `15.3`
5. Geben Sie unter Einstellungen eine DB-Cluster-ID ein.
6. Wählen Sie unter Instanzkonfiguration Serverless v2 und wählen Sie eine entsprechende Kapazität aus.
7. Wählen Sie unter Konnektivität die Option RDS-Daten-API aktivieren aus.
8. Wählen Sie unter Datenbankauthentifizierung die Option IAM-Datenbankauthentifizierung aus.
9. Geben Sie unter Zusätzliche Konfiguration im Feld Ursprünglicher Datenbankname einen ersten Datenbanknamen für Ihre Datenbank ein.

Aurora MySQL

Gehen Sie wie folgt vor, um einen Aurora MySQL-Datenbankcluster für die Verwendung mit App Studio zu erstellen und zu konfigurieren.

Aurora MySQL unterstützt für Versionen, die Data API oder Serverless v1 unterstützen, keine Erstellung über die Benutzeroberfläche. Um einen Aurora MySQL-Cluster zu erstellen, der die Daten-API unterstützt, müssen Sie den verwenden AWS CLI.

> **Note**
>
> Um Aurora MySQL-Datenbanken mit App Studio verwenden zu können, müssen sie sich in einer Virtual Private Cloud (VPC) befinden. Weitere Informationen finden Sie unter Arbeiten mit einem DB-Cluster in einer VPC im Amazon Aurora Aurora-Benutzerhandbuch.

So richten Sie Aurora MySQL für die Verwendung mit App Studio ein

1. Falls erforderlich, installieren Sie das, AWS CLI indem Sie den Anweisungen unter Installieren oder Aktualisieren Sie auf die neueste Version von AWS CLI im AWS Command Line Interface Benutzerhandbuch folgen.
2. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon RDS-Konsole unter https://console.aws.amazon.com/rds/.
3. Wählen Sie in der linken Navigationsleiste Subnetzgruppen aus.
4. Wählen Sie Create DB subnet group (DB-Subnetzgruppe erstellen) aus.
5. Füllen Sie die Informationen aus und erstellen Sie die Subnetzgruppe. Weitere Informationen zu Subnetzgruppen und deren Verwendung VPCs finden Sie unter Arbeiten mit einem DB-Cluster in einer VPC im Amazon Aurora Aurora-Benutzerhandbuch.
6. Führen Sie den folgenden Befehl aus: AWS CLI

```
aws rds create-db-cluster --database-name db_name \
    --db-cluster-identifier db_cluster_identifier \
    --engine aurora-mysql \
    --engine-version 5.7.mysql_aurora.2.08.3 \
    --engine-mode serverless \
    --scaling-configuration
 MinCapacity=4,MaxCapacity=32,SecondsUntilAutoPause=1000,AutoPause=true \
    --master-username userName \
    --master-user-password userPass \
    --availability-zones us-west-2b us-west-2c \
    --db-subnet-group-name subnet-group-name
```

Ersetzen Sie die folgenden Felder:

- *db_name*Ersetzen Sie durch den gewünschten Datenbanknamen.
- Ersetzen Sie *db_cluster_identifier* durch den gewünschten Datenbank-Cluster-Bezeichner.
- (Optional) Ersetzen Sie die Zahlen im `scaling-configuration` Feld wie gewünscht.
- *userName*Ersetzen Sie es durch einen gewünschten Benutzernamen.
- *userPass*Durch ein gewünschtes Passwort ersetzen.
- Fügen Sie `availability-zones` unter die Availability Zones aus der Subnetzgruppe hinzu, die Sie erstellt haben.

- *subnet-group-name*Ersetzen Sie durch den Namen der Subnetzgruppe, die Sie erstellt haben.

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Aurora-Berechtigungen

Um Aurora-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Richtlinie erstellen und sie an eine IAM-Rolle anhängen, mit der App Studio Berechtigungen für den Zugriff auf die konfigurierten Ressourcen erteilt werden. Die IAM-Richtlinie und -Rolle steuern den Umfang der Daten, die Builder verwenden können, und welche Operationen für diese Daten aufgerufen werden können, z. B. Erstellen, Lesen, Aktualisieren oder Löschen.

Wir empfehlen, mindestens eine IAM-Rolle pro Service und Richtlinie zu erstellen.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit den entsprechenden Aurora-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

So erstellen Sie eine IAM-Richtlinie mit den entsprechenden Aurora-Berechtigungen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Ersetzen Sie das vorhandene Snippet durch das folgende Snippet und *111122223333* ersetzen Sie es durch die AWS Kontonummer, in der die Amazon Redshift- und Aurora-Ressourcen enthalten sind.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "BaselineAuroraForAppStudio",
            "Effect": "Allow",
            "Action": [
```

```
                "rds-data:ExecuteStatement",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:rds:*:111122223333:cluster:*",
                "arn:aws:secretsmanager:*:111122223333:secret:rds*"
            ]
        }
    ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen ein, z. B. **`Aurora_AppStudio`** und eine Beschreibung (optional).
8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Aurora-Ressourcen zu gewähren

Erstellen Sie nun eine IAM-Rolle, die die zuvor erstellte Richtlinie verwendet. App Studio verwendet diese Richtlinie, um Zugriff auf die konfigurierten Aurora-Ressourcen zu erhalten.

So erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Aurora-Ressourcen zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.

- *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie zuvor erstellt haben, und wählen Sie sie aus (**`Aurora_AppStudio`**). Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.
7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

8. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den Aurora-Connector in App Studio erstellen.

Schritt 3: Aurora-Connector in App Studio erstellen

Nachdem Sie Ihre Aurora-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit Aurora zu verbinden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können.

Um einen Konnektor für Aurora zu erstellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie den Amazon Aurora Aurora-Connector.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Aurora-Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Aurora-Anschluss ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Aurora-Ressourcen zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Geheimer ARN: Geben Sie den geheimen ARN des Datenbank-Clusters ein. Informationen darüber, wo Sie den geheimen ARN finden, finden Sie unter Details zu einem geheimen Schlüssel für einen DB-Cluster anzeigen im Amazon Aurora Aurora-Benutzerhandbuch.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre Aurora-Ressourcen befinden.
   - Datenbank-ARN: Geben Sie den ARN des Datenbank-Clusters ein. Der ARN befindet sich auf der Registerkarte Konfiguration des Datenbank-Clusters, ähnlich dem geheimen ARN.
   - Datenbanktyp: Wählen Sie den Datenbanktyp, MySQL oder PostgreSQL, der dem Datenbanktyp entspricht, in dem erstellt wurde. Schritt 1: Aurora-Ressourcen erstellen und konfigurieren

- Datenbankname: Geben Sie den Namen der Datenbank ein, den Sie auch auf der Registerkarte Konfiguration des Datenbank-Clusters finden.
- Verfügbare Tabellen: Wählen Sie die Tabellen aus, die Sie mit App Studio über diesen Connector verwenden möchten.

6. Wählen Sie Weiter, um die Entitätszuordnungen zu überprüfen oder zu definieren.
7. Wählen Sie Create, um den Aurora-Connector zu erstellen. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Connect zu Amazon Bedrock her

Um App Studio mit Amazon Bedrock zu verbinden, sodass Builder auf Amazon Bedrock zugreifen und es in Anwendungen verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: Amazon Bedrock-Modelle aktivieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon Bedrock-Berechtigungen
3. Schritt 3: Amazon Bedrock Connector erstellen

Schritt 1: Amazon Bedrock-Modelle aktivieren

Gehen Sie wie folgt vor, um Amazon Bedrock-Modelle zu aktivieren.

Um Amazon Bedrock-Modelle zu aktivieren

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die Amazon Bedrock-Konsole unter https://console.aws.amazon.com/bedrock/.
2. Wählen Sie im linken Navigationsbereich die Option Modellzugriff aus.
3. Aktivieren Sie die Modelle, die Sie verwenden möchten. Weitere Informationen finden Sie unter Zugriff auf Amazon Bedrock Foundation-Modelle verwalten.

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon Bedrock-Berechtigungen

Um Amazon Bedrock-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Richtlinie und -Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Richtlinie steuert, welche Ressourcen und welche Operationen für diese

Ressourcen aufgerufen werden können, z. B. `InvokeModel` Die IAM-Richtlinie wird dann an eine IAM-Rolle angehängt, die von App Studio verwendet wird.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit den entsprechenden Amazon Bedrock-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

Um eine IAM-Richtlinie mit den entsprechenden Amazon Bedrock-Berechtigungen zu erstellen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Richtlinien zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie ein JSON-Richtliniendokument ein oder fügen Sie es ein. Die folgende Beispielrichtlinie gilt für alle `InvokeModel` Amazon Bedrock-Ressourcen unter Verwendung des Platzhalters ()*.

   Aus Sicherheitsgründen sollten Sie den Platzhalter durch den Amazon-Ressourcennamen (ARN) der Ressourcen ersetzen, die Sie mit App Studio verwenden möchten.

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Sid": "BedrockAccessForAppStudio",
            "Effect": "Allow",
            "Action": [
               "bedrock:InvokeModel"
            ],
            "Resource": "*"
         }
      ]
   }
   ```

6. Wählen Sie Weiter aus.

7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen, z. B.**`BedrockAccessForAppStudio`**, und eine Beschreibung (optional) ein.
8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon Bedrock zu gewähren

Um Amazon Bedrock mit App Studio zu verwenden, müssen Administratoren eine IAM-Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Rolle steuert den Umfang der Berechtigungen, die App Studio-Apps verwenden dürfen, und wird bei der Erstellung des Connectors verwendet. Wir empfehlen, mindestens eine IAM-Rolle pro Dienst und Richtlinie zu erstellen.

Um eine IAM-Rolle zu erstellen, um App Studio Zugriff auf Amazon Bedrock zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie im vorherigen Schritt erstellt haben, und wählen Sie sie aus (**`BedrockAccessForAppStudio`**). Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

6. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.
7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

8. Wählen Sie Rolle erstellen und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie im nächsten Schritt den Amazon Bedrock-Connector in App Studio erstellen.

Schritt 3: Amazon Bedrock Connector erstellen

Nachdem Sie Ihre Amazon Bedrock-Ressourcen sowie die IAM-Richtlinie und -Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit Amazon Bedrock zu verbinden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können.

Um einen Connector für Amazon Bedrock zu erstellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie Andere AWS Dienste aus der Liste der Konnektortypen aus.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Amazon Bedrock Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Amazon Bedrock Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon Bedrock zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Service: Wählen Sie Bedrock Runtime.

     > **Note**
     >
     > Bedrock Runtime wird verwendet, um Inferenzanfragen für Modelle zu stellen, die in Amazon Bedrock gehostet werden, wohingegen Bedrock zur Verwaltung, Schulung und Bereitstellung von Modellen verwendet wird.

   - Region: Wählen Sie die AWS Region aus, in der sich Ihre Amazon Bedrock-Ressourcen befinden.
6. Wählen Sie Erstellen aus.

7. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Connect zu Amazon Simple Email Service her

Um App Studio mit Amazon SES zu verbinden, damit Builder es zum Senden von E-Mail-Benachrichtigungen aus ihren Apps verwenden können, müssen Sie die folgenden Schritte ausführen:

1. Schritt 1: Amazon SES SES-Ressourcen konfigurieren
2. Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon SES SES-Berechtigungen
3. Schritt 3: Amazon SES SES-Connector erstellen

Schritt 1: Amazon SES SES-Ressourcen konfigurieren

Falls nicht, müssen Sie zunächst Amazon SES so konfigurieren, dass es zum Senden von E-Mails verwendet wird. Weitere Informationen zur Einrichtung von Amazon SES finden Sie unter Erste Schritte mit Amazon Simple Email Service im Amazon Simple Email Service Developer Guide.

Schritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden Amazon SES SES-Berechtigungen

Um Amazon SES SES-Ressourcen mit App Studio zu verwenden, müssen Administratoren eine IAM-Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Rolle steuert, welche Amazon SES SES-Funktionen oder Ressourcen in App Studio-Apps verwendet werden können.

Wir empfehlen, mindestens eine IAM-Rolle pro Service und Richtlinie zu erstellen.

Schritt 2a: Erstellen Sie eine IAM-Richtlinie mit den entsprechenden Amazon SES SES-Berechtigungen

Die IAM-Richtlinie, die Sie mit App Studio erstellen und verwenden, sollte nur die minimal erforderlichen Berechtigungen für die entsprechenden Ressourcen enthalten, damit die Anwendung den besten Sicherheitspraktiken folgt.

Um eine IAM-Richtlinie mit entsprechenden Amazon SES SES-Berechtigungen zu erstellen

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Richtlinien zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im linken Navigationsbereich Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.
5. Geben Sie das folgende JSON-Richtliniendokument ein oder fügen Sie es ein.

> **Note**
>
> Die folgenden Richtlinien gelten für alle Amazon SES SES-Ressourcen, die den Platzhalter (*) verwenden. Aus Sicherheitsgründen sollten Sie den Platzhalter durch den Amazon-Ressourcennamen (ARN) der Ressourcen ersetzen, die Sie mit App Studio verwenden möchten.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ses:SendEmail",
            "Resource": "*"
        }
    ]
}
```

6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite Überprüfen und erstellen einen Richtliniennamen, z. B.**`SESForAppStudioPolicy`**, und eine Beschreibung (optional) ein.
8. Wählen Sie Richtlinie erstellen aus, um die Richtlinie zu erstellen.

Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon SES zu gewähren

Erstellen Sie nun eine IAM-Rolle, die die zuvor erstellte Richtlinie verwendet. App Studio verwendet diese Richtlinie, um Zugriff auf Amazon SES zu erhalten.

Um eine IAM-Rolle zu erstellen, um App Studio Zugriff auf Amazon SES zu gewähren

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den Administratorbenutzer zu verwenden, der in erstellt wurde. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Wählen Sie im Navigationsbereich auf der linken Seite Rollen
3. Wählen Sie Rolle erstellen aus.
4. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
5. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
```

```
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

6. Suchen Sie unter Berechtigungen hinzufügen nach der Richtlinie, die Sie im vorherigen Schritt erstellt haben, und wählen Sie sie aus (**`SESForAppStudioPolicy`**). Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt.

   Wählen Sie Weiter aus.

7. Geben Sie auf der Seite Name, Überprüfung und Erstellung einen Rollennamen und eine Beschreibung ein.
8. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

9. Wählen Sie Create role und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den Amazon SES-Connector in App Studio erstellen.

Schritt 3: Amazon SES SES-Connector erstellen

Nachdem Sie Amazon SES und eine IAM-Richtlinie und Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um Amazon SES in ihren Apps zu verwenden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Konnektoren erstellen zu können.

So erstellen Sie einen Connector für Amazon SES

1. Navigieren Sie zu App Studio.

2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie Andere AWS Dienste aus der Liste der Konnektortypen aus.
5. Konfigurieren Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen:

   - Name: Geben Sie einen Namen für Ihren Amazon SES SES-Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Amazon SES SES-Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) aus der IAM-Rolle ein, die in erstellt wurde. Schritt 2b: Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Amazon SES zu gewähren Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.
   - Service: Wählen Sie Simple Email Service.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre Amazon SES SES-Ressourcen befinden.
6. Wählen Sie Erstellen aus.
7. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Stellen Sie mithilfe des Connectors für andere AWS Dienste eine Verbindung zu Diensten her AWS

App Studio bietet zwar einige Konnektoren, die für bestimmte AWS Dienste spezifisch sind, Sie können jedoch auch über den Connector Andere AWS Dienste eine Verbindung zu anderen AWS Diensten herstellen.

> **Note**
>
> Es wird empfohlen, den für den AWS Dienst spezifischen Konnektor zu verwenden, sofern dieser verfügbar ist.

Um App Studio mit AWS Diensten zu verbinden, sodass Builder auf die Ressourcen des Dienstes in Anwendungen zugreifen und diese nutzen können, müssen Sie die folgenden Schritte ausführen:

1. Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Ressourcen zu AWS gewähren
2. Erstellen Sie einen Connector AWS für andere Dienste

Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Ressourcen zu AWS gewähren

Um AWS Dienste und Ressourcen mit App Studio nutzen zu können, müssen Administratoren eine IAM-Rolle erstellen, um App Studio Berechtigungen für den Zugriff auf die Ressourcen zu erteilen. Die IAM-Rolle steuert den Umfang der Ressourcen, auf die Builder zugreifen können, und steuert, welche Operationen für die Ressourcen aufgerufen werden können. Wir empfehlen, mindestens eine IAM-Rolle pro Service und Richtlinie zu erstellen.

Um eine IAM-Rolle zu erstellen, um App Studio Zugriff auf Ressourcen zu gewähren AWS

1. Melden Sie sich bei der IAM-Konsole mit einem Benutzer an, der berechtigt ist, IAM-Rollen zu erstellen. Wir empfehlen, den in erstellten Administratorbenutzer zu verwenden. Erstellen Sie einen Administratorbenutzer für die Verwaltung von AWS Ressourcen
2. Klicken Sie im Navigationsbereich der Konsole auf Roles (Rollen) und wählen Sie dann Create role (Rolle erstellen).
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option Benutzerdefinierte Vertrauensrichtlinie aus.
4. Ersetzen Sie die Standardrichtlinie durch die folgende Richtlinie, damit App Studio-Anwendungen diese Rolle in Ihrem Konto übernehmen können.

   Sie müssen die folgenden Platzhalter in der Richtlinie ersetzen. Die zu verwendenden Werte finden Sie in App Studio auf der Seite mit den Kontoeinstellungen.

   - Ersetzen Sie sie durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
   - *11111111-2222-3333-4444-555555555555*Ersetzen Sie es durch Ihre App Studio-Instanz-ID, die in den Kontoeinstellungen Ihrer App Studio-Instanz als Instanz-ID aufgeführt ist.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "sts:AssumeRole",
```

```
            "Condition": {
                "StringEquals": {
                        "aws:PrincipalTag/IsAppStudioAccessRole": "true",
                        "sts:ExternalId": "11111111-2222-3333-4444-555555555555"
                }
            }
        }
    ]
}
```

Wählen Sie Weiter aus.

5. Suchen Sie unter Berechtigungen hinzufügen nach den Richtlinien, die die entsprechenden Berechtigungen für die Rolle gewähren, und wählen Sie sie aus. Wenn Sie das Pluszeichen (+) neben einer Richtlinie auswählen, wird die Richtlinie so erweitert, dass die von ihr erteilten Berechtigungen angezeigt werden. Wenn Sie das Kontrollkästchen aktivieren, wird die Richtlinie ausgewählt. Weitere Informationen zu IAM finden Sie im IAM-Benutzerhandbuch.

   Wählen Sie Weiter aus.

6. Geben Sie im Feld Rollendetails einen Namen und eine Beschreibung ein.
7. Wählen Sie in Schritt 3: Tags hinzufügen die Option Neues Tag hinzufügen aus, um das folgende Tag hinzuzufügen und App Studio-Zugriff zu gewähren:

   - Schlüssel: `IsAppStudioDataAccessRole`
   - Wert: `true`

8. Wählen Sie Rolle erstellen und notieren Sie sich den generierten Amazon-Ressourcennamen (ARN). Sie benötigen ihn, wenn Sie den Connector für andere AWS Dienste in App Studio erstellen.

Erstellen Sie einen Connector AWS für andere Dienste

Nachdem Sie Ihre IAM-Rolle konfiguriert haben, verwenden Sie diese Informationen, um den Connector in App Studio zu erstellen, den Builder verwenden können, um ihre Apps mit dem Dienst und den Ressourcen zu verbinden.

> **Note**
>
> Sie benötigen die Administratorrolle in App Studio, um Connectors erstellen zu können.

Um über den Connector „Andere AWS Dienste“ eine Verbindung zu AWS Diensten herzustellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie Andere AWS Dienste im Bereich AWS Konnektoren der Liste der unterstützten Dienste aus.
5. Konfigurieren Sie Ihren AWS Service Connector, indem Sie die folgenden Felder ausfüllen:
   - Name: Geben Sie einen Namen für Ihren Connector ein.
   - Beschreibung: Geben Sie eine Beschreibung für Ihren Connector ein.
   - IAM-Rolle: Geben Sie den Amazon-Ressourcennamen (ARN) der IAM-Rolle ein, die in erstellt wurde. Erstellen Sie eine IAM-Rolle, um App Studio Zugriff auf Ressourcen zu AWS gewähren
   - Service: Wählen Sie den AWS Service aus, den Sie mit App Studio verbinden möchten.
   - Region: Wählen Sie die AWS Region aus, in der sich Ihre AWS Ressourcen befinden.
6. Wählen Sie Erstellen aus. Der neu erstellte Connector wird in der Connectorliste angezeigt.

Verwenden Sie verschlüsselte Datenquellen mit CMKs

Dieses Thema enthält Informationen zum Einrichten und Verbinden von App Studio mit Datenquellen, die mit einem vom AWS KMS Kunden verwalteten Schlüssel (CMK) verschlüsselt sind.

Inhalt

- Verwenden von verschlüsselten verwalteten Datenspeichertabellen
- Verschlüsselte DynamoDB-Tabellen verwenden

Verwenden von verschlüsselten verwalteten Datenspeichertabellen

Verwenden Sie das folgende Verfahren, um die DynamoDB-Tabellen zu verschlüsseln, die von verwalteten Speicherentitäten in Ihren App Studio-Apps verwendet werden. Weitere Informationen zu verwalteten Datenentitäten finden Sie unter. Verwaltete Datenentitäten in AWS App Studio

So verwenden Sie verschlüsselte verwaltete Datenspeichertabellen

1. Erstellen Sie bei Bedarf die verwalteten Datenentitäten in einer Anwendung in App Studio. Weitere Informationen finden Sie unter Eine Entität mit einer von App Studio verwalteten Datenquelle erstellen.
2. Fügen Sie der `AppStudioManagedStorageDDBAccess` IAM-Rolle eine Richtlinienerklärung mit Berechtigungen zum Verschlüsseln und Entschlüsseln von Tabellendaten mit Ihrem CMK hinzu, indem Sie die folgenden Schritte ausführen:

   a. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.

   > **Important**
   >
   > Sie müssen dasselbe Konto verwenden, mit dem Sie Ihre App Studio-Instanz erstellt haben.

   b. Wählen Sie im Navigationsbereich der IAM Console Roles (Rollen) aus.

   c. Wählen Sie `AppStudioManagedStorageDDBAccess`.

   d. Wählen Sie unter Berechtigungsrichtlinien die Option Berechtigungen hinzufügen und dann Inline-Richtlinie erstellen aus.

   e. Wählen Sie JSON und ersetzen Sie den Inhalt durch die folgende Richtlinie, die Folgendes ersetzt:

      - Ersetzen Sie es durch die AWS Kontonummer des Kontos, *111122223333* mit dem die App Studio-Instanz eingerichtet wurde. Diese ist in den AWS Kontoeinstellungen Ihrer App Studio-Instanz als Konto-ID aufgeführt.
      - *CMK_id*Durch CMK-ID ersetzen. Informationen dazu, wie Sie ihn finden, finden Sie unter Finden der Schlüssel-ID und des Schlüssel-ARN.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "connector_cmk_support",
      "Effect": "Allow",
      "Action": [ "kms:Decrypt", "kms:Encrypt"],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/CMK_id"
    }
```

```
  ]
}
```

3. Verschlüsseln Sie die DynamoDB-Tabellen, die von Ihren verwalteten App Studio-Datenentitäten verwendet werden, indem Sie die folgenden Schritte ausführen:

   a. Öffnen Sie die Amazon DynamoDB DynamoDB-Konsole unter. https://console.aws.amazon.com/dynamodbv2/

   b. Wählen Sie die Tabelle aus, die Sie verschlüsseln möchten. Sie finden den Tabellennamen auf der Registerkarte Verbindung der entsprechenden Entität in App Studio.

   c. Wählen Sie Zusätzliche Einstellungen

   d. Wählen Sie unter Verschlüsselung die Option Verschlüsselung verwalten aus.

   e. Wählen Sie In Ihrem Konto gespeichert und von Ihnen verwaltet aus und wählen Sie Ihr CMK aus.

4. Testen Sie Ihre Änderungen, indem Sie Ihre App erneut veröffentlichen und sicherstellen, dass das Lesen und Schreiben von Daten sowohl in der Test- als auch in der Produktionsumgebung funktioniert und dass die Verwendung dieser Tabelle in einer anderen Entität erwartungsgemäß funktioniert.

> **Note**
>
> Alle neu hinzugefügten verwalteten Datenentitäten verwenden standardmäßig den von DynamoDB verwalteten Schlüssel und müssen mithilfe der vorherigen Schritte auf die Verwendung des CMK aktualisiert werden.

Verschlüsselte DynamoDB-Tabellen verwenden

Gehen Sie wie folgt vor, um verschlüsselte DynamoDB-Tabellen für die Verwendung in Ihren App Studio-Apps zu konfigurieren.

So verwenden Sie verschlüsselte DynamoDB-Tabellen

1. Folgen Sie den Anweisungen unter Schritt 1: DynamoDB-Ressourcen erstellen und konfigurieren mit den folgenden Änderungen:

- Konfigurieren Sie Ihre Tabellen so, dass sie verschlüsselt werden. Weitere Informationen finden Sie unter Angeben des Verschlüsselungsschlüssels für eine neue Tabelle im Amazon DynamoDB Developer Guide.

2. Folgen Sie den Anweisungen unter und aktualisieren Sie dann die Berechtigungsrichtlinie für die neue RolleSchritt 2: Erstellen Sie eine IAM-Richtlinie und -Rolle mit den entsprechenden DynamoDB-Berechtigungen, indem Sie eine neue Richtlinienerklärung hinzufügen, die es ihr erlaubt, Tabellendaten mit Ihrem CMK zu ver- und entschlüsseln, indem Sie die folgenden Schritte ausführen:

   a. Navigieren Sie bei Bedarf in der IAM-Konsole zu Ihrer Rolle.

   b. Wählen Sie unter Berechtigungsrichtlinien die Option Berechtigungen hinzufügen und dann Inline-Richtlinie erstellen aus.

   c. Wählen Sie JSON und ersetzen Sie den Inhalt durch die folgende Richtlinie, die Folgendes ersetzt:

      - *team_account_id*Ersetzen Sie es durch Ihre App Studio-Team-ID, die Sie in Ihren Kontoeinstellungen finden.
      - *CMK_id*Durch CMK-ID ersetzen. Informationen dazu, wie Sie ihn finden, finden Sie unter Finden der Schlüssel-ID und des Schlüssel-ARN.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "connector_cmk_support",
      "Effect": "Allow",
      "Action": [ "kms:Decrypt", "kms:Encrypt"],
      "Resource": "arn:aws:kms:us-west-2:team_account_id:key/CMK_id"
    }
  ]
}
```

3. Erstellen Sie den Connector, indem Sie den Anweisungen unter folgen DynamoDB-Konnektor erstellen und die Rolle verwenden, die Sie zuvor erstellt haben.

4. Testen Sie die Konfiguration, indem Sie eine App, die den DynamoDB-Connector und die DynamoDB-Tabelle verwendet, in Testing oder Production veröffentlichen. Stellen Sie sicher,

dass das Lesen und Schreiben von Daten funktioniert, und dass die Verwendung dieser Tabelle zum Erstellen einer anderen Entität ebenfalls funktioniert.

> **Note**
>
> Wenn neue DynamoDB-Tabellen erstellt werden, müssen Sie sie so konfigurieren, dass sie mit einem CMK verschlüsselt werden, indem Sie die vorherigen Schritte ausführen.

Connect zu Diensten von Drittanbietern her

Themen

- OpenAPI-Konnektor im Vergleich zu API-Konnektor
- Connect zu Diensten von Drittanbietern her und APIs (generisch)
- Stellen Sie mit OpenAPI eine Connect zu Diensten her
- Mit Salesforce Connect

OpenAPI-Konnektor im Vergleich zu API-Konnektor

Um API-Anfragen von App Studio-Anwendungen an Dienste von Drittanbietern zu senden, müssen Sie einen Connector erstellen und konfigurieren, den die Anwendung zur Authentifizierung beim Dienst und zur Konfiguration der API-Aufrufe verwendet. App Studio stellt hierfür `API Connector` sowohl die `OpenAPI Connector` Konnektortypen als auch die Konnektortypen bereit, die wie folgt beschrieben werden:

- API-Konnektor: Wird verwendet, um die Authentifizierung zu konfigurieren und Informationen für jede Art von REST-API anzufordern.
- OpenAPI-Konnektor: Wird verwendet, um die Authentifizierung zu konfigurieren und Informationen für Personen anzufordern APIs , die die OpenAPI-Spezifikation (OAS) übernommen haben. APIs die sich an die OAS halten, bieten mehrere Vorteile, darunter Standardisierung, Sicherheit, Verwaltung und Dokumentation.

App Studio empfiehlt die Verwendung von `OpenAPI Connector` für alle APIs , die dem OAS entsprechen, und eine OpenAPI-Spezifikationsdatei bereitzustellen. Weitere Informationen zu OpenAPI finden Sie unter Was ist OpenAPI? in der Swagger-Dokumentation.

Connect zu Diensten von Drittanbietern her und APIs (generisch)

Gehen Sie wie folgt vor, um einen generischen API-Connector in App Studio zu erstellen. Der API Connector wird verwendet, um App Studio-Apps Zugriff auf Dienste, Ressourcen oder Operationen von Drittanbietern zu gewähren.

Um mit dem API Connector eine Verbindung zu Diensten von Drittanbietern herzustellen

1. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
2. Wählen Sie + Connector erstellen.
3. Wählen Sie API-Konnektor. Konfigurieren Sie nun Ihren Connector, indem Sie die folgenden Felder ausfüllen.
4. Connectorname: Geben Sie einen Namen für Ihren Connector ein.
5. Beschreibung des Connectors: Geben Sie eine Beschreibung für Ihren Connector ein.
6. Basis-URL: Die Website oder der Host der Drittanbieter-Verbindung. Beispiel, `www.slack.com`.
7. Authentifizierungsmethode: Wählen Sie die Methode für die Authentifizierung beim Zieldienst aus.
    - Keine: Greifen Sie ohne Authentifizierung auf den Zieldienst zu.
    - Basic: Greifen Sie mit einem Benutzernamen und einem Passwort auf den Zieldienst zu, mit dem Sie verbunden sind.
    - Trägertoken: Greifen Sie mit dem Token-Wert eines Authentifizierungstokens auf den Zieldienst zu, das aus dem Benutzerkonto oder den API-Einstellungen des Dienstes abgerufen wurde.
    - OAuth 2.0: Greifen Sie über das OAuth 2.0-Protokoll auf den Zieldienst zu, das App Studio Zugriff auf den Dienst und die Ressourcen gewährt, ohne Anmeldeinformationen oder Identität weiterzugeben. Um die OAuth 2.0-Authentifizierungsmethode zu verwenden, müssen Sie zunächst eine Anwendung aus dem Dienst erstellen, mit dem eine Verbindung hergestellt wird und der App Studio repräsentiert, um die erforderlichen Informationen abzurufen. Füllen Sie mit diesen Informationen die folgenden Felder aus:

        a. Fluss von Kundenanmeldedaten: Ideal für system-to-system Interaktionen, bei denen die Anwendung in ihrem eigenen Namen ohne Benutzerinteraktion agiert. Zum Beispiel eine CRM-Anwendung, die Salesforce-Datensätze automatisch auf der Grundlage

neuer von Benutzern hinzugefügter Datensätze aktualisiert, oder eine Anwendung, die Transaktionsdaten abruft und in Berichten anzeigt.

1. Geben Sie im Feld Client-ID die ID ein, die Sie von der im Zieldienst erstellten OAuth App erhalten haben.
2. Geben Sie im Feld Geheimer Client den geheimen Schlüssel ein, den Sie von der im Zieldienst erstellten OAuth App erhalten haben.
3. Geben Sie im Feld Zugriffstoken-URL die Token-URL ein, die von der im Zieldienst erstellten OAuth App abgerufen wurde.
4. Geben Sie optional im Feld Bereiche die Bereiche für die Anwendung ein. Bereiche sind Berechtigungen oder Zugriffsebenen, die für die Anwendung erforderlich sind. Informationen zu deren Geltungsbereichen finden Sie in der API-Dokumentation des Zieldienstes, und konfigurieren Sie nur die Bereiche, die Ihre App Studio-App benötigt.

Wählen Sie Verbindung verifizieren, um die Authentifizierung und Verbindung zu testen.

b. Ablauf des Autorisierungscodes: Ideal für Anwendungen, bei denen im Namen eines Benutzers gehandelt werden muss. Zum Beispiel eine Kundensupport-App, in der sich Benutzer anmelden und Supporttickets einsehen und aktualisieren, oder eine Verkaufs-App, bei der sich jedes Teammitglied anmeldet, um seine Verkaufsdaten einzusehen und zu verwalten.

1. Geben Sie im Feld Client-ID die ID ein, die Sie von der im Zieldienst erstellten OAuth App erhalten haben.
2. Geben Sie im Feld Geheimer Client den geheimen Schlüssel ein, den Sie von der im Zieldienst erstellten OAuth App erhalten haben.
3. Geben Sie im Feld Autorisierungs-URL die Autorisierungs-URL vom Zieldienst ein.
4. Geben Sie im Feld Zugriffstoken-URL die Token-URL ein, die von der im Zieldienst erstellten OAuth App abgerufen wurde.
5. Geben Sie optional im Feld Bereiche die Bereiche für die Anwendung ein. Bereiche sind Berechtigungen oder Zugriffsebenen, die für die Anwendung erforderlich sind. Informationen zu deren Geltungsbereichen finden Sie in der API-Dokumentation des Zieldienstes, und konfigurieren Sie nur die Bereiche, die Ihre App Studio-App benötigt.

8. Header: Fügen Sie HTTP-Header hinzu, die zur Bereitstellung von Metadaten zur Anfrage oder Antwort verwendet werden. Sie können sowohl Schlüssel als auch Werte hinzufügen oder nur einen Schlüssel angeben, für den der Builder in der Anwendung einen Wert angeben kann.
9. Abfrageparameter: Fügen Sie Abfrageparameter hinzu, die zur Übergabe von Optionen, Filtern oder Daten als Teil der Anforderungs-URL verwendet werden. Wie bei Headern können Sie sowohl einen Schlüssel als auch einen Wert angeben oder nur einen Schlüssel angeben, für den der Builder in der Anwendung einen Wert angeben kann.
10. Wählen Sie Erstellen aus. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Jetzt, da der Connector erstellt wurde, können Builder ihn in ihren Apps verwenden.

Stellen Sie mit OpenAPI eine Connect zu Diensten her

Gehen Sie wie folgt vor, um App Studio mithilfe von OpenAPI mit Diensten zu verbinden, sodass Builder Anwendungen erstellen können, die Anfragen senden und Antworten von den Diensten empfangen:

1. Holen Sie sich die OpenAPI-Spezifikationsdatei und sammeln Sie Serviceinformationen
2. OpenAPI-Konnektor erstellen

Holen Sie sich die OpenAPI-Spezifikationsdatei und sammeln Sie Serviceinformationen

Gehen Sie wie folgt vor, um einen Dienst mit App Studio mit OpenAPI zu verbinden:

1. Gehen Sie zu dem Dienst, den Sie mit App Studio verbinden möchten, und suchen Sie nach einer OpenAPI-Spezifikations-JSON-Datei.

   > **Note**
   >
   > App Studio unterstützt OpenAPI-Spezifikationsdateien, die der Version OpenAPI-Spezifikation Version 3.0.0 oder höher entsprechen.

2. Sammeln Sie die erforderlichen Daten, um den OpenAPI-Konnektor zu konfigurieren, einschließlich der folgenden:

   - Die Basis-URL für die Verbindung mit dem Dienst.
   - Anmeldeinformationen zur Authentifizierung, z. B. ein Token oder Benutzername/Passwort.
   - Falls zutreffend, alle Header.

- Falls zutreffend, alle Abfrageparameter.

OpenAPI-Konnektor erstellen

Um einen Connector für OpenAPI zu erstellen

1. Navigieren Sie zu App Studio.
2. Wählen Sie im linken Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
3. Wählen Sie + Connector erstellen.
4. Wählen Sie OpenAPI Connector aus der Liste der Connectortypen aus. Konfigurieren Sie nun Ihren Connector, indem Sie die folgenden Felder ausfüllen.
5. Name: Geben Sie einen Namen für Ihren OpenAPI-Connector ein.
6. Beschreibung: Geben Sie eine Beschreibung für Ihren OpenAPI-Konnektor ein.
7. Basis-URL: Geben Sie die Basis-URL für die Verbindung mit dem Dienst ein.
8. Authentifizierungsmethode: Wählen Sie die Methode für die Authentifizierung beim Zieldienst aus.
   - Keine: Greifen Sie ohne Authentifizierung auf den Zieldienst zu.
   - Basic: Greifen Sie mit einem Benutzernamen und einem Passwort auf den Zieldienst zu, mit dem Sie verbunden sind.
   - Trägertoken: Greifen Sie mit dem Token-Wert eines Authentifizierungstokens auf den Zieldienst zu, das aus dem Benutzerkonto oder den API-Einstellungen des Dienstes abgerufen wurde.
   - OAuth 2.0: Greifen Sie über das OAuth 2.0-Protokoll auf den Zieldienst zu, das App Studio Zugriff auf den Dienst und die Ressourcen gewährt, ohne Anmeldeinformationen oder Identität weiterzugeben. Um die OAuth 2.0-Authentifizierungsmethode zu verwenden, müssen Sie zunächst eine Anwendung aus dem Dienst erstellen, mit dem eine Verbindung hergestellt wird und der App Studio repräsentiert, um die erforderlichen Informationen abzurufen. Füllen Sie mit diesen Informationen die folgenden Felder aus:

     a. Ablauf der Kundenanmeldedaten:

        1. Geben Sie im Feld Client-ID die ID des Zieldienstes ein.
        2. Geben Sie im Feld Geheimer Client den geheimen Schlüssel des Zieldienstes ein.

3. Geben Sie im Feld Zugriffstoken-URL die Token-URL des Zieldienstes ein.
4. Geben Sie optional im Feld Bereiche die Bereiche für die Anwendung ein. Bereiche sind Berechtigungen oder Zugriffsebenen, die für die Anwendung erforderlich sind. Informationen zu deren Geltungsbereichen finden Sie in der API-Dokumentation des Zieldienstes, und konfigurieren Sie nur die Bereiche, die Ihre App Studio-App benötigt.

Fügen Sie bei jedem Aufruf alle Variablen hinzu, die mit dem Dienst gesendet werden sollen, und wählen Sie Verbindung verifizieren, um die Authentifizierung und Verbindung zu testen.

b. Ablauf des Autorisierungscodes:

1. Geben Sie im Feld Client-ID die ID des Zieldienstes ein.
2. Geben Sie im Feld Geheimer Client den geheimen Schlüssel des Zieldienstes ein.
3. Geben Sie im Feld Autorisierungs-URL die Autorisierungs-URL des Zieldienstes ein.
4. Geben Sie im Feld Zugriffstoken-URL die Token-URL des Zieldienstes ein.
5. Geben Sie optional im Feld Bereiche die Bereiche für die Anwendung ein. Bereiche sind Berechtigungen oder Zugriffsebenen, die für die Anwendung erforderlich sind. Informationen zu deren Geltungsbereichen finden Sie in der API-Dokumentation des Zieldienstes, und konfigurieren Sie nur die Bereiche, die Ihre App Studio-App benötigt.

9. Variablen: Fügen Sie Variablen hinzu, die bei jedem Aufruf an den Dienst gesendet werden sollen. Variablen, die während der Konfiguration hinzugefügt wurden, werden sicher gespeichert und nur während der Laufzeit von Anwendungen abgerufen, die die Verbindung verwenden.
10. Header: Fügen Sie HTTP-Header hinzu, die zur Bereitstellung von Metadaten zur Anfrage oder Antwort verwendet werden. Sie können sowohl Schlüssel als auch Werte hinzufügen oder nur einen Schlüssel angeben, für den der Builder in der Anwendung einen Wert angeben kann.
11. Abfrageparameter: Fügen Sie Abfrageparameter hinzu, die zur Übergabe von Optionen, Filtern oder Daten als Teil der Anforderungs-URL verwendet werden. Wie bei Headern können Sie sowohl einen Schlüssel als auch einen Wert angeben oder nur einen Schlüssel angeben, für den der Builder in der Anwendung einen Wert angeben kann.
12. OpenAPI-Spezifikationsdatei: Laden Sie eine OpenAPI-Spezifikations-JSON-Datei hoch, indem Sie sie ziehen und ablegen, oder wählen Sie Datei auswählen, um in Ihrem lokalen Dateisystem zu navigieren und die hochzuladende Datei auszuwählen.

Nach dem Hinzufügen wird die Datei verarbeitet und eine Liste der verfügbaren Optionen wird angezeigt. Wählen Sie die erforderlichen Operationen für Ihren Connector aus.

13. Wählen Sie Erstellen aus. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Jetzt, da der Connector erstellt wurde, können Builder ihn in ihren Apps verwenden.

Mit Salesforce Connect

Um App Studio mit Salesforce zu verbinden, damit Builder auf Salesforce-Ressourcen zugreifen und diese in Anwendungen verwenden können, müssen Sie eine verbundene Anwendung in Salesforce erstellen und konfigurieren und in App Studio einen Salesforce-Connector erstellen.

Um Salesforce mit App Studio zu verbinden

1. Wählen Sie in App Studio im Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der vorhandenen Konnektoren mit einigen Details zu den einzelnen Konnektoren angezeigt wird.
2. Wählen Sie + Connector erstellen.
3. Wählen Sie Salesforce aus der Liste der Connector-Typen aus, um die Seite zur Connector-Erstellung zu öffnen.
4. Notieren Sie sich die Umleitungs-URL, die Sie in den folgenden Schritten zur Konfiguration von Salesforce verwenden werden.
5. Der nächste Schritt besteht darin, eine verbundene Anwendung in Salesforce zu erstellen. Navigieren Sie in einer anderen Registerkarte oder einem anderen Fenster zu Ihrer Salesforce-Instanz.
6. Suchen Sie im Feld „Schnellsuche“ **`App Manager`** und wählen Sie dann App Manager aus.
7. Wählen Sie „Neue verbundene App“.
8. Geben Sie unter Name der verbundenen App und API-Name einen Namen für Ihre App ein. Er muss nicht mit dem Namen Ihrer App Studio-App übereinstimmen.
9. Geben Sie bei Bedarf Kontaktinformationen an.
10. Aktivieren Sie im Abschnitt API (OAuth Einstellungen aktivieren) die Option OAuth Einstellungen aktivieren.
11. Geben Sie unter Rückruf-URL die Weiterleitungs-URL ein, die Sie zuvor in App Studio notiert haben.

12. Fügen Sie unter Ausgewählte OAuth Bereiche die erforderlichen Berechtigungsbereiche aus der Liste hinzu. App Studio kann mit Salesforce REST interagieren APIs , um CRUD-Operationen für fünf Objekte durchzuführen: Accounts, Kundenvorgänge, Kontakte, Leads und Opportunities. Es wird empfohlen, Vollzugriff (Vollzugriff) hinzuzufügen, um sicherzustellen, dass Ihre App Studio-Anwendung über alle relevanten Berechtigungen oder Bereiche verfügt.
13. Deaktivieren Sie die Option „Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows erforderlich“. PKCE wird von App Studio nicht unterstützt.
14. Aktivieren Sie „Geheimer Schlüssel für Webserverfluss erforderlich“ und „Geheimcode erforderlich“ für Aktualisierungstokenfluss, um die besten Sicherheitsvorkehrungen zu befolgen.
15. App Studio unterstützt die beiden folgenden Authentifizierungsabläufe:

    - Client Credentials Flow: Ideal für server-to-server Interaktionen, bei denen die Anwendung in ihrem eigenen Namen ohne Benutzerinteraktion agiert. Zum Beispiel die Auflistung aller Lead-Informationen für ein Team von temporären Mitarbeitern, die keinen Salesforce-Zugriff haben.
    - Authorization Code Flow: Geeignet für Anwendungen, die im Namen eines Benutzers handeln, z. B. für den Zugriff auf personenbezogene Daten oder für Aktionen. Zum Beispiel eine Auflistung der Leads der einzelnen Vertriebsleiter, die sie bezogen haben oder denen sie gehören, um andere Aufgaben über diese App auszuführen.

    - Für den Client Credentials Flow:

      a. Aktivieren Sie „Fluss für Client-Anmeldeinformationen aktivieren“. Überprüfen und bestätigen Sie die Nachricht.

      b. Speichern Sie die App.

      c. Sie müssen einen Ausführungsbenutzer auswählen, obwohl im Flow keine Benutzerinteraktion stattfindet. Durch die Auswahl eines Ausführungsbenutzers gibt Salesforce Zugriffstoken im Namen des Benutzers zurück.

         1. Wählen Sie im App Manager aus der Liste der Apps den Pfeil der App Studio-Anwendung und dann Verwalten aus.
         2. Wählen Sie „Richtlinien bearbeiten“
         3. Fügen Sie im Client Credentials Flow den entsprechenden Benutzer hinzu.

    - Aktivieren Sie für den Autorisierungscodefluss die Option Autorisierungscode- und Anmeldedatenfluss aktivieren

16. Salesforce stellt eine Client-ID und einen geheimen Clientschlüssel bereit, die zur Konfiguration des Connectors in App Studio in den folgenden Schritten verwendet werden müssen.

    a. Wählen Sie im App Manager den Pfeil der App Studio-Anwendung und dann Ansicht aus.

    b. Wählen Sie im Abschnitt API (OAuth Einstellungen aktivieren) die Option Kundendaten verwalten aus. Dadurch wird möglicherweise eine E-Mail mit einem Bestätigungsschlüssel gesendet, den Sie zur Bestätigung eingeben müssen.

    c. Notieren Sie sich den Consumer Key (Client ID) und den Consumer Secret (Client Secret).

17. Zurück in App Studio, konfigurieren und erstellen Sie Ihren Connector, indem Sie die folgenden Felder ausfüllen.

18. Geben Sie unter Name einen Namen für Ihren Salesforce-Connector ein.

19. Geben Sie unter Beschreibung eine Beschreibung für Ihren Salesforce-Connector ein.

20. Geben Sie im Feld Basis-URL die Basis-URL für Ihre Salesforce-Instanz ein. Sie sollte wie folgt aussehen: `https://hostname.salesforce.com/services/data/v60.0` und durch den Namen Ihrer Salesforce-Instanz *hostname* ersetzen.

21. Stellen Sie sicher, dass unter Authentifizierungsmethode OAuth 2.0 ausgewählt ist.

22. Wählen Sie in OAuth 2.0 Flow die OAuth Authentifizierungsmethode aus und füllen Sie die entsprechenden Felder aus:

    - Wählen Sie Client Credentials Flow für die Verwendung in Anwendungen, die in ihrem eigenen Namen agieren, für system-to-system Integrationen aus.

      a. Geben Sie im Feld Client-ID den Verbraucherschlüssel ein, den Sie zuvor von Salesforce erhalten haben.

      b. Geben Sie im Feld Kundengeheimnis den Consumer Secret ein, den Sie zuvor von Salesforce erhalten haben.

      c. Geben Sie im Feld Zugriffstoken-URL den OAuth 2.0-Token-Endpunkt ein. Er sollte wie folgt aussehen: `https://hostname/services/oauth2/token` und durch den Namen Ihrer Salesforce-Instanz *hostname* ersetzen. Weitere Informationen finden Sie in der Dokumentation zu Salesforce OAuth Endpoints.

      d. Wählen Sie Verbindung verifizieren aus, um die Authentifizierung und Verbindung zu testen.

    - Wählen Sie den Autorisierungscodefluss für die Verwendung in Anwendungen aus, die im Namen des Benutzers handeln.

a. Geben Sie im Feld Client-ID den zuvor von Salesforce erhaltenen Verbraucherschlüssel ein.

b. Geben Sie im Feld Kundengeheimnis den Consumer Secret ein, den Sie zuvor von Salesforce erhalten haben.

c. Geben Sie im Feld Autorisierungs-URL den Autorisierungsendpunkt ein. Er sollte wie folgt aussehen: `https://hostname/services/oauth2/authorize` und durch den Namen Ihrer Salesforce-Instanz *hostname* ersetzen. Weitere Informationen finden Sie in der Dokumentation zu Salesforce OAuth Endpoints.

d. Geben Sie im Feld Zugriffstoken-URL den OAuth 2.0-Token-Endpunkt ein. Er sollte wie folgt aussehen: `https://hostname/services/oauth2/token` und durch den Namen Ihrer Salesforce-Instanz *hostname* ersetzen. Weitere Informationen finden Sie in der Dokumentation zu Salesforce OAuth Endpoints.

23. Wählen Sie unter Operations die Salesforce-Operationen aus, die Ihr Connector unterstützen wird. Die Operationen in dieser Liste sind vordefiniert und stellen allgemeine Aufgaben in Salesforce dar, z. B. das Erstellen, Abrufen, Aktualisieren oder Löschen von Datensätzen aus allgemeinen Objekten.
24. Wählen Sie Erstellen aus. Der neu erstellte Connector wird in der Connectors-Liste angezeigt.

Konnektoren anzeigen, bearbeiten und löschen

Um bestehende Konnektoren anzuzeigen, zu bearbeiten oder zu löschen

1. Wählen Sie im Navigationsbereich im Abschnitt Verwalten die Option Connectors aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste vorhandener Konnektoren mit den folgenden Details für jeden Connector angezeigt wird:
   - Name: Der Name des Connectors, der bei der Erstellung angegeben wurde.
   - Beschreibung: Die Beschreibung des Connectors, der bei der Erstellung angegeben wurde.
   - Verbunden mit: Der Dienst, den der Connector mit App Studio verbindet. Der Wert API steht für eine Verbindung zu einem Dienst eines Drittanbieters.
   - Erstellt von: Der Benutzer, der den Connector erstellt hat.
   - Erstellungsdatum: Das Datum, an dem der Connector erstellt wurde.
2. Gehen Sie wie folgt vor, um weitere Details zu einem Connector anzuzeigen oder einen Connector zu bearbeiten oder zu löschen:

- Um weitere Informationen zu einem bestimmten Connector anzuzeigen, wählen Sie View für diesen Connector aus.
- Um einen Connector zu bearbeiten, wählen Sie das Drop-down-Menü neben Ansicht und wählen Sie Bearbeiten.
- Um einen Connector zu löschen, wählen Sie das Drop-down-Menü neben Ansicht und wählen Sie Löschen.

Löschen einer App Studio-Instanz

Gehen Sie wie in diesem Thema beschrieben vor, um Ihre App Studio-Instanz zu löschen. Wenn Sie Ressourcen in anderen Diensten für die Verwendung mit App Studio erstellt haben, überprüfen und löschen Sie diese nach Bedarf, um zu vermeiden, dass ihnen Gebühren berechnet werden.

Möglicherweise möchten Sie eine App Studio-Instanz aus den folgenden Gründen löschen:

- Sie möchten App Studio nicht mehr verwenden.
- Sie möchten eine App Studio-Instanz in einer anderen AWS Region erstellen. Da App Studio jeweils nur eine Instanz in einer Region unterstützt, müssen Sie alle vorhandenen Instanzen löschen, um eine weitere zu erstellen.

> **Warning**
>
> Durch das Löschen einer App Studio-Instanz werden auch alle App Studio-Ressourcen wie Anwendungen und Konnektoren gelöscht. Das Löschen einer Instanz kann nicht rückgängig gemacht werden.

Um Ihre App Studio-Instanz zu löschen

1. Öffnen Sie die App Studio-Konsole unter https://console.aws.amazon.com/appstudio/.
2. Wählen Sie die Region aus, in der Ihre App Studio-Instanz existiert.
3. Wählen Sie im Navigationsbereich Instanz aus.
4. Wählen Sie Aktionen, um das Drop-down-Menü mit weiteren Instanzaktionen zu öffnen.
5. Wählen Sie App Studio-Instanz löschen.
6. Geben Sie **`confirm`** ein und wählen Sie Delete (Löschen).

7. Es kann eine Weile dauern, bis das Löschen Ihrer Instanz abgeschlossen ist. Sobald die Datei gelöscht wurde, erhalten Sie eine Bestätigungs-E-Mail. Sobald Sie die E-Mail erhalten haben, können Sie bei Bedarf eine weitere Instanz erstellen.

Builder-Dokumentation

Die folgenden Themen enthalten Informationen, die Benutzern in App Studio helfen sollen, die Anwendungen erstellen, bearbeiten und veröffentlichen.

Themen

- Tutorials
- Erstellen Sie Ihre App Studio-App mit generativer KI
- Anwendungen erstellen, bearbeiten und löschen
- Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen
- Seiten und Komponenten: Erstellen Sie die Benutzeroberfläche Ihrer App
- Automatisierungen und Aktionen: Definieren Sie die Geschäftslogik Ihrer App
- Entitäten und Datenaktionen: Konfigurieren Sie das Datenmodell Ihrer App
- Seiten- und Automatisierungsparameter
- Verwenden JavaScript , um Ausdrücke in App Studio zu schreiben
- Datenabhängigkeiten und zeitliche Überlegungen
- Eine App mit mehreren Benutzern erstellen
- Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren

Tutorials

Themen

- Erstellen Sie mit Amazon Bedrock eine KI-App zur Textzusammenfassung
- Interaktion mit Amazon Simple Storage Service mit Komponenten und Automatisierungen
- Lambda-Funktionen in einer App Studio-App aufrufen

Erstellen Sie mit Amazon Bedrock eine KI-App zur Textzusammenfassung

In diesem Tutorial erstellen Sie eine Anwendung in App Studio, die Amazon Bedrock verwendet, um präzise Zusammenfassungen der Texteingaben von Endbenutzern bereitzustellen. Die Anwendung enthält eine einfache Benutzeroberfläche, über die Benutzer beliebigen Text eingeben können, den sie zusammenfassen möchten. Dies können Besprechungsnotizen, Artikelinhalte,

Forschungsergebnisse oder andere Textinformationen sein. Nachdem Benutzer den Text eingegeben haben, können sie eine Taste drücken, um den Text an Amazon Bedrock zu senden. Amazon Bedrock verarbeitet ihn mit dem Modell Claude 3 Sonnet und sendet eine zusammengefasste Version zurück.

Inhalt

- Voraussetzungen
- Schritt 1: Erstellen und konfigurieren Sie eine IAM-Rolle und einen App Studio-Connector
- Schritt 2: Erstellen Sie eine Anwendung
- Schritt 3: Erstellen und konfigurieren Sie eine Automatisierung
- Schritt 4: Seiten und Komponenten erstellen
  - Benennen Sie die Standardseite um
  - Fügen Sie der Seite Komponenten hinzu
  - Konfigurieren Sie die Seitenkomponenten
- Schritt 5: Veröffentlichen Sie die Anwendung in der Testumgebung
- (Optional) Bereinigen

Voraussetzungen

Bevor Sie beginnen, sollten Sie die folgenden Voraussetzungen überprüfen und erfüllen:

- Zugriff auf AWS App Studio. Beachten Sie, dass Sie in diesem Tutorial über die Administratorrolle verfügen müssen, um einen Connector zu erstellen.
- Optional: Lesen AWS App Studio-Konzepte und dannTutorial: Beginne mit dem Erstellen von einer leeren App, um sich mit wichtigen App Studio-Konzepten vertraut zu machen.

Schritt 1: Erstellen und konfigurieren Sie eine IAM-Rolle und einen App Studio-Connector

Um App Studio Zugriff auf Amazon Bedrock-Modelle zu gewähren, müssen Sie:

1. Aktivieren Sie die Amazon Bedrock-Modelle, die Sie in Ihrer App verwenden möchten. Für dieses Tutorial verwenden Sie Claude 3 Sonnet. Stellen Sie also sicher, dass Sie dieses Modell aktivieren.
2. Erstellen Sie eine IAM-Rolle mit den entsprechenden Berechtigungen für Amazon Bedrock.

3. Erstellen Sie einen App Studio-Connector mit der IAM-Rolle, die in Ihrer App verwendet wird.

Connect zu Amazon Bedrock herAusführliche Anweisungen finden Sie unter und kehren Sie zu diesem Tutorial zurück, nachdem Sie die Schritte ausgeführt und den Connector erstellt haben.

Schritt 2: Erstellen Sie eine Anwendung

Gehen Sie wie folgt vor, um in App Studio eine leere App zu erstellen, die Sie in die Textzusammenfassungs-App integrieren.

1. Melden Sie sich bei App Studio an.
2. Navigieren Sie zum Builder-Hub und wählen Sie + App erstellen.
3. Wählen Sie Von vorne beginnen.
4. Geben Sie im Feld App-Name einen Namen für Ihre App ein, z. **`Text Summarizer`** B.
5. Wenn Sie aufgefordert werden, Datenquellen oder einen Connector auszuwählen, wählen Sie für die Zwecke dieses Tutorials die Option Überspringen.
6. Wählen Sie Weiter aus, um fortzufahren.
7. (Optional): Sehen Sie sich das Video-Tutorial an, um einen schnellen Überblick über die Erstellung von Apps in App Studio zu erhalten.
8. Wählen Sie App bearbeiten, um zum Anwendungsstudio zu gelangen.

Schritt 3: Erstellen und konfigurieren Sie eine Automatisierung

Sie definieren die Logik und das Verhalten einer App Studio-App in Automatisierungen. Automatisierungen bestehen aus einzelnen Schritten, den sogenannten Aktionen, Parametern, mit denen Daten aus anderen Ressourcen an die Aktion übergeben werden, und einer Ausgabe, die von anderen Automatisierungen oder Komponenten verwendet werden kann. In diesem Schritt erstellen Sie eine Automatisierung, die die Interaktion mit Amazon Bedrock wie folgt abwickelt:

- Eingaben: Ein Parameter, um die Texteingabe vom Benutzer an die Automatisierung weiterzuleiten.
- Aktionen: Eine GenAI Prompt-Aktion, die die Texteingabe an Amazon Bedrock sendet und die ausgegebene Textzusammenfassung zurückgibt.
- Ausgaben: Eine Automatisierungsausgabe, die aus der verarbeiteten Zusammenfassung von Amazon Bedrock besteht und in Ihrer App verwendet werden kann.

Um eine Automatisierung zu erstellen und zu konfigurieren, die eine Aufforderung an Amazon Bedrock sendet und eine Zusammenfassung verarbeitet und zurücksendet

1. Wählen Sie oben auf der Leinwand die Registerkarte Automationen aus.
2. Wählen Sie + Automatisierung hinzufügen.
3. Wählen Sie im rechten Bereich Eigenschaften.
4. Aktualisieren Sie den Namen der Automatisierung, indem Sie das Stiftsymbol auswählen. Geben Sie ein **`InvokeBedrock`** und drücken Sie die Eingabetaste.
5. Fügen Sie der Automatisierung einen Parameter hinzu, der verwendet wird, um die Eingabeaufforderungseingabe des Benutzers an die Automatisierung weiterzuleiten, die in der Anfrage an Amazon Bedrock verwendet werden soll, indem Sie die folgenden Schritte ausführen:
   a. Wählen Sie auf der Leinwand im Parameterfeld + Hinzufügen aus.
   b. Geben Sie unter Name **`input`** ein.
   c. Geben Sie im Feld Beschreibung eine beliebige Beschreibung ein, z. **`Text to be sent to Amazon Bedrock`** B.
   d. Wählen Sie unter Typ die Option Zeichenfolge aus.
   e. Wählen Sie Hinzufügen, um den Parameter zu erstellen.
6. Fügen Sie eine GenAI Prompt-Aktion hinzu, indem Sie die folgenden Schritte ausführen:
   a. Wählen Sie im rechten Bereich Aktionen aus.
   b. Wählen Sie GenAI Prompt, um eine Aktion hinzuzufügen.
7. Konfigurieren Sie die Aktion, indem Sie die folgenden Schritte ausführen:
   a. Wählen Sie die Aktion auf der Arbeitsfläche aus, um das Eigenschaftenmenü auf der rechten Seite zu öffnen.
   b. Benennen Sie die Aktion um in, **`PromptBedrock`** indem Sie das Stiftsymbol wählen, den Namen eingeben und die Eingabetaste drücken.
   c. Wählen Sie unter Connector den Connector aus, der in erstellt wurdeSchritt 1: Erstellen und konfigurieren Sie eine IAM-Rolle und einen App Studio-Connector.
   d. Wählen Sie unter Modell das Amazon Bedrock-Modell aus, das Sie für die Verarbeitung der Aufforderung verwenden möchten. In diesem Tutorial wählen Sie Claude 3.5 Sonnet.
   e. Geben `{{params.input}}` Sie im Feld Benutzeraufforderung den Wert ein. Dies stellt den `input` Parameter dar, den Sie zuvor erstellt haben, und enthält die Texteingabe Ihrer App-Benutzer.

f. Geben Sie unter Systemaufforderung die Anweisungen zur Systemaufforderung ein, die Sie an Amazon Bedrock senden möchten. Geben Sie für dieses Tutorial Folgendes ein:

```
You are a highly efficient text summarizer. Provide a concise summary of the
 prompted text, capturing the key points and main ideas.
```

g. Wählen Sie „Einstellungen anfordern“, um es zu erweitern, und aktualisieren Sie die folgenden Felder:

- Geben Sie im Feld Temperatur den Wert ein`0`. Die Temperatur bestimmt die Zufälligkeit oder Kreativität der Ausgabe auf einer Skala von 0 bis 10. Je höher die Zahl, desto kreativer ist die Reaktion.
- Geben Sie im Feld Max. Tokens ein, `4096` um die Länge der Antwort zu begrenzen.

8. Die Ausgabe dieser Automatisierung ist der zusammengefasste Text, jedoch erzeugen Automatisierungen standardmäßig keine Ausgaben. Konfigurieren Sie die Automatisierung so, dass sie eine Automatisierungsausgabe erstellt, indem Sie die folgenden Schritte ausführen:

a. Wählen Sie in der linken Navigationsleiste die InvokeBedrockAutomatisierung aus.

b. Wählen Sie im rechten Eigenschaftenmenü unter Ausgabe die Option + Hinzufügen.

c. Geben **`{{results.PromptBedrock.text}}`** Sie im Feld Ausgabe den Wert ein. Dieser Ausdruck gibt den Inhalt der `processResults` Aktion zurück.

Schritt 4: Seiten und Komponenten erstellen

In App Studio stellt jede Seite einen Bildschirm der Benutzeroberfläche (UI) Ihrer Anwendung dar, mit dem Ihre Benutzer interagieren werden. Auf diesen Seiten können Sie verschiedene Komponenten wie Tabellen, Formulare, Schaltflächen und mehr hinzufügen, um das gewünschte Layout und die gewünschte Funktionalität zu erstellen.

Benennen Sie die Standardseite um

Die Textzusammenfassungs-App in diesem Tutorial wird nur eine Seite enthalten. Neu erstellte Anwendungen verfügen über eine Standardseite, sodass Sie diese umbenennen, anstatt eine hinzuzufügen.

Um die Standardseite umzubenennen

1. Wählen Sie im Navigationsmenü der oberen Leiste die Option Seiten aus.

2. Wählen Sie im linken Bereich Page1 und dann im rechten Bereich das Eigenschaften-Panel aus.
3. Wählen Sie das Stiftsymbol, geben Sie die Eingabetaste ein und drücken Sie **`TextSummarizationTool`** die Eingabetaste.
4. Geben Sie in der Navigationsbeschriftung ein**`TextSummarizationTool`**.

Fügen Sie der Seite Komponenten hinzu

Für dieses Tutorial hat die Textzusammenfassungs-App eine Seite, die die folgenden Komponenten enthält:

- Eine Texteingabekomponente, mit der Endbenutzer eine Aufforderung zur Zusammenfassung eingeben.
- Eine Button-Komponente, die verwendet wird, um die Aufforderung an Amazon Bedrock zu senden.
- Eine Textbereichskomponente, die die Zusammenfassung von Amazon Bedrock anzeigt.

Fügen Sie der Seite eine Texteingabekomponente hinzu, über die Benutzer eine Texteingabeaufforderung zur Zusammenfassung eingeben.

Um eine Texteingabekomponente hinzuzufügen

1. Suchen Sie im rechten Bedienfeld „Komponenten“ die Texteingabekomponente und ziehen Sie sie auf die Leinwand.
2. Wählen Sie die Texteingabe auf der Leinwand aus, um sie auszuwählen.
3. Aktualisieren Sie im Eigenschaftenbereich auf der rechten Seite die folgenden Einstellungen:
    a. Wählen Sie das Stiftsymbol, in das Sie die Texteingabe umbenennen möchten**`inputPrompt`**.
    b. Geben Sie im Feld Bezeichnung den Wert ein**`Prompt`**.
    c. Geben **`Enter text to be summarized`** Sie im Feld Platzhalter den Wert ein.

Fügen Sie nun eine Button-Komponente hinzu, mit der Benutzer die Aufforderung an Amazon Bedrock senden können.

Um eine Button-Komponente hinzuzufügen

1. Suchen Sie im rechten Bedienfeld „Komponenten“ die Button-Komponente und ziehen Sie sie auf die Arbeitsfläche.
2. Wählen Sie die Schaltfläche auf der Leinwand aus, um sie auszuwählen.
3. Aktualisieren Sie im Eigenschaftenbereich auf der rechten Seite die folgenden Einstellungen:
    a. Wählen Sie das Stiftsymbol, in das Sie die Schaltfläche umbenennen möchten**`sendButton`**.
    b. Geben Sie im Feld Schaltflächenbezeichnung den Wert ein**`Send`**.

Fügen Sie nun eine Textbereichskomponente hinzu, die die von Amazon Bedrock zurückgegebene Zusammenfassung anzeigt.

Um eine Textbereichskomponente hinzuzufügen

1. Suchen Sie im rechten Bedienfeld „Komponenten“ die Textbereichskomponente und ziehen Sie sie auf die Leinwand.
2. Wählen Sie den Textbereich auf der Leinwand aus, um ihn auszuwählen.
3. Aktualisieren Sie im Eigenschaftenbereich auf der rechten Seite die folgenden Einstellungen:
    a. Wählen Sie das Stiftsymbol, in das Sie die Schaltfläche umbenennen möchten**`textSummary`**.
    b. Geben Sie im Feld Bezeichnung den Wert ein**`Summary`**.

Konfigurieren Sie die Seitenkomponenten

Da die App nun eine Seite mit Komponenten enthält, besteht der nächste Schritt darin, die Komponenten so zu konfigurieren, dass sie das entsprechende Verhalten ausführen. Um eine Komponente, z. B. eine Schaltfläche, so zu konfigurieren, dass sie Aktionen ausführt, wenn mit ihr interagiert wird, müssen Sie ihr einen Auslöser hinzufügen. Für die App in diesem Tutorial fügen Sie der `sendButton` Schaltfläche zwei Auslöser hinzu, um Folgendes zu tun:

- Der erste Trigger sendet den Text in der `textPrompt` Komponente zur Analyse an Amazon Bedrock.
- Der zweite Auslöser zeigt die von Amazon Bedrock zurückgegebene Zusammenfassung in der `textSummary` Komponente an.

Um einen Auslöser hinzuzufügen, der die Aufforderung an Amazon Bedrock sendet

1. Wählen Sie die Schaltfläche auf der Leinwand, um sie auszuwählen.
2. Wählen Sie im Eigenschaftenfenster auf der rechten Seite im Bereich Auslöser die Option + Hinzufügen.
3. Wählen Sie „Automatisierung aufrufen“.
4. Wählen Sie den einen InvokeAutomationTrigger aus, der erstellt wurde, um ihn zu konfigurieren.
5. Geben Sie im Feld Aktionsname den Wert ein**`invokeBedrockAutomation`**.
6. Wählen Sie unter Invoke Automation die InvokeBedrockAutomatisierung aus, die zuvor erstellt wurde.
7. Geben Sie im Parameterfeld in den Eingabeparameter, der zuvor erstellt wurde, den Text ein**`{{ui.inputPrompt.value}}`**, der den Inhalt der `inputPrompt` Texteingabekomponente übergibt.
8. Klicken Sie oben im Bedienfeld auf den Linkspfeil, um zum Menü mit den Komponenteneigenschaften zurückzukehren.

Jetzt haben Sie einen Trigger konfiguriert, der die Automatisierung zum Senden einer Anfrage an Amazon Bedrock aufruft, wenn auf die Schaltfläche geklickt wird. Der nächste Schritt besteht darin, einen zweiten Trigger zu konfigurieren, der die Ergebnisse in der Komponente anzeigt. `textSummary`

Um einen Trigger hinzuzufügen, der die Amazon Bedrock-Ergebnisse in der Textbereichskomponente anzeigt

1. Wählen Sie auf der rechten Seite des Eigenschaftenfensters der Schaltfläche im Bereich Auslöser die Option + Hinzufügen.
2. Wählen Sie „Komponentenaktion ausführen“.
3. Wählen Sie den Runcomponentaction1-Trigger aus, der für die Konfiguration erstellt wurde.
4. Geben Sie im Feld Aktionsname den Wert ein. **`setTextSummary`**
5. Wählen Sie unter Komponente die TextSummary-Komponente aus.
6. Wählen Sie unter Aktion die Option Wert festlegen aus.
7. Geben Sie im Feld Wert setzen auf ein**`{{results.invokeBedrockAutomation}}`**.

Schritt 5: Veröffentlichen Sie die Anwendung in der Testumgebung

Während der Entwicklung einer App empfiehlt es sich in der Regel, eine Vorschau der App anzusehen, um zu sehen, wie sie aussieht, und erste Tests ihrer Funktionalität durchzuführen. Da Anwendungen in der Vorschauumgebung jedoch nicht mit externen Diensten interagieren, veröffentlichen Sie die App stattdessen in der Testumgebung, um das Senden von Anfragen und das Empfangen von Antworten von Amazon Bedrock testen zu können.

Um Ihre App in der Testumgebung zu veröffentlichen

1. Wählen Sie in der oberen rechten Ecke des App Builders die Option Veröffentlichen aus.
2. Fügen Sie eine Versionsbeschreibung für die Testumgebung hinzu.
3. Überprüfen Sie das Kontrollkästchen für das SLA und aktivieren Sie es.
4. Wählen Sie Starten. Die Veröffentlichung kann bis zu 15 Minuten dauern.
5. (Optional) Wenn Sie bereit sind, können Sie anderen Zugriff gewähren, indem Sie „Teilen“ wählen und den Anweisungen folgen. Weitere Informationen zum Teilen von App Studio-Apps finden Sie unterGemeinsame Nutzung veröffentlichter Anwendungen.

Wählen Sie nach dem Testen Ihrer Anwendung erneut Veröffentlichen, um die Anwendung in die Produktionsumgebung hochzustufen. Beachten Sie, dass Apps in der Produktionsumgebung für Endbenutzer erst verfügbar sind, wenn sie gemeinsam genutzt werden. Weitere Informationen zu den verschiedenen Anwendungsumgebungen finden Sie unterAnwendungsumgebungen.

(Optional) Bereinigen

Sie haben das Tutorial nun erfolgreich abgeschlossen und mit Amazon Bedrock eine App zur Textzusammenfassung in App Studio erstellt. Sie können Ihre App weiterhin verwenden oder die Ressourcen bereinigen, die in diesem Tutorial erstellt wurden. Die folgende Liste enthält eine Liste der Ressourcen, die bereinigt werden müssen:

- Der in App Studio erstellte Amazon Bedrock-Connector. Weitere Informationen finden Sie unter Konnektoren anzeigen, bearbeiten und löschen.
- Die Textzusammenfassungs-App in App Studio. Weitere Informationen finden Sie unter Löschen einer Anwendung.
- Die in der IAM-Konsole erstellte IAM-Rolle. Weitere Informationen finden Sie im AWS Identity and Access Management Benutzerhandbuch unter Löschen von Rollen oder Instanzprofilen.

- Wenn Sie Modellzugriff für Claude 3 Sonnet beantragt haben und den Zugriff rückgängig machen möchten, finden Sie weitere Informationen unter Zugriff auf Amazon Bedrock Foundation-Modelle verwalten im Amazon Bedrock-Benutzerhandbuch.

Interaktion mit Amazon Simple Storage Service mit Komponenten und Automatisierungen

Sie können verschiedene Amazon S3 S3-Operationen von einer App Studio-App aus aufrufen. Sie könnten beispielsweise ein einfaches Admin-Panel erstellen, um Ihre Benutzer und Bestellungen zu verwalten und Ihre Medien von Amazon S3 aus anzuzeigen. Sie können zwar jeden Amazon S3 S3-Vorgang mit der AWS Aktion Aufrufen aufrufen, es gibt jedoch vier spezielle Amazon S3 S3-Aktionen, die Sie zu Automatisierungen in Ihrer App hinzufügen können, um allgemeine Operationen an Amazon S3 S3-Buckets und -Objekten durchzuführen. Die vier Aktionen und ihre Operationen lauten wie folgt:

- Objekt platzieren: Verwendet den `Amazon S3 PutObject` Vorgang, um ein Objekt zu einem Amazon S3 S3-Bucket hinzuzufügen.
- Objekt abrufen: Verwendet den `Amazon S3 GetObject` Vorgang, um ein Objekt aus einem Amazon S3 S3-Bucket abzurufen.
- Objekte auflisten: Verwendet den `Amazon S3 ListObjects` Vorgang, um Objekte in einem Amazon S3 S3-Bucket aufzulisten.
- Objekt löschen: Verwendet den `Amazon S3 DeleteObject` Vorgang, um ein Objekt aus einem Amazon S3 S3-Bucket zu löschen.

Zusätzlich zu den Aktionen gibt es eine S3-Upload-Komponente, die Sie Seiten in Anwendungen hinzufügen können. Benutzer können diese Komponente verwenden, um eine Datei zum Hochladen auszuwählen, und die Komponente ruft `Amazon S3 PutObject` auf, um die Datei in den konfigurierten Bucket und Ordner hochzuladen. In diesem Tutorial wird diese Komponente anstelle der eigenständigen Automatisierungsaktion „Objekt platzieren“ verwendet. (Die eigenständige Aktion sollte in komplexeren Szenarien verwendet werden, die zusätzliche Logik oder Aktionen beinhalten, die vor oder nach dem Hochladen ausgeführt werden müssen.)

Voraussetzungen

In diesem Handbuch wird davon ausgegangen, dass Sie die folgenden Voraussetzungen erfüllt haben:

1. Erstellung und Konfiguration eines Amazon S3-Buckets, einer IAM-Rolle und -Richtlinie sowie eines Amazon S3 S3-Connectors, um Amazon S3 erfolgreich in App Studio zu integrieren. Um einen Connector zu erstellen, benötigen Sie die Administratorrolle. Weitere Informationen finden Sie unter Connect zu Amazon Simple Storage Service (Amazon S3) her.

Erstellen Sie eine leere Anwendung

Erstellen Sie eine leere Anwendung, die Sie in diesem Handbuch verwenden können, indem Sie die folgenden Schritte ausführen.

Um eine leere Anwendung zu erstellen

1. Wählen Sie im Navigationsbereich Meine Anwendungen aus.
2. Wählen Sie + App erstellen.
3. Geben Sie Ihrer Anwendung im Dialogfeld „App erstellen“ einen Namen, wählen Sie „Von vorne beginnen“ und dann „Weiter“.
4. Wählen Sie im Dialogfeld Mit vorhandenen Daten verbinden die Option Überspringen aus, um die Anwendung zu erstellen.
5. Wählen Sie App bearbeiten, um zur Arbeitsfläche Ihrer neuen App zu gelangen, wo Sie Komponenten, Automatisierungen und Daten verwenden können, um das Aussehen und die Funktion Ihrer Anwendung zu konfigurieren.

Seiten erstellen

Erstellen Sie drei Seiten in Ihrer Anwendung, um Informationen zu sammeln oder anzuzeigen.

Um Seiten zu erstellen

1. Wählen Sie bei Bedarf den Tab Seiten oben auf der Leinwand aus.
2. In der linken Navigation befindet sich eine einzelne Seite, die mit Ihrer App erstellt wurde. Wählen Sie zweimal + Hinzufügen, um zwei weitere Seiten zu erstellen. Im Navigationsbereich sollten insgesamt drei Seiten angezeigt werden.
3. Aktualisieren Sie den Namen der Seite Page1, indem Sie die folgenden Schritte ausführen:
   a. Wählen Sie das Ellipsensymbol und dann Seiteneigenschaften.
   b. Wählen Sie im rechten Eigenschaftenmenü das Stiftsymbol, um den Namen zu bearbeiten.

c. Geben Sie die Eingabetaste ein **`FileList`** und drücken Sie die Eingabetaste.

4. Wiederholen Sie die vorherigen Schritte, um die zweite und dritte Seite wie folgt zu aktualisieren:

   - Benennen Sie Page2 um in. **`UploadFile`**
   - Benennen Sie Seite3 um in. **`FailUpload`**

Jetzt sollte Ihre App drei Seiten mit den Namen FileList, und haben UploadFileFailUpload, die im Seitenmenü auf der linken Seite angezeigt werden.

Als Nächstes erstellen und konfigurieren Sie die Automatisierungen, die mit Amazon S3 interagieren.

Automatisierungen erstellen und konfigurieren

Erstellen Sie die Automatisierungen Ihrer Anwendung, die mit Amazon S3 interagieren. Verwenden Sie die folgenden Verfahren, um die folgenden Automatisierungen zu erstellen:

- Eine GetFiles-Automatisierung, die die Objekte in Ihrem Amazon S3 S3-Bucket auflistet, die zum Füllen einer Tabellenkomponente verwendet werden.
- Eine DeleteFile-Automatisierung, die ein Objekt aus Ihrem Amazon S3 S3-Bucket löscht, das verwendet wird, um einer Tabellenkomponente eine Löschschaltfläche hinzuzufügen.
- Eine ViewFile-Automatisierung, die ein Objekt aus Ihrem Amazon S3 S3-Bucket abruft und anzeigt. Dies wird verwendet, um weitere Details zu einem einzelnen Objekt anzuzeigen, das aus einer Tabellenkomponente ausgewählt wurde.

Erstellen Sie eine Automatisierung **`getFiles`**

Erstellen Sie eine Automatisierung, die die Dateien in einem bestimmten Amazon S3 S3-Bucket auflistet.

1. Wählen Sie oben auf der Leinwand die Registerkarte Automationen aus.
2. Wählen Sie + Automatisierung hinzufügen.
3. Wählen Sie im rechten Bereich Eigenschaften aus.
4. Aktualisieren Sie den Namen der Automatisierung, indem Sie das Stiftsymbol auswählen. Geben Sie ein **`getFiles`** und drücken Sie die Eingabetaste.
5. Fügen Sie eine Aktion Objekte auflisten hinzu, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie im rechten Bereich Aktionen aus.

b. Wählen Sie Objekte auflisten, um eine Aktion hinzuzufügen. Die Aktion sollte benannt werden`ListObjects1`.

6. Konfigurieren Sie die Aktion, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie die Aktion auf der Arbeitsfläche aus, um das Eigenschaftenmenü auf der rechten Seite zu öffnen.

   b. Wählen Sie für Connector den Amazon S3 S3-Connector aus, den Sie aus den Voraussetzungen erstellt haben.

   c. Geben Sie für Konfiguration den folgenden Text ein und *`bucket_name`* ersetzen Sie ihn durch den Bucket, den Sie in den Voraussetzungen erstellt haben:

```
{
  "Bucket": "bucket_name",
  "Prefix": ""
}
```

   > **Note**
   >
   > Sie können das `Prefix` Feld verwenden, um die Antwort auf Objekte zu beschränken, die mit der angegebenen Zeichenfolge beginnen.

7. Die Ausgabe dieser Automatisierung wird verwendet, um eine Tabellenkomponente mit Objekten aus Ihrem Amazon S3 S3-Bucket zu füllen. Standardmäßig erzeugen Automatisierungen jedoch keine Ausgaben. Konfigurieren Sie die Automatisierung so, dass sie eine Automatisierungsausgabe erstellt, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie in der linken Navigationsleiste die GetFiles-Automatisierung aus.

   b. Wählen Sie im rechten Eigenschaftenmenü unter Automationsausgabe die Option + Ausgabe hinzufügen.

   c. Geben **`{{results.ListObjects1.Contents}}`** Sie für Ausgabe den Wert ein. Dieser Ausdruck gibt den Inhalt der Aktion zurück und kann nun zum Auffüllen einer Tabellenkomponente verwendet werden.

Erstellen Sie eine Automatisierung **`deleteFile`**

Erstellen Sie eine Automatisierung, die ein Objekt aus einem angegebenen Amazon S3 S3-Bucket löscht.

1. Wählen Sie im Automationsbereich auf der linken Seite die Option + Hinzufügen.
2. Wählen Sie + Automatisierung hinzufügen.
3. Wählen Sie im rechten Bereich Eigenschaften aus.
4. Aktualisieren Sie den Namen der Automatisierung, indem Sie das Stiftsymbol auswählen. Geben Sie ein **`deleteFile`** und drücken Sie die Eingabetaste.
5. Fügen Sie einen Automatisierungsparameter hinzu, der verwendet wird, um Daten an eine Automatisierung zu übergeben, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie im rechten Eigenschaftenmenü unter Automatisierungsparameter die Option + Hinzufügen aus.

   b. Wählen Sie das Stiftsymbol, um den Automatisierungsparameter zu bearbeiten. Aktualisieren Sie den Parameternamen auf **`fileName`** und drücken Sie die Eingabetaste.
6. Fügen Sie eine Aktion „Objekt löschen“ hinzu, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie im rechten Bereich Aktionen aus.

   b. Wählen Sie Objekt löschen, um eine Aktion hinzuzufügen. Die Aktion sollte benannt werden`DeleteObject1`.
7. Konfigurieren Sie die Aktion, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie die Aktion auf der Arbeitsfläche aus, um das Eigenschaftenmenü auf der rechten Seite zu öffnen.

   b. Wählen Sie für Connector den Amazon S3 S3-Connector aus, den Sie aus den Voraussetzungen erstellt haben.

   c. Geben Sie für Konfiguration den folgenden Text ein und *`bucket_name`* ersetzen Sie ihn durch den Bucket, den Sie in den Voraussetzungen erstellt haben:

```
{
  "Bucket": "bucket_name",
  "Key": params.fileName
}
```

Erstellen Sie eine **`viewFile`** Automatisierung

Erstellen Sie eine Automatisierung, die ein einzelnes Objekt aus einem bestimmten Amazon S3 S3-Bucket abruft. Später werden Sie diese Automatisierung mit einer Datei-Viewer-Komponente konfigurieren, um das Objekt anzuzeigen.

1. Wählen Sie im Automationsbereich auf der linken Seite die Option + Hinzufügen.
2. Wählen Sie + Automatisierung hinzufügen.
3. Wählen Sie im rechten Bereich Eigenschaften aus.
4. Aktualisieren Sie den Namen der Automatisierung, indem Sie das Stiftsymbol auswählen. Geben Sie ein **`viewFile`** und drücken Sie die Eingabetaste.
5. Fügen Sie einen Automatisierungsparameter hinzu, der verwendet wird, um Daten an eine Automatisierung zu übergeben, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie im rechten Eigenschaftenmenü unter Automatisierungsparameter die Option + Hinzufügen aus.

   b. Wählen Sie das Stiftsymbol, um den Automatisierungsparameter zu bearbeiten. Aktualisieren Sie den Parameternamen auf **`fileName`** und drücken Sie die Eingabetaste.
6. Fügen Sie eine Aktion „Objekt abrufen“ hinzu, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie im rechten Bereich Aktionen aus.

   b. Wählen Sie Objekt abrufen, um eine Aktion hinzuzufügen. Die Aktion sollte benannt werden`GetObject1`.
7. Konfigurieren Sie die Aktion, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie die Aktion auf der Arbeitsfläche aus, um das Eigenschaftenmenü auf der rechten Seite zu öffnen.

   b. Wählen Sie für Connector den Amazon S3 S3-Connector aus, den Sie aus den Voraussetzungen erstellt haben.

   c. Geben Sie für Konfiguration den folgenden Text ein und *`bucket_name`* ersetzen Sie ihn durch den Bucket, den Sie in den Voraussetzungen erstellt haben:

   ```
   {
     "Bucket": "bucket_name",
     "Key": params.fileName
   }
   ```
8. Standardmäßig erzeugen Automatisierungen keine Ausgaben. Konfigurieren Sie die Automatisierung so, dass sie eine Automatisierungsausgabe erstellt, indem Sie die folgenden Schritte ausführen:

   a. Wählen Sie in der linken Navigationsleiste die ViewFile-Automatisierung aus.

b. Wählen Sie im rechten Eigenschaftenmenü unter Automationsausgabe die Option + Ausgabe hinzufügen.

c. Geben **`{{results.GetObject1.Body.transformToWebStream()}}`** Sie für Ausgabe den Wert ein. Dieser Ausdruck gibt den Inhalt der Aktion zurück.

> **Note**
>
> Sie können die Antwort von auf folgende `S3 GetObject` Weise lesen:
>
> - `transformToWebStream`: Gibt einen Stream zurück, der zum Abrufen der Daten verbraucht werden muss. Wenn sie als Automatisierungsausgabe verwendet wird, verarbeitet die Automatisierung dies, und die Ausgabe kann als Datenquelle für eine Bild- oder PDF-Viewer-Komponente verwendet werden. Sie kann auch als Eingabe für eine andere Operation verwendet werden, z. `S3 PutObject`
> - `transformToString`: Gibt die Rohdaten der Automatisierung zurück und sollte in einer JavaScript Aktion verwendet werden, wenn Ihre Dateien Textinhalte wie JSON-Daten enthalten. Muss abgewartet werden, zum Beispiel: `await results.GetObject1.Body.transformToString();`
> - `transformToByteArray`: Gibt ein Array von 8-Bit-Ganzzahlen ohne Vorzeichen zurück. Diese Antwort dient dem Zweck eines Byte-Arrays, das das Speichern und Bearbeiten von Binärdaten ermöglicht. Muss abgewartet werden, zum Beispiel: `await results.GetObject1.Body.transformToByteArray();`

Als Nächstes fügen Sie den Seiten, die Sie zuvor erstellt haben, Komponenten hinzu und konfigurieren sie mit Ihren Automatisierungen, sodass Benutzer Ihre App zum Anzeigen und Löschen von Dateien verwenden können.

Fügen Sie Seitenkomponenten hinzu und konfigurieren Sie sie

Nachdem Sie nun die Automatisierungen erstellt haben, die die Geschäftslogik und Funktionalität Ihrer App definieren, werden Sie Komponenten erstellen und beide miteinander verbinden.

Fügen Sie der Seite Komponenten hinzu FileList

Die FileListSeite, die Sie zuvor erstellt haben, wird verwendet, um eine Liste der Dateien im konfigurierten Amazon S3 S3-Bucket und weitere Details zu jeder Datei anzuzeigen, die aus der Liste ausgewählt wurde. Gehen Sie dazu wie folgt vor:

1. Erstellen Sie eine Tabellenkomponente, um die Liste der Dateien anzuzeigen. Sie konfigurieren die Zeilen der Tabelle so, dass sie mit der Ausgabe der GetFiles-Automatisierung gefüllt werden, die Sie zuvor erstellt haben.
2. Erstellen Sie eine PDF-Viewer-Komponente, um ein einzelnes PDF anzuzeigen. Sie konfigurieren die Komponente so, dass sie eine aus der Tabelle ausgewählte Datei anzeigt, und verwenden dabei die ViewFile-Automatisierung, die Sie zuvor erstellt haben, um die Datei aus Ihrem Bucket abzurufen.

Um Komponenten zur Seite hinzuzufügen FileList

1. Wählen Sie oben auf der Leinwand den Tab Seiten.
2. Wählen Sie im Seitenbedienfeld auf der linken FileListSeite die Seite aus.
3. Suchen Sie auf der rechten Komponentenseite nach der Tabellenkomponente und ziehen Sie sie in die Mitte der Arbeitsfläche.
4. Wählen Sie die Tabellenkomponente aus, die Sie der Seite gerade hinzugefügt haben.
5. Wählen Sie im rechten Eigenschaftenmenü die Dropdownliste Quelle und dann Automatisierung aus.
6. Wählen Sie das Drop-down-Menü Automatisierung und wählen Sie die GetFiles-Automatisierung aus. Die Tabelle verwendet die Ausgabe der GetFiles-Automatisierung als Inhalt.
7. Fügen Sie eine Spalte hinzu, die mit dem Namen der Datei gefüllt werden soll.
    a. Wählen Sie auf der rechten Seite im Eigenschaftenmenü neben Spalten die Option + Hinzufügen.
    b. Wählen Sie das Pfeilsymbol rechts neben der Spalte Spalte1, die gerade hinzugefügt wurde.
    c. Benennen Sie die Spalte für Spaltenbezeichnung in um. **`Filename`**
    d. Geben Sie für Wert **`{{currentRow.Key}}`** ein.
    e. Wählen Sie das Pfeilsymbol oben im Bedienfeld, um zum Hauptfenster „Eigenschaften“ zurückzukehren.
8. Fügen Sie eine Tabellenaktion hinzu, um die Datei nacheinander zu löschen.
    a. Wählen Sie auf der rechten Seite im Eigenschaftenmenü neben Aktionen die Option + Hinzufügen aus.
    b. Benennen Sie Button unter Aktionen in um**`Delete`**.

c. Wählen Sie das Pfeilsymbol rechts neben der Aktion Löschen, die gerade umbenannt wurde.

d. Wählen Sie unter Beim Klicken die Option + Aktion hinzufügen und dann Automatisierung aufrufen aus.

e. Wählen Sie die Aktion aus, die hinzugefügt wurde, um sie zu konfigurieren.

f. Für Aktionsname geben Sie **`DeleteRecord`** ein.

g. Wählen Sie unter Automatisierung aufrufen die Option aus**`deleteFile`**.

h. Geben **`{{currentRow.Key}}`** Sie in das Textfeld für den Parameter ein.

i. Geben Sie für Wert **`{{currentRow.Key}}`** ein.

9. Wählen Sie im rechten Bereich „Komponenten“, um das Komponentenmenü aufzurufen. Es gibt zwei Möglichkeiten, Dateien anzuzeigen:

   - Ein Bildbetrachter zum Anzeigen von Dateien mit der `.jpg` Erweiterung `.png.jpeg`, oder.
   - Eine PDF-Viewer-Komponente zum Anzeigen von PDF-Dateien.

   In diesem Tutorial werden Sie die PDF-Viewer-Komponente hinzufügen und konfigurieren.

10. Fügen Sie die PDF-Viewer-Komponente hinzu.

    a. Suchen Sie auf der rechten Komponentenseite nach der PDF-Viewer-Komponente und ziehen Sie sie auf die Arbeitsfläche unter der Tabellenkomponente.

    b. Wählen Sie die PDF-Viewer-Komponente aus, die gerade hinzugefügt wurde.

    c. Wählen Sie im rechten Eigenschaftenmenü die Dropdownliste Quelle und dann Automatisierung aus.

    d. Wählen Sie das Drop-down-Menü Automatisierung und dann die ViewFile-Automatisierung aus. Die Tabelle verwendet die Ausgabe der ViewFile-Automatisierung als Inhalt.

    e. Geben **`{{ui.table1.selectedRow["Filename"]}}`** Sie in das Parametertextfeld ein.

    f. Im rechten Bereich befindet sich auch ein Feld für den Dateinamen. Der Wert dieses Felds wird als Header für die PDF-Viewer-Komponente verwendet. Geben Sie denselben Text wie im vorherigen Schritt ein:**`{{ui.table1.selectedRow["Filename"]}}`**.

Fügen Sie der UploadFileSeite Komponenten hinzu

Die UploadFileSeite wird eine Dateiauswahl enthalten, mit der Sie eine Datei auswählen und in den konfigurierten Amazon S3 S3-Bucket hochladen können. Sie fügen der Seite die S3-Upload-Komponente hinzu, mit der Benutzer eine Datei auswählen und hochladen können.

1. Wählen Sie im Seitenbereich auf der linken UploadFileSeite die Seite aus.
2. Suchen Sie auf der rechten Komponentenseite nach der S3-Upload-Komponente und ziehen Sie sie in die Mitte der Arbeitsfläche.
3. Wählen Sie die S3-Upload-Komponente aus, die Sie der Seite gerade hinzugefügt haben.
4. Konfigurieren Sie im rechten Eigenschaftenmenü die Komponente:

    a. Wählen Sie in der Dropdownliste Connector den Amazon S3 S3-Connector aus, der in den Voraussetzungen erstellt wurde.
    b. Geben Sie für Bucket den Namen Ihres Amazon S3 S3-Buckets ein.
    c. Geben Sie als Dateiname ein**`{{ui.s3Upload1.files[0]?.nameWithExtension}}`**.
    d. Geben Sie unter Max. Dateigröße den Wert **`5`** in das Textfeld ein und stellen Sie sicher, dass dieser Wert in der Dropdownliste ausgewählt **`MB`** ist.
    e. Fügen Sie im Abschnitt Trigger Aktionen hinzu, die nach erfolgreichen oder erfolglosen Uploads ausgeführt werden, indem Sie die folgenden Schritte ausführen:

        Um eine Aktion hinzuzufügen, die nach erfolgreichen Uploads ausgeführt wird:

        1. Wählen Sie unter Bei Erfolg die Option + Aktion hinzufügen und anschließend Navigieren aus.
        2. Wählen Sie die Aktion aus, die hinzugefügt wurde, um sie zu konfigurieren.
        3. Wählen Sie als Navigationstyp die Option Seite aus.
        4. Wählen Sie für Navigieren zu die Option**`FileList`**.
        5. Wählen Sie das Pfeilsymbol oben im Bedienfeld, um zum Hauptfenster „Eigenschaften“ zurückzukehren.

        Um eine Aktion hinzuzufügen, die nach erfolglosen Uploads ausgeführt wird:

        1. Wählen Sie unter Bei einem Fehler die Option + Aktion hinzufügen und anschließend Navigieren aus.
        2. Wählen Sie die Aktion aus, die hinzugefügt wurde, um sie zu konfigurieren.

3. Wählen Sie als Navigationstyp die Option Seite aus.
4. Wählen Sie für Navigieren zu die Option**`FailUpload`**.
5. Wählen Sie das Pfeilsymbol oben im Bedienfeld, um zum Hauptfenster „Eigenschaften“ zurückzukehren.

Fügen Sie der FailUploadSeite Komponenten hinzu

Die FailUploadSeite ist eine einfache Seite mit einem Textfeld, das Benutzer darüber informiert, dass ihr Upload fehlgeschlagen ist.

1. Wählen Sie im Seitenmenü auf der linken FailUploadSeite die Seite aus.
2. Suchen Sie auf der rechten Komponentenseite nach der Textkomponente und ziehen Sie sie in die Mitte der Leinwand.
3. Wählen Sie die Textkomponente aus, die Sie der Seite gerade hinzugefügt haben.
4. Geben Sie im rechten Eigenschaftenmenü im Feld Wert den Text ein**`Failed to upload, try again`**.

Aktualisieren Sie Ihre App-Sicherheitseinstellungen

Jede Anwendung in App Studio verfügt über Inhaltssicherheitseinstellungen, mit denen Sie externe Medien oder Ressourcen einschränken oder festlegen können, in welche Domänen in Amazon S3 Sie Objekte hochladen können. Die Standardeinstellung ist das Blockieren aller Domains. Um Objekte aus Ihrer Anwendung nach Amazon S3 hochzuladen, müssen Sie die Einstellung aktualisieren, um die Domains zuzulassen, in die Sie Objekte hochladen möchten.

Um Domains für das Hochladen von Objekten auf Amazon S3 zuzulassen

1. Wählen Sie den Tab App-Einstellungen.
2. Wählen Sie den Tab Content Security Settings.
3. Wählen Sie für Quelle verbinden die Option Alle Verbindungen zulassen aus.
4. Wählen Sie Speichern.

Nächste Schritte: Zeigen Sie eine Vorschau der Anwendung an und veröffentlichen Sie sie zum Testen

Die Anwendung ist jetzt bereit zum Testen. Weitere Informationen zur Vorschau und Veröffentlichung von Anwendungen finden Sie unterAnwendungen in der Vorschau anzeigen, veröffentlichen und teilen.

Lambda-Funktionen in einer App Studio-App aufrufen

Dieses Tutorial zeigt Ihnen, wie Sie App Studio mit Lambda verbinden und Lambda-Funktionen aus Ihren Apps aufrufen.

Voraussetzungen

In diesem Handbuch wird davon ausgegangen, dass Sie die folgenden Voraussetzungen erfüllt haben:

1. Eine App Studio-App erstellt. Wenn Sie noch keine haben, können Sie eine leere App erstellen, die Sie im Tutorial verwenden können. Weitere Informationen finden Sie unter Erstellen einer Anwendung.

> **Note**
>
> Sie benötigen zwar keine Lambda-Funktion, um diesem Tutorial zu folgen und zu lernen, wie man sie konfiguriert, aber es kann hilfreich sein, eine zu haben, um sicherzustellen, dass Sie die App korrekt konfiguriert haben. Dieses Tutorial enthält keine Informationen zum Erstellen von Lambda-Funktionen. Weitere Informationen finden Sie im AWS Lambda Entwicklerhandbuch.

Erstellen Sie einen Lambda-Konnektor

Um Lambda-Funktionen in Ihrer App Studio-App verwenden zu können, müssen Sie einen Connector verwenden, um App Studio mit Lambda zu verbinden, um Zugriff auf Ihre Funktionen zu gewähren. Sie müssen Administrator sein, um Connectors in App Studio erstellen zu können. Weitere Informationen zum Erstellen von Lambda-Konnektoren, einschließlich der Schritte zum Erstellen eines solchen, finden Sie unterVerbinden mit AWS Lambda.

Erstellen und konfigurieren Sie eine Automatisierung

Automatisierungen werden verwendet, um die Logik Ihrer Anwendung zu definieren und bestehen aus Aktionen. Um eine Lambda-Funktion in Ihrer App aufzurufen, fügen Sie zunächst eine Aktion „Lambda aufrufen“ zu einer Automatisierung hinzu und konfigurieren sie. Gehen Sie wie folgt vor, um eine Automatisierung zu erstellen und ihr die Aktion „Lambda aufrufen“ hinzuzufügen.

1. Wählen Sie bei der Bearbeitung Ihrer App den Tab Automatisierungen.
2. Wählen Sie + Automatisierung hinzufügen.
3. Wählen Sie im rechten Aktionsmenü die Option Lambda aufrufen, um den Schritt zu Ihrer Automatisierung hinzuzufügen.
4. Wählen Sie den neuen Lambda-Schritt auf der Arbeitsfläche aus, um seine Eigenschaften anzuzeigen und zu konfigurieren.
5. Konfigurieren Sie den Schritt im rechten Eigenschaftenmenü, indem Sie die folgenden Schritte ausführen:

    a. Wählen Sie unter Connector den Connector aus, der erstellt wurde, um App Studio mit Ihren Lambda-Funktionen zu verbinden.

    b. Geben Sie im Feld Funktionsname den Namen Ihrer Lambda-Funktion ein.

    c. Geben Sie im Feld Funktionsereignis das Ereignis ein, das an die Lambda-Funktion übergeben werden soll. In der folgenden Liste sind einige gängige Anwendungsfälle aufgeführt:

    - Übergeben des Werts eines Automatisierungsparameters, z. B. eines Dateinamens oder einer anderen Zeichenfolge: `varName: params.paramName`
    - Übergabe des Ergebnisses einer vorherigen Aktion: `varName: results.actionName1.data[0].fieldName`
    - Wenn Sie innerhalb einer Loop-Aktion eine Aktion „Lambda aufrufen“ hinzufügen, können Sie Felder von jedem iterierten Element senden, die Parametern ähneln: `varName: currentItem.fieldName`

    d. Das Ausgabefeld Mocked kann verwendet werden, um eine Scheinausgabe bereitzustellen, um die App während der Vorschau zu testen, wenn die Konnektoren nicht aktiv sind.

Konfigurieren Sie ein UI-Element, um die Automatisierung auszuführen

Da Sie nun über eine Automatisierung verfügen, die mit einer Aktion zum Aufrufen Ihrer Lambda-Funktion konfiguriert ist, können Sie ein UI-Element konfigurieren, um die Automatisierung auszuführen. In diesem Tutorial erstellen Sie eine Schaltfläche, die die Automatisierung ausführt, wenn Sie darauf klicken.

> **Tip**
>
> Mit der Aktion Automatisierung aufrufen können Sie auch Automatisierungen von anderen Automatisierungen aus ausführen.

Um Ihre Automatisierung über eine Schaltfläche auszuführen

1. Wählen Sie bei der Bearbeitung Ihrer App den Tab Seiten.
2. Wählen Sie im Menü auf der rechten Seite die Button-Komponente aus, um der Seite eine Schaltfläche hinzuzufügen.
3. Wählen Sie die neue Schaltfläche, um sie zu konfigurieren.
4. Wählen Sie im rechten Eigenschaftenmenü unter Trigger die Option + Hinzufügen und dann Automatisierung aufrufen aus.
5. Wählen Sie den neuen Auslöser für den Automatisierungsaufruf aus, um ihn zu konfigurieren.
6. Wählen Sie unter Automatisierung aufrufen die Automatisierung aus, die Ihre Lambda-Funktion aufruft, und konfigurieren Sie alle Parameter, die Sie an die Automatisierung senden möchten.

Jetzt veranlasst jeder Benutzer, der diese Schaltfläche in Ihrer App auswählt, die konfigurierte Automatisierung auszuführen.

Nächste Schritte: Zeigen Sie eine Vorschau der Anwendung an und veröffentlichen Sie sie zum Testen

Ihre Anwendung ist jetzt bereit zum Testen. Bei der Vorschau Ihrer App in der Entwicklungsumgebung sind Konnektoren nicht aktiv, sodass Sie die Automatisierung nicht während der Vorschau testen können, da sie einen Connector verwendet, um eine Verbindung herzustellen. AWS Lambda Um die Funktionalität Ihrer App zu testen, die von Konnektoren abhängt, müssen Sie die App in der Testumgebung veröffentlichen. Weitere Informationen zur Vorschau

und Veröffentlichung von Anwendungen finden Sie unterAnwendungen in der Vorschau anzeigen, veröffentlichen und teilen.

Erstellen Sie Ihre App Studio-App mit generativer KI

AWS App Studio bietet integrierte generative KI-Funktionen, um die Entwicklung zu beschleunigen und allgemeine Aufgaben zu rationalisieren. Sie können generative KI nutzen, um Apps, Datenmodelle und Beispieldaten zu generieren und zu bearbeiten und sogar kontextuelle Hilfe bei der Erstellung von Apps zu erhalten.

Generieren Sie Ihre App

Für einen schnelleren Start können Sie ganze Anwendungen mithilfe von KI-gestützten Eingabeaufforderungen in natürlicher Sprache generieren. Mit dieser Funktion können Sie Ihre gewünschte App-Funktionalität beschreiben, und KI erstellt automatisch die Datenmodelle, Benutzeroberflächen, Workflows und Konnektoren. Weitere Informationen zur Generierung einer App mit KI finden Sie unterErstellen einer Anwendung.

Ihre App erstellen oder bearbeiten

Während der Bearbeitung Ihrer Anwendung können Sie den Chat verwenden, um die Änderungen zu beschreiben, die Sie vornehmen möchten, und Ihre App wird automatisch aktualisiert. Sie können aus den vorhandenen Beispielaufforderungen wählen oder Ihre eigene Eingabeaufforderung eingeben. Der Chat kann verwendet werden, um unterstützte Komponenten hinzuzufügen, zu bearbeiten und zu entfernen sowie Automatisierungen und Aktionen zu erstellen und zu konfigurieren. Gehen Sie wie folgt vor, um Ihre Anwendung mithilfe von KI zu bearbeiten oder zu erstellen.

Um Ihre App mit KI zu bearbeiten

1. Bearbeiten Sie bei Bedarf Ihre App, um zum Anwendungsstudio zu navigieren.
2. (Optional) Wählen Sie die Seite oder Komponente aus, die Sie mit KI bearbeiten möchten.
3. Wählen Sie in der unteren linken Ecke die Option Mit KI erstellen aus, um den Chat zu öffnen.
4. Geben Sie die Änderungen ein, die Sie vornehmen möchten, oder wählen Sie aus den Beispielaufforderungen aus.
5. Überprüfen Sie die vorzunehmenden Änderungen. Wenn Sie möchten, dass die Änderungen vorgenommen werden, wählen Sie Bestätigen. Geben Sie andernfalls eine weitere Aufforderung ein.
6. Überprüfen Sie die Zusammenfassung der Änderungen.

Generieren Sie Ihre Datenmodelle

Sie können automatisch eine Entität mit Feldern, Datentypen und Datenaktionen auf der Grundlage des angegebenen Entitätsnamens generieren. Weitere Informationen zum Erstellen von Entitäten, einschließlich zum Erstellen von Entitäten mithilfe von GenAi, finden Sie unterEine Entität in einer App Studio-App erstellen.

Sie können eine bestehende Entität auch auf folgende Weise aktualisieren:

- Fügen Sie einer Entität weitere Felder hinzu. Weitere Informationen finden Sie unter Entitätsfelder hinzufügen, bearbeiten oder löschen.
- Fügen Sie Datenaktionen zu einer Entität hinzu. Weitere Informationen finden Sie unter Datenaktionen erstellen.

Generieren von Beispieldaten

Sie können Beispieldaten für Ihre Entitäten auf der Grundlage der Felder der Entität generieren. Dies ist nützlich, um Ihre Anwendung zu testen, bevor Sie externe Datenquellen verbinden, oder um Ihre Anwendung in der Entwicklungsumgebung zu testen, die nicht mit externen Datenquellen kommuniziert. Weitere Informationen finden Sie unter Beispieldaten hinzufügen oder löschen.

Sobald Sie Ihre App in Testing oder Production veröffentlicht haben, werden Ihre Live-Datenquellen und Konnektoren in diesen Umgebungen verwendet.

Aktionen für AWS Dienste konfigurieren

Bei der Integration mit AWS Diensten wie Amazon Simple Email Service können Sie KI verwenden, um eine Beispielkonfiguration mit vorausgefüllten Feldern basierend auf dem ausgewählten Service zu generieren. Um es auszuprobieren, erweitern Sie im Eigenschaftenmenü einer Automatisierungsaktion „AWS aufrufen“ das Feld Konfiguration, indem Sie den doppelseitigen Pfeil auswählen. Wählen Sie dann Beispielkonfiguration generieren aus.

Spöttische Antworten

Sie können gefälschte Antworten für AWS Serviceaktionen generieren. Dies ist hilfreich, wenn Sie Ihre Anwendung in der Entwicklungsumgebung testen möchten, die nicht mit externen Datenquellen kommuniziert.

KI beim Bauen um Hilfe bitten

Im Anwendungsstudio findest du unter den unterstützten Ressourcen oder Eigenschaften die Schaltfläche „KI um Hilfe bitten“. Verwenden Sie diese Option, um kontextbezogene Vorschläge, Dokumentation und Anleitungen zur aktuellen Ansicht oder ausgewählten Komponente zu erhalten. Stellen Sie allgemeine Fragen zu App Studio, zu bewährten Methoden für die App-Entwicklung oder zu Ihrem speziellen Anwendungsfall, um maßgeschneiderte Informationen und Empfehlungen zu erhalten.

Anwendungen erstellen, bearbeiten und löschen

Inhalt

- Erstellen einer Anwendung
- Anwendungen importieren
  - Von App Studio bereitgestellte importierbare Apps
- Anwendungen duplizieren
- Eine Anwendung bearbeiten oder erstellen
- Eine zuvor veröffentlichte App-Version bearbeiten
- Eine Anwendung umbenennen
- Löschen einer Anwendung

Erstellen einer Anwendung

Gehen Sie wie folgt vor, um eine Anwendung in App Studio zu erstellen.

So erstellen Sie eine Anwendung

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus, um zu einer Liste Ihrer Anwendungen zu navigieren.
2. Wählen Sie + App erstellen.
3. Geben Sie Ihrer Anwendung im Dialogfeld „App erstellen“ einen Namen und wählen Sie eine der folgenden Methoden zur App-Erstellung aus:
   - App mit KI generieren: Wählen Sie diese Option, um Ihre App in natürlicher Sprache zu beschreiben, und lassen Sie KI die App und ihre Ressourcen für Sie generieren.

- Von vorne beginnen: Wählen Sie diese Option, um mit der Entwicklung von einer leeren App aus zu beginnen.

4. Wählen Sie Weiter aus.
5. Wenn du „App mit KI generieren“ ausgewählt hast:

   a. Fügen Sie im Dialogfeld Mit vorhandenen Daten verbinden alle vorhandenen Datenquellen zu Ihrer App hinzu, indem Sie den Connector auswählen, der App Studio Zugriff auf die Datenquellen bietet, wählen Sie dann die Tabellen und dann Weiter aus. Wenn Sie hier Datenquellen hinzufügen, kann KI eine optimierte App für Sie erstellen. Sie können diesen Schritt überspringen und später Datenquellen hinzufügen, indem Sie Überspringen wählen.

   b. Nach einer kurzen Verzögerung (einige Minuten) werden Sie zur Seite Generieren Sie Ihre App mithilfe von KI weitergeleitet, auf der Sie die App beschreiben können, die Sie erstellen möchten.

   c. Sie können im Chat mit der Beschreibung Ihrer App beginnen oder eine bereitgestellte Beispielaufforderung auswählen und anpassen.

   d. Nachdem deine Aufforderung analysiert wurde, überprüfe die Anforderungen und den Überblick über die App. Verwenden Sie den Chat, um Änderungen anzufordern, oder wählen Sie Neu beginnen, um mit einer leeren Eingabeaufforderung zu beginnen.

   e. Wenn Sie bereit sind, wählen Sie App generieren.

   f. Sobald die App generiert ist, können Sie eine Vorschau auf einem anderen Tab anzeigen, indem Sie „App vorschauen“ wählen. Wenn Sie bereit sind, mit der Bearbeitung zu beginnen, können Sie App bearbeiten auswählen. Stöbern Sie durch die Seiten, Automatisierungen und Daten Ihrer Anwendung, um sich damit vertraut zu machen. Überprüfen Sie alle Fehler oder Warnungen im unteren Debug-Bereich. Weitere Informationen zum Generieren einer App mithilfe von KI finden Sie unterTutorial: Generieren Sie eine App mit KI. Allgemeine Informationen zur Funktionsweise der Erstellung in App Studio finden Sie unterSo funktioniert AWS App Studio.

6. Wenn Sie „Von vorne beginnen“ ausgewählt haben:

   a. Fügen Sie im Dialogfeld Mit vorhandenen Daten verbinden alle vorhandenen Datenquellen zu Ihrer App hinzu, indem Sie den Connector auswählen, der App Studio Zugriff auf die Datenquellen bietet, wählen Sie dann die Tabellen und dann Weiter aus. Sie können diesen Schritt überspringen und später Datenquellen hinzufügen, indem Sie Überspringen wählen.

   b. Sobald Ihre App erstellt ist, wählen Sie App bearbeiten, um mit der Bearbeitung Ihrer App zu beginnen. Weitere Informationen zum Erstellen aus einer leeren App finden Sie

unterTutorial: Beginne mit dem Erstellen von einer leeren App. Allgemeine Informationen darüber, wie das Erstellen in App Studio funktioniert, finden Sie unterSo funktioniert AWS App Studio.

Anwendungen importieren

Sie können eine Kopie einer exportierten Anwendung in Ihre App Studio-Instanz importieren. Sie können Apps importieren, die aus anderen App Studio-Instanzen exportiert wurden, oder Apps aus einem von App Studio bereitgestellten Katalog. Das Importieren einer App aus dem App Studio-App-Katalog kann Ihnen den Einstieg in eine App mit ähnlichen Funktionen erleichtern oder Ihnen helfen, mehr über die Erstellung von Apps in App Studio zu erfahren, indem Sie die importierte App erkunden.

Wenn Sie eine App in Ihre Instanz importieren, wird eine Kopie der ursprünglichen App in Ihrer Instanz erstellt. Nachdem die neue App erstellt wurde, werden Sie zur Entwicklungsumgebung der App weitergeleitet, wo Sie eine Vorschau der App anzeigen und die Funktionen der App durchsuchen können.

> **Warning**
>
> Wenn Sie eine App importieren, importieren Sie die gesamte Logik aus der Anwendung, was zu unerwünschtem oder unerwartetem Verhalten führen kann. Beispielsweise könnte es destruktive Abfragen geben, die Daten aus Datenbanken löschen, die Sie mit der Anwendung verbinden. Wir empfehlen, die Anwendung und ihre Konfiguration gründlich zu überprüfen und sie auf Geräten zu testen, die nicht für die Produktion bestimmt sind, bevor Sie Produktionsdaten mit der Anwendung verbinden.

Um eine Anwendung zu importieren

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus, um zu einer Liste Ihrer Anwendungen zu gelangen.
2. Wählen Sie den Dropdown-Pfeil neben + App erstellen aus.
3. Wählen Sie App importieren.
4. Geben Sie im Dialogfeld App importieren unter Importcode den Importcode der Anwendung ein, die Sie importieren möchten. Eine Liste der von App Studio bereitgestellten Apps, die importiert

werden können, einschließlich App-Beschreibungen und Importcodes, finden Sie unterVon App Studio bereitgestellte importierbare Apps.

5. Wählen Sie Import, um die App zu importieren, und wechseln Sie zur Entwicklungsumgebung der importierten App, um sie anzusehen oder zu bearbeiten. Informationen zum Erstellen von Apps in App Studio finden Sie unter So funktioniert AWS App Studio

Von App Studio bereitgestellte importierbare Apps

App Studio bietet Apps, die in Ihre Instanz importiert werden können, damit Sie mehr über die Erstellung von Apps erfahren. Um zu sehen, wie bestimmte App-Funktionen in App Studio implementiert sind, können Sie eine Vorschau der Anwendungen anzeigen und dann ihre Konfiguration in der Entwicklungsumgebung durchsuchen.

Die folgende Tabelle enthält die Liste der Anwendungen, ihre Importcodes und eine kurze Beschreibung der Apps. Jede App enthält eine `README` Seite mit Informationen über die App, die Sie nach dem Import anzeigen können.

Name der App	Beschreibung	Code importieren
Swag-Umfrage anfordern	Eine interne Swag-Request-Anwendung, mit der Mitarbeiter Markenartikel von Unternehmen bestellen können. Mitarbeiter können Artikel auswählen und Größen angeben und ihre Anfrage über ein einfaches Formular einreichen. Diese Anwendung verarbeitet alle Daten über den integrierten Speicher, sodass keine externen Verbindungen erforderlich sind.	Umfrage zu Werbeaktionen anfordern/EC4F5FAF-E2F8-42EE-AB8D-6723D8CA21B2
Sprintverfolgung	Eine Sprint-Management-Anwendung, mit der Teams die Softwareentwicklung	Sprinttracking/8f31e160-771f-48d7-87b0-374e285e2fbc

Name der App	Beschreibung	Code importieren
	organisieren und verfolgen können. Benutzer können Sprints erstellen, Aufgaben hinzufügen, Aufgaben zuweisen und den Fortschritt mithilfe spezieller Sprint-, Track- und Aufgabenansichten überwachen. Diese Anwendung verarbeitet alle Daten über den integrierten Speicher, sodass keine externen Verbindungen erforderlich sind.	
Amazon Review Sentiment Tracker	Diese Anwendung ist ein Tool zur Analyse von Kundenfeedback, das Stimmungswerte aus Produktbewertungen generiert, um Unternehmen dabei zu helfen, die Kundenzufriedenheit zu verstehen. Die Anwendung enthält Hilfsprogramme zur Generierung von Beispieldaten für schnelle Tests und benötigt einen Amazon Bedrock-Konnektor für KI-gestützte Erkenntnisse, während alle anderen Daten im integrierten Speichersystem beibehalten werden.	Amazon Bewertung Sentiment Tracker/60F0DAE4-F8E2-4C20-9583-FA456F5EBFAB

Name der App	Beschreibung	Code importieren
Rechnungs- und Quittungsverarbeitung	Diese Anwendung zur Belegverarbeitung spart Zeit und reduziert Fehler, indem sie die manuelle Dateneingabe automatisiert und die Workflows zur Genehmigung von Dokumenten rationalisiert. Die Lösung erfordert Amazon Textract-, Amazon S3- und Amazon SES SES-Konnektoren. Es verwendet Amazon Textract, um Daten aus in Amazon S3 gespeicherten Belegen zu analysieren und zu extrahieren. Anschließend werden die extrahierten Informationen mithilfe von Amazon SES verarbeitet und per E-Mail an Genehmiger gesendet.	Rechnungs- und Belegbearbeitung/98BDE3AE-E454-4B18-A1E6-6F23E8B2A4F1

Name der App	Beschreibung	Code importieren
Inspektion und Inventarprüfung	Eine Anwendung zur Verwaltung von Lagerinspektionen und zur Gerätever folgung. Benutzer können pass/fail Gerätebewertungen nach Raumstandort durchführen, den Lagerbestand überwachen und den Inspektionsverlauf einsehen. Die Anwendung bietet ein zentrales System zur Verfolgung sowohl der Anlageninspektionen als auch des Gerätestatus. Diese Anwendung verarbeitet alle Daten über den integrierten Speicher, sodass keine externen Verbindungen erforderlich sind.	Inspektion und Inventarprüfung/CF570A06-1C5E-4DD7-9EA8-5C04723D687F

Name der App	Beschreibung	Code importieren
Tracker zur Produktakzeptanz	Eine umfassende Anwendung zur Verwaltung der Produkten twicklung, die Kundenfee dback, Funktionsanfragen und Notizen zu Kundenges prächen zentralisiert. Teams können Kundeninteraktione n verfolgen, Anforderungen organisieren und KI-gestützte Berichte für die vierteljährliche Roadmap-Planung erstellen . Die Anwendung umfasst Hilfsprogramme für Beispield aten und nutzt Amazon Bedrock für KI-Einblicke und Amazon Aurora PostgreSQL für das Datenmanagement.	Product Adoption Tracker/9 B3A4437-BB50-467F-AE9E- D108776B7CA1
QuickSight Einbetten	Eine Demo-Anwendung, mit der Benutzer Analysen anzeigen können, während sie mit den zugrunde liegenden Daten arbeiten. Die App enthält zwei Methoden zum Einbetten von QuickSigh t Amazon-Dashboards in App Studio: einen API-basie rten Ansatz für registrie rte und anonyme Benutzer (QuickSight Connector erforderlich) und eine iFrame-Integration für öffentliche Dashboards.	Quicksight Embedding /0CDC15FC-CA8B-41B 7-869E-ED13C9072BC8

Name der App	Beschreibung	Code importieren
Spüle in der Küche	Eine Referenzanwendung mit Tipps und bewährten Methoden für die Entwicklung von App Studio für Fortgeschrittene. Enthält praktische Beispiele für Statusverwaltung, Umgang mit CSV-Daten, Browser-API-Integration und Benutzeroberflächenmuster, die Entwickler untersuchen und in ihren eigenen Anwendungen implementieren können. Keines der Beispiele erfordert externe Verbindungen.	App Studio Spülbecken/1CFE6B2F-544C-4611-B82C-80EADC76A0C8

Anwendungen duplizieren

Anwendungsinhaber und Miteigentümer können ihre Apps duplizieren, um eine exakte Kopie der App zu erstellen. Das Duplizieren von Apps ist hilfreich, wenn Sie den aktuellen Status zu Testzwecken beibehalten oder die duplizierte App als Ausgangspunkt für die Erstellung einer neuen App verwenden möchten.

Um eine Anwendung zu duplizieren

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der Anwendungen angezeigt wird, auf die Sie Zugriff haben.
2. Wählen Sie das Drop-down-Menü in der Spalte Aktionen der Anwendung aus, die Sie duplizieren möchten.
3. Wählen Sie Duplicate (Duplikat) aus. Wenn die Option Duplizieren nicht verfügbar ist, sind Sie wahrscheinlich kein Eigentümer oder Miteigentümer der Anwendung.
4. Geben Sie optional einen Namen der duplizierten App an. Der Standardname lautet *Current_App_Name COPY*.

5. Wählen Sie Duplicate (Duplikat) aus.

Eine Anwendung bearbeiten oder erstellen

Gehen Sie wie folgt vor, um eine Anwendung in App Studio zu bearbeiten.

Um eine Anwendung zu bearbeiten (zu erstellen)

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der Anwendungen angezeigt wird, auf die Sie Zugriff haben.
2. Wählen Sie in der Spalte Aktionen der Anwendung, die Sie bearbeiten möchten, die Option Bearbeiten aus. Dadurch gelangen Sie zur Entwicklungsumgebung, in der Sie Komponenten, Automatisierungen und Daten verwenden können, um das Aussehen und die Funktion Ihrer Anwendung zu konfigurieren. Informationen zum Erstellen von Anwendungen finden Sie unterErste Schritte mit AWS App Studio.

Eine zuvor veröffentlichte App-Version bearbeiten

Gehen Sie wie folgt vor, um eine zuvor veröffentlichte Version Ihrer App Studio-Anwendung zu bearbeiten. Nachdem Sie sich entschieden haben, die zuvor veröffentlichte Version zu bearbeiten, können Sie die App in der Entwicklungsumgebung bearbeiten oder sie in Testing und dann in Production veröffentlichen.

> **Warning**
>
> Die zuvor veröffentlichte Version ersetzt die in Bearbeitung befindliche Version der App in der Entwicklungsumgebung. Alle unveröffentlichten Änderungen an Ihrer App gehen verloren.

Das Bearbeiten einer zuvor veröffentlichten Version ist nützlich, falls Sie versehentlich unerwünschte Änderungen oder Änderungen veröffentlichen, die die Anwendung für Ihre Benutzer beeinträchtigen, und Sie die vorherige App-Version weiter erstellen oder bearbeiten möchten.

> **Note**
>
> Wenn Sie Probleme mit einer veröffentlichten App feststellen und sofort eine zuvor funktionierende Version veröffentlichen müssen, oder wenn Sie eine frühere Version

veröffentlichen möchten, aber die neuesten Updates der App in der Entwicklungsumgebung beibehalten möchten, sollten Sie stattdessen ein Rollback der App durchführen. Weitere Informationen finden Sie unter Rollback zu einer zuvor veröffentlichten Version.

Um eine zuvor veröffentlichte App-Version zu bearbeiten

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie den Dropdown-Pfeil neben der Schaltfläche „Veröffentlichen“ und dann „Veröffentlichungszentrum“.
3. Wählen Sie Versionsverlauf, um eine Liste der zuvor veröffentlichten Versionen der Anwendung anzuzeigen.
4. Suchen Sie die Version, die Sie bearbeiten möchten, und wählen Sie Bearbeiten.
5. Überprüfen Sie die Informationen und wählen Sie „Wiederherstellen“.
6. Die Version, die Sie zur Bearbeitung ausgewählt haben, ist jetzt die aktuelle Version in der Entwicklungsumgebung. Sie können Änderungen daran vornehmen oder sie unverändert in der Testumgebung veröffentlichen, indem Sie Veröffentlichen wählen. Nach der Veröffentlichung in Testing können Sie bei Bedarf erneut in der Produktionsumgebung veröffentlichen.

Eine Anwendung umbenennen

Gehen Sie wie folgt vor, um eine Anwendung in App Studio umzubenennen. Sie können eine Anwendung in der Liste der Anwendungen oder in der Entwicklungsumgebung beim Erstellen der App umbenennen.

Um eine Anwendung umzubenennen

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der Anwendungen angezeigt wird, auf die Sie Zugriff haben.
2. Sie können eine Anwendung aus dieser Liste oder während der Bearbeitung in der Entwicklungsumgebung umbenennen.
   - Um aus dieser Liste umzubenennen:
     a. Wählen Sie das Drop-down-Menü in der Spalte Aktionen der Anwendung, die Sie umbenennen möchten, und wählen Sie dann Umbenennen aus.

b. Geben Sie Ihrer Anwendung einen neuen Namen und wählen Sie „Umbenennen“.

- Um in der Entwicklungsumgebung umzubenennen:

a. Wählen Sie in der Spalte Aktionen der Anwendung, die Sie bearbeiten möchten, die Option Bearbeiten aus.

b. Wählen Sie in der Entwicklungsumgebung den Namen der Anwendung aus und aktualisieren Sie ihn. Drücken Sie dann die Eingabetaste oder gehen Sie vom Textfeld weg, um Ihre Änderungen zu speichern.

Löschen einer Anwendung

Gehen Sie wie folgt vor, um eine Anwendung in App Studio zu löschen.

So löschen Sie eine Anwendung

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der Anwendungen angezeigt wird, auf die Sie Zugriff haben.
2. Wählen Sie das Drop-down-Menü in der Spalte Aktionen der Anwendung aus, die Sie löschen möchten.
3. Wählen Sie Löschen aus.
4. Lesen Sie sich im Dialogfeld „Anwendung löschen“ die Informationen zum Löschen von Anwendungen sorgfältig durch. Wenn Sie die Anwendung löschen möchten, wählen Sie Löschen.

Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen

Themen

- Eine Vorschau von Anwendungen anzeigen
- Anwendungen veröffentlichen
- Gemeinsame Nutzung veröffentlichter Anwendungen
- Rollback zu einer zuvor veröffentlichten Version
- Anwendungen exportieren

Eine Vorschau von Anwendungen anzeigen

Sie können eine Vorschau von Anwendungen in App Studio anzeigen, um zu sehen, wie sie Benutzern angezeigt werden. Außerdem können Sie ihre Funktionalität testen, indem Sie sie verwenden und die Protokolle in einem Debug-Bereich überprüfen.

Die Anwendungsvorschauumgebung unterstützt nicht die Anzeige von Live-Daten oder die Verbindung mit externen Ressourcen über Konnektoren, wie z. B. Datenquellen. Um die Funktionalität in der Vorschauumgebung zu testen, können Sie simulierte Ausgaben in Automatisierungen und Beispieldaten in Entitäten verwenden. Um Ihre Anwendung mit Echtzeitdaten anzeigen zu können, müssen Sie Ihre App veröffentlichen. Weitere Informationen finden Sie unter Anwendungen veröffentlichen.

In der Vorschau- oder Entwicklungsumgebung wird die in den anderen Umgebungen veröffentlichte Anwendung nicht aktualisiert. Wenn eine Anwendung nicht veröffentlicht wurde, können Benutzer erst darauf zugreifen, wenn sie veröffentlicht und gemeinsam genutzt wurde. Wenn eine Anwendung bereits veröffentlicht und gemeinsam genutzt wurde, greifen Benutzer weiterhin auf die veröffentlichte Version zu und nicht auf die Version, die in einer Vorschauumgebung verwendet wurde.

Um eine Vorschau Ihrer Anwendung anzuzeigen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio der Anwendung, für die Sie eine Vorschau anzeigen möchten:
   a. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus.
   b. Wählen Sie Bearbeiten für die Anwendung aus.
2. Wählen Sie Vorschau, um die Vorschauumgebung für die Anwendung zu öffnen.
3. (Optional) Erweitern Sie das Debug-Bedienfeld, indem Sie die entsprechende Kopfzeile am unteren Bildschirmrand auswählen. Sie können das Fenster nach Nachrichtentyp filtern, indem Sie den Nachrichtentyp im Abschnitt Protokolle filtern auswählen. Sie können die Protokolle des Panels löschen, indem Sie Konsole löschen wählen.
4. In der Vorschauumgebung können Sie Ihre Anwendung testen, indem Sie auf ihren Seiten navigieren, ihre Komponenten verwenden und ihre Schaltflächen auswählen, um Automatisierungen zu starten, die Daten übertragen. Da die Vorschauumgebung keine Live-Daten oder Verbindungen zu externen Quellen unterstützt, können Sie sich Beispiele für die übertragenen Daten im Debug-Bereich ansehen.

Anwendungen veröffentlichen

Wenn Sie mit der Erstellung und Konfiguration Ihrer Anwendung fertig sind, besteht der nächste Schritt darin, sie zu veröffentlichen, um Datenübertragungen zu testen oder sie für Endbenutzer freizugeben. Um das Veröffentlichen von Anwendungen in App Studio zu verstehen, ist es wichtig, die verfügbaren Umgebungen zu verstehen. App Studio bietet drei separate Umgebungen, die in der folgenden Liste beschrieben werden:

1. Entwicklung: Hier erstellen Sie Ihre Anwendung und zeigen eine Vorschau an. Sie müssen nicht in der Entwicklungsumgebung veröffentlichen, da die neueste Version Ihrer Anwendung dort automatisch gehostet wird. In dieser Umgebung sind keine Live-Daten oder Dienste oder Ressourcen von Drittanbietern verfügbar.
2. Testen: Hier können Sie umfassende Tests Ihrer Anwendung durchführen. In der Testumgebung können Sie eine Verbindung zu anderen Diensten herstellen, Daten an diese senden und Daten von anderen Diensten empfangen.
3. Produktion: Die Live-Betriebsumgebung für die Nutzung durch Endbenutzer.

Ihre gesamte App-Erstellung findet in der Entwicklungsumgebung statt. Veröffentlichen Sie anschließend in der Testumgebung, um die Datenübertragung zwischen anderen Diensten zu testen, sowie Benutzerakzeptanztests (UAT), indem Sie Endbenutzern eine Zugriffs-URL zur Verfügung stellen. Veröffentlichen Sie Ihre App anschließend in der Produktionsumgebung, um abschließende Tests durchzuführen, bevor Sie sie mit Benutzern teilen. Weitere Informationen zu den Anwendungsumgebungen finden Sie unterAnwendungsumgebungen.

Wenn Sie eine Anwendung veröffentlichen, ist sie für Benutzer erst verfügbar, wenn sie gemeinsam genutzt wird. Auf diese Weise haben Sie die Möglichkeit, die Anwendung in der Test- und Produktionsumgebung zu verwenden und zu testen, bevor Benutzer darauf zugreifen können. Wenn Sie eine Anwendung in der Produktionsumgebung veröffentlichen, die zuvor veröffentlicht und gemeinsam genutzt wurde, wird die Version aktualisiert, die Benutzern zur Verfügung steht.

Anwendungen veröffentlichen

Gehen Sie wie folgt vor, um eine App Studio-Anwendung in der Test- oder Produktionsumgebung zu veröffentlichen.

Um eine Anwendung in der Test- oder Produktionsumgebung zu veröffentlichen

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Sie werden zu einer Seite weitergeleitet, auf der eine Liste der Anwendungen angezeigt wird, auf die Sie Zugriff haben.
2. Wählen Sie Bearbeiten für die Anwendung, die Sie veröffentlichen möchten.
3. Wählen Sie in der oberen rechten Ecke Veröffentlichen aus.
4. Gehen Sie im Dialogfeld „Updates veröffentlichen“ wie folgt vor:

    a. Lesen Sie die Informationen zur Veröffentlichung einer Anwendung.

    b. (Optional) Fügen Sie unter Versionsbeschreibung eine Beschreibung dieser Version der Anwendung hinzu.

    c. Wählen Sie das Feld aus, um die Informationen über die Umgebung zu bestätigen.

    d. Wählen Sie Starten. Es kann bis zu 15 Minuten dauern, bis die Anwendung in der Live-Umgebung aktualisiert ist.
5. Informationen zum Anzeigen von Anwendungen in der Test- oder Produktionsumgebung finden Sie unterVeröffentlichte Anwendungen anzeigen.

> **Note**
>
> Die Verwendung der Anwendung in der Test- oder Produktionsumgebung führt zu Live-Datenübertragungen, z. B. zur Erstellung von Datensätzen in Tabellen mit Datenquellen, die über Konnektoren verbunden wurden.

Veröffentlichte Anwendungen, die noch nie gemeinsam genutzt wurden, stehen Benutzern oder anderen Entwicklern nicht zur Verfügung. Um eine Anwendung Benutzern zur Verfügung zu stellen, müssen Sie sie nach der Veröffentlichung teilen. Weitere Informationen finden Sie unter Gemeinsame Nutzung veröffentlichter Anwendungen.

Veröffentlichte Anwendungen anzeigen

Sie können sich die in den Test- und Produktionsumgebungen veröffentlichten Anwendungen ansehen, um die Anwendung zu testen, bevor Sie sie für Endbenutzer oder andere Entwickler freigeben.

Um veröffentlichte Anwendungen in der Test- oder Produktionsumgebung anzuzeigen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio der Anwendung, für die Sie eine Vorschau anzeigen möchten:
   a. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus.
   b. Wählen Sie Bearbeiten für die Anwendung aus.
2. Wählen Sie den Dropdown-Pfeil neben Veröffentlichen in der oberen rechten Ecke und wählen Sie Publish Center aus.
3. Im Publishing Center können Sie die Umgebungen anzeigen, in denen Ihre Anwendung veröffentlicht wurde. Wenn Ihre Anwendung in der Test- oder Produktionsumgebung veröffentlicht wurde, können Sie die App über den URL-Link für jede Umgebung anzeigen.

> **Note**
>
> Wenn Sie die Anwendung in der Test- oder Produktionsumgebung verwenden, erfolgt eine Live-Datenübertragung, z. B. das Erstellen von Datensätzen in Tabellen mit Datenquellen, die mit Konnektoren verbunden wurden.

Anwendungsumgebungen

AWS App Studio bietet Funktionen für das Application Lifecycle Management (ALM) in drei separaten Umgebungen: Entwicklung, Testen und Produktion. Auf diese Weise können Sie Best Practices wie die Verwaltung separater Umgebungen, Versionskontrolle, gemeinsame Nutzung und Überwachung über den gesamten App-Lebenszyklus hinweg einfacher anwenden.

Entwicklungsumgebung

Die Entwicklungsumgebung ist eine isolierte Sandbox, in der Sie mithilfe des Anwendungsstudios und Beispieldaten Apps erstellen können, ohne eine Verbindung zu Live-Datenquellen oder -diensten herstellen zu müssen. In der Entwicklungsumgebung können Sie eine Vorschau Ihrer App anzeigen, um sie anzusehen und zu testen, ohne die Produktionsdaten zu gefährden.

Ihre App stellt zwar keine Verbindung zu anderen Diensten in der Entwicklungsumgebung her, Sie können jedoch verschiedene Ressourcen in Ihrer App konfigurieren, um Live-Datenkonnektoren und Automatisierungen nachzuahmen.

Am unteren Rand des Anwendungsstudios in der Entwicklungsumgebung befindet sich ein zusammenklappbares Debug-Bedienfeld, das Fehler und Warnungen enthält, damit Sie die App während der Erstellung überprüfen und debuggen können. Weitere Informationen zur Problembehandlung und zum Debuggen von Apps finden Sie unter. Problembehandlung und Debuggen von App Studio

Testumgebung

Sobald Ihre erste App-Entwicklung abgeschlossen ist, besteht der nächste Schritt darin, sie in der Testumgebung zu veröffentlichen. In der Testumgebung kann Ihre App eine Verbindung zu anderen Diensten herstellen, Daten an diese senden und Daten von anderen Diensten empfangen. Daher können Sie diese Umgebung verwenden, um umfassende Tests durchzuführen, einschließlich Benutzerakzeptanztests (UAT), indem Sie Endbenutzern eine Zugriffs-URL zur Verfügung stellen.

> **Note**
>
> Ihre erste Veröffentlichung in der Testumgebung kann bis zu 15 Minuten dauern.

Die in der Testumgebung veröffentlichte Version Ihrer App wird nach 3 Stunden Inaktivität des Endbenutzers entfernt. Alle Versionen bleiben jedoch bestehen und können über den Tab Versionsverlauf wiederhergestellt werden.

Die wichtigsten Funktionen der Testumgebung sind wie folgt:

- Integrationstests mit Live-Datenquellen und APIs
- Benutzerakzeptanztests (UAT) werden durch kontrollierten Zugriff erleichtert
- Umgebung zum Sammeln von Feedback und zur Behebung von Problemen
- Fähigkeit, sowohl clientseitige als auch serverseitige Aktivitäten mithilfe von Browserkonsolen und Entwicklertools zu überprüfen und zu debuggen.

Weitere Informationen zur Problembehandlung und zum Debuggen von Apps finden Sie unter. Problembehandlung und Debuggen von App Studio

Produktionsumgebung

Nachdem Sie alle Probleme getestet und behoben haben, können Sie die Version Ihrer Anwendung von der Testumgebung in die Produktionsumgebung hochstufen, sodass Sie sie live nutzen können.

Die Produktionsumgebung ist zwar die Live-Betriebsumgebung für die Nutzung durch Endbenutzer, Sie können die veröffentlichte Version jedoch testen, bevor Sie sie mit Benutzern teilen.

Ihre veröffentlichte Version in der Produktionsumgebung wird nach 14 Tagen Inaktivität durch den Endbenutzer entfernt. Alle Versionen bleiben jedoch bestehen und können über die Registerkarte Versionsverlauf wiederhergestellt werden.

Die Hauptmerkmale der Produktionsumgebung sind wie folgt:

- Live-Betriebsumgebung für die Nutzung durch den Endbenutzer
- Granulare, rollenbasierte Zugriffskontrolle
- Funktionen für Versionskontrolle und Rollback
- Möglichkeit, nur clientseitige Aktivitäten zu überprüfen und zu debuggen
- Verwendet Live-Konnektoren, Daten, Automatisierungen und APIs

Versionierung und Release-Management

App Studio bietet Funktionen für Versionskontrolle und Release-Management über sein Versionssystem im Publish Center.

Die wichtigsten Funktionen zur Versionierung:

- Beim Veröffentlichen in der Testumgebung werden neue Versionsnummern generiert (1.0, 2.0, 3.0...).
- Die Versionsnummer ändert sich nicht, wenn Sie von der Test- zur Produktionsumgebung heraufstufen.
- Sie können aus dem Versionsverlauf zu jeder vorherigen Version zurückkehren.
- In der Testumgebung veröffentlichte Anwendungen werden nach 3 Stunden Inaktivität angehalten. Versionen werden dauerhaft gespeichert und können aus dem Versionsverlauf wiederhergestellt werden.
- In der Produktionsumgebung veröffentlichte Anwendungen werden nach 14 Tagen Inaktivität entfernt. Versionen werden dauerhaft gespeichert und können aus dem Versionsverlauf wiederhergestellt werden.

Dieses Versionsmodell ermöglicht eine schnelle Iteration bei gleichzeitiger Wahrung der Rückverfolgbarkeit, der Rollback-Funktionen und der optimalen Leistung während des gesamten Entwicklungs- und Testzyklus der App.

Wartung und Betrieb

App Studio muss Ihre Anwendung möglicherweise automatisch erneut veröffentlichen, um bestimmte Wartungsaufgaben und betriebliche Aktivitäten zu erledigen und neue Softwarebibliotheken zu integrieren. Sie, der Builder, müssen nichts unternehmen, aber Endbenutzer müssen sich möglicherweise wieder bei der Anwendung anmelden. In bestimmten Situationen müssen Sie Ihre Anwendung möglicherweise erneut veröffentlichen, um neue Funktionen und Bibliotheken zu integrieren, die wir nicht automatisch hinzufügen können. Vor der erneuten Veröffentlichung müssen Sie alle Fehler beheben und die Warnungen überprüfen.

Gemeinsame Nutzung veröffentlichter Anwendungen

Wenn Sie eine Anwendung veröffentlichen, die noch nicht veröffentlicht wurde, ist sie für Benutzer erst verfügbar, wenn sie gemeinsam genutzt wird. Sobald eine veröffentlichte Anwendung gemeinsam genutzt wurde, steht sie den Benutzern zur Verfügung und muss nicht erneut freigegeben werden, wenn eine andere Version veröffentlicht wird.

> **Note**
>
> In diesem Abschnitt geht es darum, veröffentlichte Anwendungen mit Endbenutzern oder Testern zu teilen. Informationen dazu, wie Sie andere Benutzer einladen, eine App zu erstellen, finden Sie unterEine App mit mehreren Benutzern erstellen.

So teilen Sie eine veröffentlichte Anwendung

1. Rufen Sie das Dialogfeld „Teilen“ entweder über die Anwendungsliste oder das Anwendungsstudio Ihrer App auf, indem Sie die folgenden Anweisungen befolgen:
   - So greifen Sie über die Anwendungsliste auf das Dialogfeld „Teilen“ zu: Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus. Wählen Sie das Dropdownmenü in der Spalte Aktionen der Anwendung aus, die Sie teilen möchten, und wählen Sie Teilen aus.
   - So greifen Sie vom Anwendungsstudio aus auf das Dialogfeld „Teilen“ zu: Wählen Sie im Anwendungsstudio Ihrer App in der oberen Kopfzeile die Option Teilen aus.
2. Wählen Sie im Dialogfeld „Teilen“ die Registerkarte für die Umgebung aus, die Sie teilen möchten. Wenn die Tabs „Testen“ oder „Produktion“ nicht angezeigt werden, wurde Ihre App möglicherweise nicht in der entsprechenden Umgebung veröffentlicht. Weitere Informationen zum Veröffentlichen finden Sie unter Anwendungen veröffentlichen.

3. Wählen Sie auf der entsprechenden Registerkarte Gruppen aus dem Drop-down-Menü aus, um die Umgebung mit ihnen zu teilen.
4. (Optional) Weisen Sie der Gruppe eine Rolle auf App-Ebene zu, um die bedingte Seitensichtbarkeit zu testen oder zu konfigurieren. Weitere Informationen finden Sie unter Konfiguration der rollenbasierten Sichtbarkeit von Seiten.
5. Wählen Sie Freigeben.
6. (Optional) Kopieren Sie den Link und teilen Sie ihn mit Benutzern. Nur Benutzer, mit denen die Anwendung und die Umgebung gemeinsam genutzt wurden, können auf die Anwendung in der entsprechenden Umgebung zugreifen.

Rollback zu einer zuvor veröffentlichten Version

Gehen Sie wie folgt vor, um die Produktionsumgebung Ihrer App Studio-App auf eine zuvor veröffentlichte Version zurückzusetzen. Die Endbenutzer Ihrer Anwendung werden davon betroffen sein und die Rollback-Version Ihrer App sehen, nachdem sie bereitgestellt wurde. Wenn Sie eine Anwendung rückgängig machen, wird auch der Komponentencode auf die Version von der vorherigen Veröffentlichung zurückgesetzt. Dies wirkt sich auf den gesamten Stapel der Anwendungsbereitstellung aus (Benutzercode, Status der Komponentenkonfiguration). Das bedeutet, dass alle Aktualisierungen, die App Studio am Komponentencode vorgenommen hat, wie z. B. Feld- oder andere Konfigurationsänderungen, zurückgesetzt werden, um sicherzustellen, dass die zurückgesetzte Anwendungsversion genauso funktioniert wie bei der ursprünglichen Veröffentlichung.

Die in Bearbeitung befindliche Version Ihrer Anwendung in der Entwicklungsumgebung ist nicht betroffen, wenn Sie die veröffentlichte Version zurücksetzen.

Ein Rollback der veröffentlichten Version einer Anwendung ist hilfreich, wenn Sie Probleme mit einer veröffentlichten App feststellen und sofort eine zuvor funktionierende Version veröffentlichen müssen, oder wenn Sie eine frühere Version veröffentlichen und die neuesten Updates der App in der Entwicklungsumgebung beibehalten möchten.

> **Note**
>
> Wenn Sie die Entwicklungsumgebung einer App auf eine zuvor veröffentlichte Version zurücksetzen möchten, sollten Sie die Anwendung zurücksetzen. Weitere Informationen finden Sie unter Eine zuvor veröffentlichte App-Version bearbeiten.

Um die Version der Produktionsumgebung auf eine zuvor veröffentlichte App-Version zurückzusetzen

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten. Weitere Informationen finden Sie unter Eine Anwendung bearbeiten oder erstellen.
2. Wählen Sie den Dropdown-Pfeil für die Version oben auf der Kachel der Produktionsumgebung, um die verfügbaren Versionen für ein Rollback anzuzeigen. Die Dropdownliste enthält Versionen, die in den letzten 30 Tagen veröffentlicht wurden. Wenn dieses Dropdown deaktiviert ist, kann das daran liegen, dass eine App-Veröffentlichung bereits im Gange ist und nur eine Veröffentlichung gleichzeitig erfolgen kann.
3. Wählen Sie die Version aus, zu der Sie zurückkehren möchten.
4. Geben Sie einen Grund für das Rollback ein und wählen Sie Rollback. Die Rollback-Veröffentlichung wird gestartet. Sobald der Vorgang abgeschlossen ist, wird die Produktionsumgebung Ihrer Anwendung auf die gewählte Version aktualisiert.

> **Note**
>
> Sie können nach dem Rollback auch ein Rollforward zu einer zuvor veröffentlichten App-Version durchführen.

Anwendungen exportieren

Sie können einen Snapshot Ihrer Anwendung exportieren, um ihn mit anderen App Studio-Instanzen zu teilen. Wenn Sie eine App exportieren, wird ein Snapshot aus der Entwicklungsumgebung der App erstellt und ein Importcode generiert. Der Importcode kann dann verwendet werden, um die Anwendung in andere App Studio-Instanzen zu importieren, wo sie angezeigt und erstellt werden kann.

Exportierte Apps können in alle von App Studio AWS-Region unterstützten Instanzen importiert werden.

Um eine Anwendung zu exportieren

1. Wählen Sie im Navigationsbereich im Abschnitt Build die Option Meine Anwendungen aus, um zu einer Liste Ihrer Anwendungen zu gelangen.
2. Wählen Sie das Drop-down-Menü in der Spalte Aktionen der Anwendung aus, die Sie exportieren möchten.

3. Wählen Sie Export aus.
4. Das Verfahren zum Generieren und Teilen eines Importcodes hängt davon ab, ob bereits ein Importcode für die App erstellt wurde oder nicht.
   - Wenn kein Importcode erstellt wurde:
     a. Geben Sie unter Importberechtigungen für Anwendungen an, welche Instanzen die exportierte App importieren können. Sie können allen Instanzen Importberechtigungen erteilen oder bestimmte App Studio-Instanzen hinzufügen, indem Sie deren Instanz eingeben IDs. Trennen Sie mehrere Instanzen IDs durch ein Komma.

        Um Ihre Instanz-ID zu finden, navigieren Sie zu den Kontoeinstellungen Ihrer Instanz, indem Sie in der App Studio-Konsole Kontoeinstellungen auswählen.
     b. Wählen Sie Importcode generieren aus.
     c. Kopieren Sie den generierten Importcode und teilen Sie ihn.
   - Wenn bereits ein Importcode erstellt wurde:
     - Um die aktuell exportierte App zu teilen, kopieren Sie den vorhandenen Importcode und teilen Sie ihn mit anderen. Um eine neue exportierte App mit den neuesten Änderungen an Ihrer App zu erstellen, wählen Sie Neuen Code generieren. Sie können die Importberechtigungen bei Bedarf auch aktualisieren.

Seiten und Komponenten: Erstellen Sie die Benutzeroberfläche Ihrer App

Themen

- Seiten verwalten
- Verwaltung von Komponenten
- Konfiguration der rollenbasierten Sichtbarkeit von Seiten
- Seiten in der App-Navigation ordnen und organisieren
- Ändern Sie die Farben in Ihrer App mit App-Themen
- Referenz zu den Komponenten

Seiten verwalten

Gehen Sie wie folgt vor, um Seiten aus Ihrer AWS App Studio-Anwendung zu erstellen, zu bearbeiten oder zu löschen.

Seiten sind Container für Komponenten, die die Benutzeroberfläche einer Anwendung in App Studio bilden. Jede Seite stellt einen Bildschirm der Benutzeroberfläche (UI) Ihrer Anwendung dar, mit dem Ihre Benutzer interagieren werden. Seiten werden auf der Registerkarte Seiten des Anwendungsstudios erstellt und bearbeitet.

Eine Seite erstellen

Gehen Sie wie folgt vor, um eine Seite in einer Anwendung in App Studio zu erstellen. Informationen zum Duplizieren einer vorhandenen Seite finden Sie unterEine Seite duplizieren.

Um eine Seite zu erstellen

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Wähle im Seitenmenü auf der linken Seite die Option + Hinzufügen.

Eine Seite duplizieren

Gehen Sie wie folgt vor, um eine Seite in einer Anwendung in App Studio zu duplizieren.

Um eine Seite zu duplizieren

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Wählen Sie im Seitenmenü auf der linken Seite das Ellipsenmenü neben dem Namen der Seite, die Sie duplizieren möchten, und wählen Sie Duplizieren aus. Die duplizierte Seite wird direkt nach der Originalseite hinzugefügt.

Seiteneigenschaften anzeigen und bearbeiten

Gehen Sie wie folgt vor, um eine Seite in einer Anwendung in App Studio zu bearbeiten. Sie können Eigenschaften wie den Namen der Seite, ihre Parameter und ihr Layout bearbeiten.

Um Seiteneigenschaften anzuzeigen oder zu bearbeiten

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Wählen Sie im Seitenmenü auf der linken Seite das Ellipsenmenü neben dem Namen der Seite, die Sie bearbeiten möchten, und wählen Sie Seiteneigenschaften aus. Dadurch wird das Eigenschaftenmenü auf der rechten Seite geöffnet.
4. Um den Seitennamen zu bearbeiten:

   > **Note**
   >
   > Gültige Zeichen für Seitennamen: A-Z, a-z, 0-9, _, $

   a. Wählen Sie das Stiftsymbol neben dem Namen oben im Eigenschaften-Menü.
   b. Geben Sie den neuen Namen für Ihre Seite ein und drücken Sie die Eingabetaste.
5. Um Seitenparameter zu erstellen, zu bearbeiten oder zu löschen:

   a. Um einen Seitenparameter zu erstellen, wählen Sie + Neu hinzufügen im Abschnitt Seitenparameter.
   b. Um den Schlüssel - oder Beschreibungswert eines Seitenparameters zu bearbeiten, wählen Sie das Eingabefeld der Eigenschaft, die Sie ändern möchten, und geben Sie einen neuen Wert ein. Ihre Änderungen werden gespeichert, während Sie sie bearbeiten.
   c. Um einen Seitenparameter zu löschen, wählen Sie das Papierkorbsymbol des Seitenparameters, den Sie löschen möchten.
6. Um das Logo oder das Banner einer Seite hinzuzufügen, zu bearbeiten oder zu entfernen:

   a. Um ein Seitenlogo oder ein Banner hinzuzufügen, aktivieren Sie die entsprechende Option im Bereich Stil. Konfigurieren Sie die Bildquelle und geben Sie optional Alternativtext an.
   b. Um ein Seitenlogo oder ein Banner zu bearbeiten, aktualisieren Sie die Felder im Bereich Stil.

c. Um ein Seitenlogo oder ein Banner zu entfernen, deaktivieren Sie die entsprechende Option im Bereich Stil.

7. Um das Layout einer Seite zu bearbeiten:

   • Aktualisieren Sie die Felder im Abschnitt Layout.

Eine Seite löschen

Gehen Sie wie folgt vor, um eine Seite aus einer Anwendung in App Studio zu löschen.

Um eine Seite zu löschen

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Wählen Sie im Seitenmenü auf der linken Seite das Ellipsenmenü neben dem Namen der Seite, die Sie löschen möchten, und wählen Sie Löschen aus.

Verwaltung von Komponenten

Gehen Sie wie folgt vor, um Komponenten auf oder von Seiten im App Studio Application Studio hinzuzufügen, zu bearbeiten und zu löschen, um die gewünschte Benutzeroberfläche für Ihre Anwendung zu erstellen.

Komponenten zu einer Seite hinzufügen

Gehen Sie wie folgt vor, um einer Seite in App Studio eine Komponente hinzuzufügen. Informationen zum Duplizieren einer vorhandenen Komponente finden Sie unterKomponenten duplizieren.

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Das Komponentenfenster befindet sich im Menü auf der rechten Seite, das die verfügbaren Komponenten enthält.
4. Ziehen Sie die gewünschte Komponente per Drag & Drop aus dem Bedienfeld auf die Leinwand. Alternativ können Sie im Bedienfeld auf die Komponente doppelklicken, um sie automatisch in die Mitte der aktuellen Seite einzufügen.

5. Nachdem Sie eine Komponente hinzugefügt haben, können Sie im Eigenschaftenfenster auf der rechten Seite ihre Einstellungen anpassen, z. B. die Datenquelle, das Layout und das Verhalten. Ausführliche Informationen zur Konfiguration der einzelnen Komponententypen finden Sie unterReferenz zu den Komponenten.

Komponenten duplizieren

Gehen Sie wie folgt vor, um eine Komponente in einer App Studio-App zu duplizieren. Duplizierte Komponenten enthalten alle verknüpften Automatisierungen oder Entitäten aus der Originalkomponente.

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Es gibt zwei Möglichkeiten, eine Komponente zu duplizieren:
   - Erweitern Sie im Seitenmenü auf der linken Seite die Seite, die die Komponente enthält, die Sie duplizieren möchten. Wählen Sie das Ellipsenmenü neben dem Namen der Komponente, die Sie duplizieren möchten, und wählen Sie Duplizieren aus.
   - Wählen Sie die Komponente aus, die Sie duplizieren möchten, und klicken Sie auf das Symbol „Duplizieren“.

   Die duplizierte Komponente wird direkt nach der Originalkomponente hinzugefügt.

> **Tip**
>
> Sie können eine Komponentenduplizierung zusammen mit vielen anderen Aktionen in der Entwicklungsumgebung rückgängig machen, indem Sie die Tastenkombinationen STRG+Z oder CMD+Z verwenden.

Komponenteneigenschaften anzeigen und bearbeiten

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.

3. Erweitern Sie im Seitenmenü auf der linken Seite die Seite, die die Komponente enthält, und wählen Sie die Komponente aus, die angezeigt oder bearbeitet werden soll. Alternativ können Sie die Seite und dann die Komponente auf der Arbeitsfläche auswählen.
4. Im Eigenschaftenbereich auf der rechten Seite werden die konfigurierbaren Einstellungen für die ausgewählte Komponente angezeigt.
5. Erkunden Sie die verschiedenen verfügbaren Eigenschaften und Optionen und aktualisieren Sie sie bei Bedarf, um das Aussehen und Verhalten der Komponente zu konfigurieren. Möglicherweise möchten Sie beispielsweise die Datenquelle ändern, das Layout konfigurieren oder zusätzliche Funktionen aktivieren.

   Ausführliche Informationen zur Konfiguration der einzelnen Komponententypen finden Sie unterReferenz zu den Komponenten.

Komponenten löschen

1. Navigieren Sie bei Bedarf zur Entwicklungsumgebung Ihrer Anwendung, indem Sie sie bearbeiten.
2. Navigieren Sie zur Registerkarte Seiten.
3. Wählen Sie im Seitenmenü auf der linken Seite die Komponente aus, die gelöscht werden soll, um sie auszuwählen.
4. Wählen Sie im Eigenschaftenmenü auf der rechten Seite das Papierkorbsymbol aus.
5. Wählen Sie im Bestätigungs-Dialogfeld die Option Delete (Löschen).

Konfiguration der rollenbasierten Sichtbarkeit von Seiten

Sie können Rollen in einer App Studio-App erstellen und die Sichtbarkeit von Seiten auf der Grundlage dieser Rollen konfigurieren. Sie können beispielsweise Rollen auf der Grundlage von Benutzeranforderungen oder Zugriffsebenen wie Administrator, Manager oder Benutzer für Apps erstellen, die Funktionen wie Projektgenehmigungen oder die Bearbeitung von Ansprüchen bereitstellen, und bestimmte Seiten für bestimmte Rollen sichtbar machen. In diesem Beispiel haben Administratoren möglicherweise Vollzugriff, Manager haben möglicherweise Zugriff auf Berichts-Dashboards und Benutzer haben möglicherweise Zugriff auf Aufgabenseiten mit Eingabeformularen.

Gehen Sie wie folgt vor, um die rollenbasierte Sichtbarkeit von Seiten in Ihrer App Studio-App zu konfigurieren.

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung. Wählen Sie im Navigationsmenü auf der linken Seite Meine Anwendungen aus, suchen Sie nach Ihrer Bewerbung und wählen Sie Bearbeiten.
2. Erstellen Sie Rollen auf App-Ebene im Application Studio.
   a. Wählen Sie oben im Anwendungsstudio die Registerkarte App-Einstellungen aus.
   b. Wählen Sie + Rolle hinzufügen
   c. Geben Sie im Feld Rollenname einen Namen zur Identifizierung Ihrer Rolle ein. Wir empfehlen, einen Namen zu verwenden, der die Zugriffsebene oder die Aufgaben der Gruppe beschreibt, da Sie den Namen verwenden, um die Sichtbarkeit der Seite einzurichten.
   d. Fügen Sie optional unter Beschreibung eine Beschreibung für die Rolle hinzu.
   e. Wiederholen Sie diese Schritte, um so viele Rollen wie nötig zu erstellen.
3. Konfigurieren Sie die Sichtbarkeit Ihrer Seiten
   a. Wählen Sie oben im Anwendungsstudio die Registerkarte Seiten.
   b. Wählen Sie im Seitenmenü auf der linken Seite die Seite aus, für die Sie die rollenbasierte Sichtbarkeit konfigurieren möchten.
   c. Wählen Sie im Menü auf der rechten Seite die Registerkarte Eigenschaften aus.
   d. Deaktivieren Sie unter Sichtbarkeit die Option Für alle Endbenutzer öffnen.
   e. Lassen Sie die Option Rolle ausgewählt, um aus einer Liste der Rollen auszuwählen, die Sie im vorherigen Schritt erstellt haben. Wählen Sie Benutzerdefiniert, um einen JavaScript Ausdruck für komplexere Sichtbarkeitskonfigurationen zu schreiben.
      1. Wenn Rolle ausgewählt ist, markieren Sie die Kästchen der App-Rollen, für die die Seite sichtbar sein soll.
      2. Wenn Benutzerdefiniert ausgewählt ist, geben Sie einen JavaScript Ausdruck ein, der als wahr oder falsch aufgelöst wird. Überprüfen Sie anhand des folgenden Beispiels, ob der aktuelle Benutzer die Rolle eines Managers innehat:`{{currentUser.roles.includes('manager')}}`.
4. Nachdem Ihre Sichtbarkeit konfiguriert ist, können Sie die Sichtbarkeit der Seite testen, indem Sie sich eine Vorschau Ihrer App ansehen.
   a. Wählen Sie Vorschau, um eine Vorschau Ihrer App zu öffnen.

b. Wählen Sie oben rechts in der Vorschau das Menü Vorschau als und markieren Sie die Kästchen der Rollen, die Sie testen möchten. Die sichtbaren Seiten sollten die ausgewählten Rollen widerspiegeln.

5. Weisen Sie nun Gruppen App-Rollen für eine veröffentlichte App zu. Gruppen- und Rollenzuweisungen müssen für jede Umgebung separat konfiguriert werden. Weitere Informationen zu App-Umgebungen finden Sie unterAnwendungsumgebungen.

> **Note**
>
> Ihre App muss entweder in der Test- oder Produktionsumgebung veröffentlicht werden, um App Studio-Gruppen den Rollen zuzuweisen, die Sie erstellt und konfiguriert haben. Veröffentlichen Sie bei Bedarf Ihre App, um den Rollen Gruppen zuzuweisen. Weitere Informationen zum Veröffentlichen finden Sie unter Anwendungen veröffentlichen.

a. Wählen Sie oben rechts im Anwendungsstudio die Option Teilen aus.

b. Wählen Sie die Registerkarte für die Umgebung aus, für die Sie die Sichtbarkeit der Seite konfigurieren möchten.

c. Wählen Sie das Eingabefeld Suchgruppen aus und wählen Sie die Gruppe aus, mit der Sie die App-Version teilen möchten. Sie können Text eingeben, um nach Gruppen zu suchen.

d. Wählen Sie im Dropdownmenü die Rollen aus, die Sie der Gruppe zuweisen möchten. Sie können „Keine Rolle“ wählen, um die App-Version zu teilen und der Gruppe keine Rolle zuzuweisen. Nur Seiten, die für alle Benutzer sichtbar sind, sind für Gruppen ohne Rolle sichtbar.

e. Wählen Sie Freigeben. Wiederholen Sie diese Schritte, um so viele Gruppen wie nötig hinzuzufügen.

Seiten in der App-Navigation ordnen und organisieren

Dieses Thema enthält Informationen zum Neuanordnen und Organisieren von Seiten in App Studio-Anwendungen. Es gibt zwei Bereiche des Produkts, in denen App-Seiten angezeigt werden: im Seitenmenü auf der linken Seite, wenn Sie die App im Anwendungsstudio bearbeiten, und in der linken Navigationsleiste einer Vorschau der veröffentlichten App.

Seiten beim Bearbeiten einer App im Seitenmenü auf der linken Seite anordnen

Bei der Bearbeitung einer App im Anwendungsstudio werden die Seiten im Seitenmenü auf der linken Seite nach Erstellungszeit sortiert. Sie können die Seiten in diesem Menü nicht neu anordnen.

Seiten in der Navigation einer Vorschau-App oder einer veröffentlichten App anordnen, ein- oder ausblenden

Sie können die folgenden Einstellungen der linken Navigation einer Vorschau-App oder einer veröffentlichten App bearbeiten:

- Die Sichtbarkeit der gesamten Navigation
- Die Sichtbarkeit bestimmter Seiten in der Navigation
- Die Reihenfolge der Seiten in der Navigation

Um die linke Navigation einer Vorschau-App oder einer veröffentlichten App zu bearbeiten

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung, um sie zu bearbeiten.
2. Wählen Sie im Seitenmenü auf der linken Seite die Option Kopfzeile und Navigation aus.
3. Sehen Sie sich im Menü Kopfzeile und Navigation auf der rechten Seite Folgendes an oder bearbeiten Sie es:
    a. Verwenden Sie den Schalter für die App-Navigation, um die Navigation in der App ein- oder auszublenden.
    b. Um Seiten aus der Navigation der App auszublenden, ziehen Sie die Seiten in den Bereich Unverknüpfte Seiten.
    c. Um Seiten in der Navigation der App neu anzuordnen, ziehen Sie sie im Bereich Verlinkte Seiten in die gewünschte Reihenfolge.

Ändern Sie die Farben in Ihrer App mit App-Themen

Gehen Sie wie folgt vor, um die Farben in Ihrer Anwendung zu aktualisieren, indem Sie ein App-Thema konfigurieren.

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer App, um sie zu bearbeiten.
2. Navigieren Sie im Anwendungsstudio zur Registerkarte Seiten.

3. Wählen Sie in der linken Navigationsleiste App-Design aus, um die Einstellungen für das App-Thema auf der rechten Seite zu öffnen.
4. Wählen Sie im Basisdesign den hellen Modus oder den dunklen Modus.
5. Um Ihrer Anwendung benutzerdefinierte Farben hinzuzufügen, aktivieren Sie den Schalter Anpassen und aktualisieren Sie die folgenden Einstellungen:

   a. Wählen Sie unter Primärfarbe die Farbe aus, die auf bestimmte Komponenten und die Navigation Ihrer App angewendet wird. Sie können eine Farbe mit dem Farbwähler, RGB-, HSL- oder HEX-Code auswählen.

   > **Note**
   >
   > App Studio stellt automatisch sicher, dass auf Ihre Farben zugegriffen werden kann. Wenn Sie beispielsweise im Lichtmodus eine Lichtfarbe auswählen, wird sie aktualisiert, sodass sie leichter zugänglich ist.

   b. Wählen Sie unter Kopfzeilenfarbe die Farbe aus, die auf den Header Ihrer App angewendet wird. Sie können eine Farbe mit dem Farbwähler, RGB-, HSL- oder HEX-Code auswählen.

   c. Wählen Sie Standarddesigns zur Anzeige und Auswahl aus vordefinierten Designs oder wählen Sie „Zufällig“, um eine zufällige Primär- und Kopfzeilenfarbe zu generieren.
6. Wählen Sie „Änderungen speichern“, um Ihr App-Design zu aktualisieren.

Referenz zu den Komponenten

Dieses Thema beschreibt die einzelnen Komponenten von App Studio, ihre Eigenschaften und enthält Konfigurationsbeispiele.

Allgemeine Eigenschaften von Komponenten

In diesem Abschnitt werden die allgemeinen Eigenschaften und Funktionen beschrieben, die von mehreren Komponenten im Application Studio gemeinsam genutzt werden. Die spezifischen Implementierungsdetails und Anwendungsfälle für jeden Eigenschaftstyp können je nach Komponente variieren, aber das allgemeine Konzept dieser Eigenschaften bleibt in App Studio einheitlich.

Name

Für jede Komponente wird ein Standardname generiert. Sie können ihn jedoch bearbeiten, um für jede Komponente einen eindeutigen Namen zu wählen. Sie verwenden diesen Namen, um auf die Komponente und ihre Daten aus anderen Komponenten oder Ausdrücken auf derselben Seite zu verweisen. Einschränkung: Der Komponentenname darf keine Leerzeichen enthalten. Er darf nur Buchstaben, Zahlen, Unterstriche und Dollarzeichen enthalten. Beispiele: `userNameInput`, `ordersTable`, `metricCard1`.

Primärwert, Sekundärwert und Wert

Viele Komponenten im Application Studio bieten Felder zur Angabe von Werten oder Ausdrücken, die den Inhalt oder die Daten bestimmen, die in der Komponente angezeigt werden. Diese Felder werden je nach Komponententyp und Verwendungszweck häufig als `Primary value` oder einfach `Value` bezeichnet. `Secondary value`

Das `Primary value` Feld wird in der Regel verwendet, um den Hauptwert, den Datenpunkt oder den Inhalt zu definieren, der in der Komponente gut sichtbar angezeigt werden soll.

Das `Secondary value` Feld wird, sofern verfügbar, verwendet, um neben dem Primärwert einen zusätzlichen oder unterstützenden Wert oder eine Information anzuzeigen.

In dem `Value` Feld können Sie den Wert oder Ausdruck angeben, der in der Komponente angezeigt werden soll.

Diese Felder unterstützen sowohl statische Texteingaben als auch dynamische Ausdrücke. Mithilfe von Ausdrücken können Sie auf Daten aus anderen Komponenten, Datenquellen oder Variablen in Ihrer Anwendung verweisen und so eine dynamische und datengesteuerte Inhaltsanzeige ermöglichen.

Syntax für Ausdrücke

Die Syntax für die Eingabe von Ausdrücken in diese Felder folgt einem konsistenten Muster:

```
{{expression}}
```

Wo *`expression`* ist ein gültiger Ausdruck, der den gewünschten Wert oder die gewünschten Daten ergibt, die Sie anzeigen möchten?

Beispiel: Statischer Text

- Primärwert: Sie können eine statische Zahl oder einen statischen Wert direkt eingeben, z. B. `"123"` oder`"$1,999.99"`.
- Sekundärer Wert: Sie können eine statische Textbeschriftung eingeben, z. B. `"Goal"` oder`"Projected Revenue"`.
- Wert: Sie können eine statische Zeichenfolge eingeben, z. B. `"since last month"` oder`"Total Quantity"`.

Beispiele: Ausdrücke

- `Hello, {{currentUser.firstName}}`: Zeigt eine Begrüßung mit dem Vornamen des aktuell angemeldeten Benutzers an.
- `{{currentUser.role === 'Admin' ? 'Admin Dashboard' : 'User Dashboard'}}`: Zeigt abhängig von der Rolle des Benutzers bedingt einen anderen Dashboard-Titel an.
- `{{ui.componentName.data?.[0]?.fieldName}}`: Ruft den Wert des `fieldName` Felds aus dem ersten Element in den Daten der Komponente mit der ID ab. `componentName`
- `{{ui.componentName.value * 100}}`: Führt eine Berechnung für den Wert der Komponente mit der ID `componentName` durch.
- `{{ui.componentName.value + ' items'}}`: Verkettet den Wert der Komponente mit der ID `componentName` und der Zeichenfolge. `' items'`
- `{{ui.ordersTable.data?.[0]?.orderNumber}}`: Ruft die Bestellnummer aus der ersten Datenzeile in der Komponente ab. `ordersTable`
- `{{ui.salesMetrics.data?.[0]?.totalRevenue * 1.15}}`: Berechnet den prognostizierten Umsatz, indem der Gesamtumsatz aus der ersten Datenzeile in der `salesMetrics` Komponente um 15% erhöht wird.
- `{{ui.customerProfile.data?.[0]?.firstName + ' ' + ui.customerProfile.data?.lastName}}`: Verkettet den Vor- und Nachnamen aus den Daten in der Komponente. `customerProfile`
- `{{new Date(ui.orderDetails.data?.orderDate).toLocaleDateString()}}`: Formatiert das Bestelldatum der `orderDetails` Komponente in eine besser lesbare Datumszeichenfolge.
- `{{ui.productList.data?.length}}`: Zeigt die Gesamtzahl der Produkte in den mit der `productList` Komponente verbundenen Daten an.

- `{{ui.discountPercentage.value * ui.orderTotal.value}}`: Berechnet den Rabattbetrag auf der Grundlage des Rabattprozentsatzes und der Gesamtsumme der Bestellung.
- `{{ui.cartItemCount.value + ' items in cart'}}`: Zeigt die Anzahl der Artikel im Warenkorb zusammen mit dem Etikett `items in cart` an.

Mithilfe dieser Ausdrucksfelder können Sie dynamische und datengesteuerte Inhalte in Ihrer Anwendung erstellen, sodass Sie Informationen anzeigen können, die auf den Benutzerkontext oder den Status Ihrer Anwendung zugeschnitten sind. Dies ermöglicht personalisiertere und interaktivere Benutzererlebnisse.

Label (Bezeichnung)

Mit der Label-Eigenschaft können Sie eine Überschrift oder einen Titel für die Komponente angeben. Diese Bezeichnung wird in der Regel neben oder über der Komponente angezeigt, sodass Benutzer ihren Zweck besser verstehen.

Sie können sowohl statischen Text als auch Ausdrücke verwenden, um die Bezeichnung zu definieren.

Beispiel: Statischer Text

Wenn Sie den Text „Vorname“ in das Feld Bezeichnung eingeben, zeigt die Komponente „Vorname“ als Bezeichnung an.

Beispiel: Ausdrücke

Beispiel: Einzelhandelsgeschäft

Im folgenden Beispiel wird das Etikett für jeden Benutzer personalisiert, sodass sich die Benutzeroberfläche besser auf die Person zugeschnitten anfühlt:

```
{{currentUser.firstName}} {{currentUser.lastName}}'s Account
```

Beispiel: SaaS-Projektmanagement

Im folgenden Beispiel werden Daten aus dem ausgewählten Projekt abgerufen, um kontextspezifische Bezeichnungen bereitzustellen, sodass Benutzer sich in der Anwendung orientieren können:

```
Project {{ui.projectsTable.selectedRow.id}} - {{ui.projectsTable.selectedRow.name}}
```

Beispiel: Gesundheitsklinik

Das folgende Beispiel bezieht sich auf das aktuelle Benutzerprofil und die Informationen des Arztes, um den Patienten ein persönlicheres Erlebnis zu bieten.

```
Dr. {{ui.doctorProfileTable.data.firstName}}
       {{ui.doctorProfileTable.data.lastName}}
```

Placeholder

Mit der Placeholder-Eigenschaft können Sie einen Hinweis- oder Hinweistext angeben, der in der Komponente angezeigt wird, wenn diese leer ist. Dies kann Benutzern helfen, das erwartete Eingabeformat zu verstehen, oder zusätzlichen Kontext bereitstellen.

Sie können sowohl statischen Text als auch Ausdrücke verwenden, um den Platzhalter zu definieren.

Beispiel: Statischer Text

Wenn Sie den Text `Enter your name` in das Platzhalterfeld eingeben, wird die Komponente `Enter your name` als Platzhaltertext angezeigt.

Beispiel: Ausdrücke

Beispiel: Finanzdienstleistungen

`Enter the amount you'd like to deposit into your {{ui.accountsTable.selectedRow.balance}} account`In diesen Beispielen werden Daten aus dem ausgewählten Konto abgerufen, um entsprechende Eingabeaufforderungen anzuzeigen, sodass die Benutzeroberfläche für Bankkunden intuitiv ist.

Beispiel: E-Commerce

`Enter the coupon code for {{ui.cartTable.data.currency}} total`Der Platzhalter wird hier dynamisch auf der Grundlage des Warenkorbinhalts des Benutzers aktualisiert und sorgt so für ein reibungsloses Checkout-Erlebnis.

Beispiel: Gesundheitsklinik

`Enter your {{ui.patientProfile.data.age}}-year-old patient's symptoms`Durch die Verwendung eines Ausdrucks, der auf das Alter des Patienten verweist, kann die Anwendung einen personalisierteren und hilfreicheren Platzhalter erstellen.

Quelle

Mit der Eigenschaft Source können Sie die Datenquelle für eine Komponente auswählen. Bei der Auswahl können Sie aus den folgenden Datenquellentypen wählen: `entity`expression, oder`automation`.

Entität

Wenn Sie Entität als Datenquelle auswählen, können Sie die Komponente mit einer vorhandenen Datenentität oder einem Modell in Ihrer Anwendung verbinden. Dies ist nützlich, wenn Sie über eine klar definierte Datenstruktur oder ein klar definiertes Schema verfügen, das Sie in Ihrer gesamten Anwendung nutzen möchten.

Wann sollte die Entitätsdatenquelle verwendet werden:

- Wenn Sie über ein Datenmodell oder eine Entität verfügen, die die Informationen enthält, die Sie in der Komponente anzeigen möchten (z. B. eine Entität „Produkte“ mit Feldern wie „Name“, „Beschreibung“, „Preis“).
- Wenn Sie Daten dynamisch aus einer Datenbank, API oder einer anderen externen Datenquelle abrufen und in der Komponente präsentieren müssen.
- Wenn Sie die im Datenmodell Ihrer Anwendung definierten Beziehungen und Verknüpfungen nutzen möchten.

Eine Abfrage für eine Entität auswählen

Manchmal möchten Sie vielleicht eine Komponente mit einer bestimmten Abfrage verbinden, die Daten von einer Entität und nicht von der gesamten Entität abruft. In der Entitätsdatenquelle haben Sie die Möglichkeit, aus vorhandenen Abfragen auszuwählen oder eine neue zu erstellen.

Durch die Auswahl einer Abfrage können Sie:

- Filtern Sie die in der Komponente angezeigten Daten nach bestimmten Kriterien.
- Übergeben Sie Parameter an die Abfrage, um die Daten dynamisch zu filtern oder zu sortieren.
- Nutzen Sie komplexe Verknüpfungen, Aggregationen oder andere in der Abfrage definierte Datenmanipulationstechniken.

Zum Beispiel, wenn Sie in Ihrer Anwendung eine `Customers` Entität mit Feldern wie `Name`, `Email`, und `PhoneNumber` haben. Sie können eine Tabellenkomponente mit dieser Entität verbinden und

eine vordefinierte `ActiveCustomers` Datenaktion auswählen, die die Kunden anhand ihres Status filtert. Auf diese Weise können Sie nur die aktiven Kunden in der Tabelle anzeigen und nicht die gesamte Kundendatenbank.

Hinzufügen von Parametern zu einer Entitätsdatenquelle

Wenn Sie eine Entität als Datenquelle verwenden, können Sie der Komponente auch Parameter hinzufügen. Diese Parameter können verwendet werden, um die in der Komponente angezeigten Daten zu filtern, zu sortieren oder zu transformieren.

Wenn Sie beispielsweise eine `Products` Entität mit Feldern wie`Name`, `DescriptionPrice`, und haben`Category`. Sie können einer Tabellenkomponente`category`, die die Produktliste anzeigt, einen Parameter mit dem Namen hinzufügen. Wenn Benutzer eine Kategorie aus einer Dropdownliste auswählen, wird die Tabelle automatisch aktualisiert, sodass nur die Produkte angezeigt werden, die zu der ausgewählten Kategorie gehören. Dabei wird der `{{params.category}}` Ausdruck in der Datenaktion verwendet.

Expression

Wählen Sie Ausdruck als Datenquelle aus, um benutzerdefinierte Ausdrücke oder Berechnungen einzugeben, um die Daten für die Komponente dynamisch zu generieren. Dies ist nützlich, wenn Sie Transformationen durchführen, Daten aus mehreren Quellen kombinieren oder Daten auf der Grundlage einer bestimmten Geschäftslogik generieren müssen.

Wann sollte die Expression-Datenquelle verwendet werden:

- Wenn Sie Daten berechnen oder ableiten müssen, die nicht direkt in Ihrem Datenmodell verfügbar sind (z. B. Berechnung des Gesamtbestellwerts auf der Grundlage von Menge und Preis).
- Wenn Sie Daten aus mehreren Entitäten oder Datenquellen kombinieren möchten, um eine zusammengesetzte Ansicht zu erstellen (z. B. die Bestellhistorie eines Kunden zusammen mit seinen Kontaktinformationen anzuzeigen).
- Wenn Sie Daten auf der Grundlage bestimmter Regeln oder Bedingungen generieren müssen (z. B. Anzeige einer Liste mit „empfohlenen Produkten“, die auf dem Browserverlauf des Benutzers basiert).

Wenn Sie beispielsweise über eine *`Metrics`* Komponente verfügen, die den Gesamtumsatz für den aktuellen Monat anzeigen muss, können Sie einen Ausdruck wie den folgenden verwenden, um den monatlichen Umsatz zu berechnen und anzuzeigen:

```
{{ui.table1.orders.concat(ui.table1.orderDetails).filter(o => o.orderDate.getMonth()
 === new Date().getMonth()).reduce((a, b) => a + (b.quantity * b.unitPrice), 0)}}
```

Automatisierung

Wählen Sie Automatisierung als Datenquelle aus, um die Komponente mit einer vorhandenen Automatisierung oder einem Workflow in Ihrer Anwendung zu verbinden. Dies ist nützlich, wenn die Daten oder Funktionen für die Komponente als Teil eines bestimmten Prozesses oder Workflows generiert oder aktualisiert werden.

Wann sollte die Automatisierungs-Datenquelle verwendet werden:

- Wenn die in der Komponente angezeigten Daten das Ergebnis einer bestimmten Automatisierung oder eines bestimmten Workflows sind (z. B. eine Tabelle „Ausstehende Genehmigungen“, die im Rahmen eines Genehmigungsprozesses aktualisiert wird).
- Wenn Sie aufgrund von Ereignissen oder Bedingungen innerhalb einer Automatisierung Aktionen oder Aktualisierungen an der Komponente auslösen möchten (z. B. Aktualisierung einer Metrik mit den neuesten Verkaufszahlen für eine SKU).
- Wenn Sie die Komponente mithilfe einer Automatisierung in andere Dienste oder Systeme in Ihrer Anwendung integrieren müssen (z. B. Daten von einer Drittanbieter-API abrufen und in einer Tabelle anzeigen).

Zum Beispiel, wenn Sie über eine Stepflow-Komponente verfügen, die Benutzer durch einen Bewerbungsprozess führt. Die Stepflow-Komponente kann mit einer Automatisierung verbunden werden, die die Einreichung von Bewerbungen, die Hintergrundprüfungen und die Angebotserstellung übernimmt. Während die Automatisierung diese Schritte durchläuft, kann die Stepflow-Komponente dynamisch aktualisiert werden, um den aktuellen Status der Bewerbung widerzuspiegeln.

Durch die sorgfältige Auswahl der geeigneten Datenquelle für jede Komponente können Sie sicherstellen, dass die Benutzeroberfläche Ihrer Anwendung auf den richtigen Daten und der richtigen Logik basiert und Ihren Benutzern ein nahtloses und ansprechendes Erlebnis bietet.

Sichtbar wenn

Verwenden Sie die Eigenschaft Sichtbar, wenn Sie Komponenten oder Elemente auf der Grundlage bestimmter Bedingungen oder Datenwerte ein- oder ausblenden möchten. Dies ist nützlich, wenn Sie die Sichtbarkeit bestimmter Teile der Benutzeroberfläche Ihrer Anwendung dynamisch steuern möchten.

Die Eigenschaft Visible if verwendet die folgende Syntax:

```
{{expression ? true : false}}
```

or

```
{{expression}}
```

Wo *expression* ist ein boolescher Ausdruck, der entweder oder ergibt? `true` `false`

Wenn der Ausdruck zu ausgewertet wird`true`, ist die Komponente sichtbar. Wenn der Ausdruck als Ergebnis ausgewertet wird`false`, wird die Komponente ausgeblendet. Der Ausdruck kann auf Werte aus anderen Komponenten, Datenquellen oder Variablen in Ihrer Anwendung verweisen.

Sichtbar bei Ausdrucksbeispielen

Beispiel: Ein- oder Ausblenden eines Passwort-Eingabefeldes basierend auf einer E-Mail-Eingabe

Stellen Sie sich vor, Sie haben ein Anmeldeformular mit einem E-Mail-Eingabefeld und einem Passwort-Eingabefeld. Sie möchten das Passwort-Eingabefeld nur anzeigen, wenn der Benutzer eine E-Mail-Adresse eingegeben hat. Sie können den folgenden Ausdruck Visible if verwenden:

```
{{ui.emailInput.value !== ""}}
```

Dieser Ausdruck prüft, ob der Wert der `emailInput` Komponente keine leere Zeichenfolge ist. Wenn der Benutzer eine E-Mail-Adresse eingegeben hat, wird der Ausdruck zu ausgewertet`true`, und das Passworteingabefeld ist sichtbar. Wenn das E-Mail-Feld leer ist, wird der Ausdruck zu ausgewertet`false`, und das Passworteingabefeld wird ausgeblendet.

Beispiel: Anzeige zusätzlicher Formularfelder auf der Grundlage einer Dropdownauswahl

Nehmen wir an, Sie haben ein Formular, in dem Benutzer eine Kategorie aus einer Dropdownliste auswählen können. Abhängig von der ausgewählten Kategorie möchten Sie zusätzliche Formularfelder ein- oder ausblenden, um spezifischere Informationen zu sammeln.

Wenn der Benutzer beispielsweise die *Products* Kategorie auswählt, können Sie den folgenden Ausdruck verwenden, um ein zusätzliches *Product Details* Feld anzuzeigen:

```
{{ui.categoryDropdown.value === "Products"}}
```

Wenn der Benutzer die *Consulting* Kategorien *Services* oder auswählt, können Sie diesen Ausdruck verwenden, um einen anderen Satz zusätzlicher Felder anzuzeigen:

```
{{ui.categoryDropdown.value === "Services" || ui.categoryDropdown.value ===
 "Consulting"}}
```

Beispiele: Andere

Um die Komponente sichtbar zu machen, wenn der Wert der `textInput1` Komponente keine leere Zeichenfolge ist:

```
{{ui.textInput1.value === "" ? false : true}}
```

Um die Komponente immer sichtbar zu machen:

```
{{true}}
```

Um die Komponente sichtbar zu machen, wenn der Wert der `emailInput` Komponente keine leere Zeichenfolge ist:

```
{{ui.emailInput.value !== ""}}
```

Deaktiviert wenn

Mit der Funktion Deaktiviert wenn können Sie eine Komponente auf der Grundlage bestimmter Bedingungen oder Datenwerte bedingt aktivieren oder deaktivieren. Dies wird durch die Verwendung der Eigenschaft Disabled if erreicht, die einen booleschen Ausdruck akzeptiert, der bestimmt, ob die Komponente aktiviert oder deaktiviert werden soll.

Die Eigenschaft Disabled if verwendet die folgende Syntax:

```
{{expression ? true : false}}
```

or

```
{{expression}}
```

Deaktiviert, wenn Ausdrucksbeispiele

Beispiel: Deaktivierung einer Schaltfläche zum Absenden auf der Grundlage einer Formularüberprüfung

Wenn Sie ein Formular mit mehreren Eingabefeldern haben und die Schaltfläche „Senden“ deaktivieren möchten, bis alle erforderlichen Felder korrekt ausgefüllt sind, können Sie den folgenden Disabled If-Ausdruck verwenden:

```
{{ui.nameInput.value === "" || ui.emailInput.value === "" || ui.passwordInput.value ===
 ""}}
```

Dieser Ausdruck überprüft, ob eines der erforderlichen Eingabefelder (`nameInput`,`emailInput`,`passwordInput`) leer ist. Wenn eines der Felder leer ist, wird der Ausdruck zu ausgewertet`true`, und die Schaltfläche „Senden“ wird deaktiviert. Sobald alle erforderlichen Felder ausgefüllt sind, wird der Ausdruck zu `false` ausgewertet und die Schaltfläche „Senden“ wird aktiviert.

Durch die Verwendung dieser Typen von bedingten Ausdrücken in den Eigenschaften Visible if und Disabled ff können Sie dynamische und responsive Benutzeroberflächen erstellen, die sich an Benutzereingaben anpassen und den Benutzern Ihrer Anwendung ein optimiertes und relevanteres Erlebnis bieten.

Wo *`expression`* ist ein boolescher Ausdruck, der entweder als wahr oder falsch ausgewertet wird?

Beispiel:

```
{{ui.textInput1.value === "" ? true : false}}: The component will be Disabled if the
 textInput1 component's value is an empty string.
{{!ui.nameInput.isValid || !ui.emailInput.isValid || !ui.passwordInput.isValid}}: The
 component will be Disabled if any of the named input fields are invalid.
```

Layouts von Containern

Die Layouteigenschaften bestimmen, wie der Inhalt oder die Elemente innerhalb einer Komponente angeordnet und positioniert werden. Es sind mehrere Layoutoptionen verfügbar, die jeweils durch ein Symbol dargestellt werden:

- Spaltenlayout: Dieses Layout ordnet den Inhalt oder die Elemente vertikal in einer einzigen Spalte an.

- Zweispaltiges Layout: Dieses Layout unterteilt die Komponente in zwei Spalten gleicher Breite, sodass Sie Inhalte oder Elemente nebeneinander positionieren können.
- Zeilenlayout: Dieses Layout ordnet den Inhalt oder die Elemente horizontal in einer einzigen Zeile an.

Ausrichtung

- Horizontal: Dieses Layout ordnet den Inhalt oder die Elemente horizontal in einer einzigen Zeile an.
- Vertikal: Dieses Layout ordnet den Inhalt oder die Elemente vertikal in einer einzigen Spalte an.
- Zeilenumbruch: Dieses Layout ordnet den Inhalt oder die Elemente horizontal an, wird jedoch zur nächsten Zeile umgebrochen, wenn die Elemente die verfügbare Breite überschreiten.

Ausrichtung

- Links: Richtet den Inhalt oder die Elemente an der linken Seite der Komponente aus.
- Mittig: Zentriert den Inhalt oder die Elemente horizontal innerhalb der Komponente.
- Rechts: Richtet den Inhalt oder die Elemente an der rechten Seite der Komponente aus.

Width

Die Eigenschaft Width gibt die horizontale Größe der Komponente an. Sie können einen Prozentwert zwischen 0 und 100% eingeben, der die Breite der Komponente im Verhältnis zum übergeordneten Container oder zum verfügbaren Speicherplatz darstellt.

Höhe

Die Eigenschaft Height gibt die vertikale Größe der Komponente an. Der Wert „auto“ passt die Höhe der Komponente automatisch an ihren Inhalt oder den verfügbaren Platz an.

Abstand zwischen

Die Eigenschaft „Abstand zwischen“ bestimmt den Abstand oder die Lücke zwischen dem Inhalt oder den Elementen innerhalb der Komponente. Sie können einen Wert zwischen 0 px (kein Abstand) und 64 px mit Schritten von 4 px (z. B. 4 px, 8 px, 12 px usw.) auswählen.

Padding

Die Eigenschaft Padding steuert den Abstand zwischen dem Inhalt oder den Elementen und den Kanten der Komponente. Sie können einen Wert zwischen 0px (ohne Polsterung) und 64px mit Schritten von 4px (z. B. 4px, 8px, 12px usw.) wählen.

Hintergrund

Der Hintergrund aktiviert oder deaktiviert eine Hintergrundfarbe oder einen Stil für die Komponente.

Diese Layouteigenschaften bieten Flexibilität bei der Anordnung und Positionierung des Inhalts innerhalb einer Komponente sowie bei der Steuerung der Größe, des Abstands und der visuellen Darstellung der Komponente selbst.

Datenkomponenten

In diesem Abschnitt werden die verschiedenen Datenkomponenten behandelt, die im Application Studio verfügbar sind, darunter die Komponenten Table, Detail, Metric, Form und Repeater. Diese Komponenten werden verwendet, um Daten in Ihrer Anwendung anzuzeigen, zu sammeln und zu bearbeiten.

Tabelle

Die Tabellenkomponente zeigt Daten in einem tabellarischen Format mit Zeilen und Spalten an. Sie wird verwendet, um strukturierte Daten, wie z. B. Listen mit Elementen oder Datensätzen aus einer Datenbank, strukturiert und easy-to-read strukturiert darzustellen.

Tabelleneigenschaften

Die Tabellenkomponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten wie `NameSource`, und`Actions`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Zusätzlich zu den allgemeinen Eigenschaften verfügt die Tabellenkomponente über spezifische Eigenschaften und Konfigurationsoptionen, darunter `ColumnsSearch and export`, und`Expressions`.

Spalten

In diesem Abschnitt können Sie die Spalten definieren, die in der Tabelle angezeigt werden sollen. Jede Spalte kann mit den folgenden Eigenschaften konfiguriert werden:

- Format: Der Datentyp des Felds, zum Beispiel: Text, Zahl, Datum.
- Spaltenbezeichnung: Der Kopfzeilentext für die Spalte.
- Wert: Das Feld aus der Datenquelle, das in dieser Spalte angezeigt werden soll.

  In diesem Feld können Sie den Wert oder Ausdruck angeben, der in den Spaltenzellen angezeigt werden soll. Sie können Ausdrücke verwenden, um auf Daten aus der verbundenen Quelle oder anderen Komponenten zu verweisen.

  Beispiel: `{{currentRow.title}}` - Dieser Ausdruck zeigt den Wert des `title` Feldes aus der aktuellen Zeile in den Spaltenzellen an.
- Sortierung aktivieren: Mit diesem Schalter können Sie die Sortierfunktion für die jeweilige Spalte aktivieren oder deaktivieren. Wenn diese Option aktiviert ist, können Benutzer die Tabellendaten anhand der Werte in dieser Spalte sortieren.

Suchen und exportieren

Die Tabellenkomponente bietet die folgenden Schalter zum Aktivieren oder Deaktivieren der Such- und Exportfunktionen:

- Suche anzeigen Wenn diese Option aktiviert ist, fügt sie der Tabelle ein Sucheingabefeld hinzu, sodass Benutzer die angezeigten Daten suchen und filtern können.
- Export anzeigen Wenn diese Option aktiviert ist, fügt sie der Tabelle eine Exportoption hinzu, sodass Benutzer die Tabellendaten in verschiedenen Formaten herunterladen können, zum Beispiel: CSV.

> **Note**
>
> Standardmäßig ist die Suchfunktion auf die Daten beschränkt, die in die Tabelle geladen wurden. Um die Suche vollständig nutzen zu können, müssen Sie alle Datenseiten laden.

Zeilen pro Seite

Sie können die Anzahl der Zeilen angeben, die pro Seite in der Tabelle angezeigt werden sollen. Benutzer können dann zwischen den Seiten navigieren, um den gesamten Datensatz anzuzeigen.

Limit vor dem Abrufen

Geben Sie die maximale Anzahl von Datensätzen an, die in jeder Abfrageanforderung vorab abgerufen werden sollen. Das Maximum ist 3000.

Aktionen

Konfigurieren Sie im Abschnitt Aktionen die folgenden Eigenschaften:

- Ort der Aktion: Wenn „Nach rechts anheften“ aktiviert ist, werden alle hinzugefügten Aktionen immer auf der rechten Seite der Tabelle angezeigt, unabhängig davon, ob der Benutzer scrollt.
- Aktionen: Fügen Sie der Tabelle Aktionsschaltflächen hinzu. Sie können diese Schaltflächen so konfigurieren, dass sie bestimmte Aktionen ausführen, wenn sie von einem Benutzer angeklickt werden, z. B.:
  - Führen Sie eine Komponentenaktion aus
  - Navigieren Sie zu einer anderen Seite
  - Rufen Sie eine Datenaktion auf
  - Benutzerdefiniert ausführen JavaScript
  - Rufen Sie eine Automatisierung auf

Ausdrücke

Die Tabellenkomponente bietet mehrere Bereiche zur Verwendung von Ausdrücken und Aktionsfunktionen auf Zeilenebene, mit denen Sie die Funktionalität und Interaktivität der Tabelle anpassen und verbessern können. Sie ermöglichen es Ihnen, dynamisch auf Daten in der Tabelle zu verweisen und sie anzuzeigen. Mithilfe dieser Ausdrucksfelder können Sie dynamische Spalten erstellen, Daten an Aktionen auf Zeilenebene übergeben und Tabellendaten aus anderen Komponenten oder Ausdrücken in Ihrer Anwendung referenzieren.

Beispiele: Referenzieren von Zeilenwerten

`{{currentRow.columnName}}`oder Mit `{{currentRow["Column Name"]}}` diesen Ausdrücken können Sie auf den Wert einer bestimmten Spalte für die aktuelle Zeile verweisen, die gerade gerendert wird. Ersetzen Sie *`columnName`* oder *`Column Name`* durch den tatsächlichen Namen der Spalte, auf die Sie verweisen möchten.

Beispiele:

- `{{currentRow.productName}}`Zeigt den Produktnamen für die aktuelle Zeile an.

- `{{currentRow["Supplier Name"]}}`Zeigt den Lieferantennamen für die aktuelle Zeile an, in der sich die Spaltenüberschrift befindet`Supplier Name`.
- `{{currentRow.orderDate}}`Zeigt das Bestelldatum für die aktuelle Zeile an.

Beispiele: Verweisen auf die ausgewählte Zeile

`{{ui.table1.selectedRow["columnName"]}}`Dieser Ausdruck ermöglicht es Ihnen, auf den Wert einer bestimmten Spalte für die aktuell ausgewählte Zeile in der Tabelle mit der ID `table1` zu verweisen. `table1`Ersetzen Sie ihn durch die tatsächliche ID Ihrer Tabellenkomponente und `columnName` durch den Namen der Spalte, auf die Sie verweisen möchten.

Beispiele:

- `{{ui.ordersTable.selectedRow["totalAmount"]}}`Zeigt den Gesamtbetrag für die aktuell ausgewählte Zeile in der Tabelle mit der ID an`ordersTable`.
- `{{ui.customersTable.selectedRow["email"]}}`Zeigt die E-Mail-Adresse für die aktuell ausgewählte Zeile in der Tabelle mit der ID an`customersTable`.
- `{{ui.employeesTable.selectedRow["department"]}}`Zeigt die Abteilung für die aktuell ausgewählte Zeile in der Tabelle mit der ID an`employeesTable`.

Beispiele: Benutzerdefinierte Spalten erstellen

Sie können einer Tabelle benutzerdefinierte Spalten hinzufügen, die auf Daten basieren, die von der zugrunde liegenden Datenaktion, Automatisierung oder dem Ausdruck zurückgegeben wurden. Sie können vorhandene Spaltenwerte und JavaScript Ausdrücke verwenden, um neue Spalten zu erstellen.

Beispiele:

- `{{currentRow.quantity * currentRow.unitPrice}}`Erstellt eine neue Spalte, in der der Gesamtpreis angezeigt wird, indem die Spalten Menge und Einzelpreis multipliziert werden.
- `{{new Date(currentRow.orderDate).toLocaleDateString()}}`Erstellt eine neue Spalte, in der das Bestelldatum in einem besser lesbaren Format angezeigt wird.
- `{{currentRow.firstName + ' ' + currentRow.lastName + ' (' + currentRow.email + ')' }}`Erstellt eine neue Spalte, in der der vollständige Name und die E-Mail-Adresse für jede Zeile angezeigt werden.

Beispiele: Anpassen der Werte für die Spaltenanzeige:

Sie können den Anzeigewert eines Felds innerhalb einer Tabellenspalte anpassen, indem Sie das `Value` Feld der Spaltenzuordnung festlegen. Auf diese Weise können Sie benutzerdefinierte Formatierungen oder Transformationen auf die angezeigten Daten anwenden.

Beispiele:

- `{{ currentRow.rating >= 4 ? '##'.repeat(currentRow.rating) : currentRow.rating }}`Zeigt Stern-Emojis basierend auf dem Bewertungswert für jede Zeile an.
- `{{ currentRow.category.toLowerCase().replace(/\b\w/g, c => c.toUpperCase()) }}`Zeigt den Kategoriewert an, wobei jedes Wort für jede Zeile groß geschrieben wird.
- `{{ currentRow.status === 'Active' ? '# Active' : '# Inactive' }}`: Zeigt einen farbigen Kreis, Emoji und Text an, der auf dem Statuswert für jede Zeile basiert.

Tastenaktionen auf Zeilenebene

`{{currentRow.columnName}}`Oder `{{currentRow["Column Name"]}}` Sie können diese Ausdrücke verwenden, um den Kontext der referenzierten Zeile innerhalb einer Aktion auf Zeilenebene zu übergeben, z. B. das Navigieren zu einer anderen Seite mit den Daten der ausgewählten Zeile oder das Auslösen einer Automatisierung mit den Daten der Zeile.

Beispiele:

- Wenn Sie in der Spalte für die Zeilenaktion eine Bearbeitungsschaltfläche haben, können Sie diese `{{currentRow.orderId}}` als Parameter übergeben, um zu einer Seite zur Bearbeitung von Bestellungen mit der ID der ausgewählten Bestellung zu navigieren.
- Wenn Sie in der Spalte für die Zeilenaktion eine Schaltfläche zum Löschen haben, können Sie `{{currentRow.customerName}}` zu einer Automatisierung wechseln, die dem Kunden eine Bestätigungs-E-Mail sendet, bevor er seine Bestellung löscht.
- Wenn Sie in der Spalte für die Zeilenaktion eine Schaltfläche „Details anzeigen“ haben, können Sie `{{currentRow.employeeId}}` zu einer Automatisierung übergehen, die den Mitarbeiter protokolliert, der die Bestelldetails angesehen hat.

Durch die Nutzung dieser Ausdrucksfelder und Aktionsfunktionen auf Zeilenebene können Sie hochgradig angepasste und interaktive Tabellen erstellen, in denen Daten gemäß Ihren spezifischen

Anforderungen angezeigt und bearbeitet werden. Darüber hinaus können Sie Aktionen auf Zeilenebene mit anderen Komponenten oder Automatisierungen innerhalb Ihrer Anwendung verbinden und so einen nahtlosen Datenfluss und eine nahtlose Funktionalität ermöglichen.

Detail

Die Detailkomponente dient dazu, detaillierte Informationen zu einem bestimmten Datensatz oder Element anzuzeigen. Sie bietet einen speziellen Bereich für die Präsentation umfassender Daten, die sich auf eine einzelne Entität oder Zeile beziehen, und eignet sich somit ideal für die Präsentation detaillierter Details oder zur Erleichterung von Dateneingabe- und Bearbeitungsaufgaben.

Eigenschaften im Detail

Die Detailkomponente weist mehrere gemeinsame Eigenschaften mit anderen Komponenten auf, z. B. `NameSource`, und`Actions`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Die Detailkomponente verfügt auch über spezifische Eigenschaften und KonfigurationsoptionenFields, darunterLayout, undExpressions.

Layout

Im Abschnitt Layout können Sie die Anordnung und Darstellung der Felder innerhalb der Detailkomponente anpassen. Sie können Optionen konfigurieren wie:

- Anzahl der Spalten: Geben Sie die Anzahl der Spalten an, in denen die Felder angezeigt werden sollen.
- Reihenfolge der Felder: Ziehen Sie Felder per Drag-and-Drop, um ihr Aussehen neu zu ordnen.
- Abstand und Ausrichtung: Passen Sie den Abstand und die Ausrichtung der Felder innerhalb der Komponente an.

Ausdrücke und Beispiele

Die Detailkomponente bietet verschiedene Ausdrucksfelder, mit denen Sie Daten innerhalb der Komponente dynamisch referenzieren und anzeigen können. Mit diesen Ausdrücken können Sie benutzerdefinierte und interaktive Detailkomponenten erstellen, die sich nahtlos mit den Daten und der Logik Ihrer Anwendung verbinden.

Beispiel: Referenzieren von Daten

`{{ui.details.data[0]?.["colName"]}}`: Mit diesem Ausdruck können Sie auf den Wert der Spalte mit dem Namen „colName“ für das erste Element (Index 0) im Datenarray verweisen, das mit der Detail-Komponente mit der ID „details“ verbunden ist. Ersetzen Sie „colName“ durch den tatsächlichen Namen der Spalte, auf die Sie verweisen möchten. Der folgende Ausdruck zeigt beispielsweise den Wert der Spalte „customerName“ für das erste Element im Datenarray an, das mit der Komponente „Details“ verbunden ist:

```
{{ui.details.data[0]?.["customerName"]}}
```

> **Note**
>
> Dieser Ausdruck ist nützlich, wenn sich die Detailkomponente auf derselben Seite befindet wie die Tabelle, auf die verwiesen wird, und Sie Daten aus der ersten Zeile der Tabelle in der Detailkomponente anzeigen möchten.

Beispiel: Bedingtes Rendern

`{{ui.`*`table1`*`.selectedRow["`*`colName`*`"]}}`: Dieser Ausdruck gibt „true“ zurück, wenn die ausgewählte Zeile in der Tabelle mit der ID *`table1`* Daten für die angegebene Spalte enthält*`colName`*. Er kann verwendet werden, um die Detailkomponente bedingt ein- oder auszublenden, je nachdem, ob die ausgewählte Zeile der Tabelle leer ist oder nicht.

Beispiel:

Sie können diesen Ausdruck in der `Visible if` Eigenschaft der Detailkomponente verwenden, um sie abhängig von der ausgewählten Zeile in der Tabelle bedingt ein- oder auszublenden.

```
{{ui.table1.selectedRow["customerName"]}}
```

Wenn dieser Ausdruck als wahr ausgewertet wird (die ausgewählte Zeile in der *`table1`* Komponente hat einen Wert für die *`customerName`* Spalte), ist die Detailkomponente sichtbar. Wenn der Ausdruck als falsch ausgewertet wird (d. h. die ausgewählte Zeile ist leer oder hat keinen Wert für „customerName“), wird die Detailkomponente ausgeblendet.

Beispiel: Bedingte Anzeige

`{{(ui.Component.value === "green" ? "#" : ui.Component.value === "yellow" ? "#" : ui.detail1.data?.[0]?.CustomerStatus)}}`: Dieser Ausdruck zeigt bedingt ein Emoji an, das auf dem Wert einer Komponente oder eines Datenfeldes basiert.

Aufschlüsselung:

- `ui.Component.value`: Verweist auf den Wert einer Komponente mit der ID`Component`.
- `=== "green"`: Prüft, ob der Wert der Komponente der Zeichenfolge „green“ entspricht.
- `? "#"`: Wenn die Bedingung wahr ist, wird der grüne Kreis als Emoji angezeigt.
- `: ui.Component.value === "yellow" ? "#"`: Wenn die erste Bedingung falsch ist, wird geprüft, ob der Wert der Komponente der Zeichenfolge „gelb“ entspricht.
- `? "#"`: Wenn die zweite Bedingung wahr ist, wird das gelbe quadratische Emoji angezeigt.
- `: ui.detail1.data?.[0]?.CustomerStatus`: Wenn beide Bedingungen falsch sind, verweist sie auf den "CustomerStatus" -Wert des ersten Elements im Datenarray, das mit der Detail-Komponente mit der ID „detail1" verbunden ist.

Dieser Ausdruck kann verwendet werden, um ein Emoji oder einen bestimmten Wert anzuzeigen, der auf dem Wert einer Komponente oder eines Datenfeldes innerhalb der Detailkomponente basiert.

Metriken

Die Metrikkomponente ist ein visuelles Element, das wichtige Kennzahlen oder Datenpunkte in einem kartenähnlichen Format anzeigt. Es wurde entwickelt, um wichtige Informationen oder Leistungsindikatoren präzise und visuell ansprechend darzustellen.

Eigenschaften von Metriken

Die Metrik-Komponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten gemeinsam`Name`, z. B.`Source`, und`Actions`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Trend

Mit der Trendfunktion der Metriken können Sie einen visuellen Indikator für die Leistung oder Veränderung der angezeigten Metrik im Laufe der Zeit anzeigen.

Trendwert

In diesem Feld können Sie den Wert oder Ausdruck angeben, der zur Bestimmung der Trendrichtung und -stärke verwendet werden soll. In der Regel wäre dies ein Wert, der die Veränderung oder Leistung über einen bestimmten Zeitraum darstellt.

Beispiel:

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue}}
```

Dieser Ausdruck ruft den month-over-month Umsatzwert aus dem ersten Element in den Daten ab, die mit den „SalesMetrics“ -Metriken verknüpft sind.

Positiver Trend

In diesem Feld können Sie einen Ausdruck eingeben, der die Bedingung für einen positiven Trend definiert. Der Ausdruck sollte als wahr oder falsch ausgewertet werden.

Beispiel:

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue > 0}}
```

Dieser Ausdruck prüft, ob der month-over-month Umsatzwert größer als 0 ist, was auf einen positiven Trend hinweist.

Negativer Trend

In diesem Feld können Sie einen Ausdruck eingeben, der die Bedingung für einen negativen Trend definiert. Der Ausdruck sollte als wahr oder falsch ausgewertet werden.

Beispiel:

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue < 0}}
```

Dieser Ausdruck prüft, ob der month-over-month Umsatzwert kleiner als 0 ist, was auf einen negativen Trend hinweist.

Farbleiste

Mit diesem Schalter können Sie die Anzeige eines farbigen Balkens aktivieren oder deaktivieren, um den Trendstatus visuell anzuzeigen.

Beispiele für Farbleisten:

Beispiel: Trend der Verkaufskennzahlen

- Trendwert: `{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue}}`
- Positiver Trend: `{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue > 0}}`
- Negativer Trend: `{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue < 0}}`
- Farbleiste: Aktiviert

Beispiel: Trend der Inventarkennzahlen

- Trendwert: `{{ui.inventoryMetrics.data?.[0]?.currentInventory - ui.inventoryMetrics.data?.[1]?.currentInventory}}`
- Positiver Trend: `{{ui.inventoryMetrics.data?.[0]?.currentInventory > ui.inventoryMetrics.data?.[1]?.currentInventory}}`
- Negativer Trend: `{{ui.inventoryMetrics.data?.[0]?.currentInventory < ui.inventoryMetrics.data?.[1]?.currentInventory}}`
- Color Bbar: Aktiviert

Beispiel: Trend zur Kundenzufriedenheit

- Trendwert: `{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore}}`
- Positiver Trend: `{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore >= 8}}`
- Negativer Trend: `{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore < 7}}`
- Farbleiste: Aktiviert

Durch die Konfiguration dieser trendbezogenen Eigenschaften können Sie Metrikkomponenten erstellen, die eine visuelle Darstellung der Leistung oder der Veränderung der angezeigten Metrik im Laufe der Zeit bieten.

Mithilfe dieser Ausdrücke können Sie hochgradig angepasste und interaktive Metrikkomponenten erstellen, die Daten dynamisch referenzieren und anzeigen, sodass Sie wichtige Kennzahlen, Leistungsindikatoren und datengesteuerte Visualisierungen in Ihrer Anwendung präsentieren können.

Beispiele für Ausdrücke von Kennzahlen

Im Eigenschaftenfenster können Sie Ausdrücke eingeben, um den Titel, den Primärwert, den Sekundärwert und die Wertbezeichnung anzuzeigen, um einen Wert dynamisch anzuzeigen.

Beispiel: Primärwert referenzieren

`{{ui.metric1.primaryValue}}`: Mit diesem Ausdruck können Sie auf den Primärwert der Metrikkomponente mit der ID *`metric1`* anderer Komponenten oder Ausdrücke auf derselben Seite verweisen.

Beispiel: Zeigt `{{ui.salesMetrics.primaryValue}}` den Primärwert der *`salesMetrics`* Metrics-Komponente an.

Beispiel: Referenzierung eines sekundären Werts

`{{ui.metric1.secondaryValue}}`: Mit diesem Ausdruck können Sie auf den Sekundärwert der Metrikkomponente mit der ID *`metric1`* anderer Komponenten oder Ausdrücke auf derselben Seite verweisen.

Beispiel: Zeigt `{{ui.revenueMetrics.secondaryValue}}` den Sekundärwert der *`revenueMetrics`* Metrics-Komponente an.

Beispiel: Daten referenzieren

`{{ui.metric1.data}}`: Dieser Ausdruck ermöglicht es Ihnen, die Daten der Metrik-Komponente mit der ID *`metric1`* anderer Komponenten oder Ausdrücke auf derselben Seite zu referenzieren.

Beispiel: verweist auf `{{ui.kpiMetrics.data}}` die Daten, die mit der *`kpiMetrics`* Metrics-Komponente verbunden sind.

Beispiel: Anzeige bestimmter Datenwerte:

`{{ui.metric1.data?.[0]?.id}}`: Dieser Ausdruck ist ein Beispiel dafür, wie eine bestimmte Information aus den Daten angezeigt wird, die mit der Metrik-Komponente mit der ID verknüpft sind*`metric1`*. Dies ist nützlich, wenn Sie eine bestimmte Eigenschaft des ersten Elements in den Daten anzeigen möchten.

Aufschlüsselung:

- `ui.`*`metric1`*: Verweist auf die Metrik-Komponente mit der ID*`metric1`*.
- `data`: Bezieht sich auf die Informationen oder den Datensatz, der mit dieser Komponente verbunden ist.

- `?.[0]`: Bedeutet das erste Element oder den ersten Eintrag in diesem Datensatz.
- `?.id`: Zeigt den *`id`* Wert oder die Kennung des ersten Elements oder Eintrags an.

Beispiel: Zeigt `{{ui.`*`orderMetrics`*`.data?.[0]?.`*`orderId`*`}}` den *`orderId`* Wert des ersten Elements in den Daten an, die mit der *`orderMetrics`* Metrik-Komponente verbunden sind.

Beispiel: Datenlänge anzeigen

`{{ui.`*`metric1`*`.data?.length}}`: Dieser Ausdruck veranschaulicht, wie die Länge (Anzahl der Elemente) in den Daten angezeigt wird, die mit der Metrics-Komponente mit der ID verbunden sind*`metric1`*. Dies ist nützlich, wenn Sie die Anzahl der Elemente in den Daten anzeigen möchten.

Aufschlüsselung:

- `ui.`*`metric1`*`.data`: Verweist auf den Datensatz, der mit der Komponente verbunden ist.
- `?.length`: Greift auf die Gesamtzahl oder Anzahl der Elemente oder Einträge in diesem Datensatz zu.

Beispiel: Zeigt `{{ui.`*`productMetrics`*`.data?.length}}` die Anzahl der Elemente in den Daten an, die mit der *`productMetrics`* Metrik-Komponente verbunden sind.

Repeater

Die Repeater-Komponente ist eine dynamische Komponente, mit der Sie eine Sammlung von Elementen auf der Grundlage einer bereitgestellten Datenquelle generieren und anzeigen können. Sie wurde entwickelt, um die Erstellung von Listen, Rastern oder sich wiederholenden Mustern innerhalb der Benutzeroberfläche Ihrer Anwendung zu erleichtern. Zu den Anwendungsbeispielen gehören:

- Anzeige einer Karte für jeden Benutzer in einem Konto
- Es wird eine Liste von Produkten angezeigt, die Bilder und eine Schaltfläche zum Hinzufügen zum Warenkorb enthalten
- Es wird eine Liste der Dateien angezeigt, auf die der Benutzer zugreifen kann

Die Repeater-Komponente unterscheidet sich von der Table-Komponente mit umfangreichem Inhalt. Eine Tabellenkomponente hat ein striktes Zeilen- und Spaltenformat. Der Repeater kann Ihre Daten flexibler anzeigen.

Eigenschaften des Repeaters

Die Repeater-Komponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten wie `NameSource`, und. `Actions` Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Zusätzlich zu den allgemeinen Eigenschaften verfügt die Repeater-Komponente über die folgenden zusätzlichen Eigenschaften und Konfigurationsoptionen.

Vorlage für Artikel

Die Artikelvorlage ist ein Container, in dem Sie die Struktur und die Komponenten definieren können, die für jedes Element in der Datenquelle wiederholt werden. Sie können andere Komponenten wie Text, Bild, Schaltfläche oder jede andere Komponente, die Sie zur Darstellung der einzelnen Elemente benötigen, per Drag & Drop in diesen Container ziehen.

In der Artikelvorlage können Sie mithilfe von Ausdrücken im Format auf Eigenschaften oder Werte des aktuellen Elements verweisen`{{currentItem.propertyName}}`.

Wenn Ihre Datenquelle beispielsweise eine `itemName` Eigenschaft enthält, können Sie `{{currentItem.itemName}}` sie verwenden, um die Elementnamen des aktuellen Elements anzuzeigen.

Layout

Im Abschnitt Layout können Sie die Anordnung der wiederholten Elemente innerhalb der Repeater-Komponente konfigurieren.

Ausrichtung

- Liste: Ordnet die wiederholten Elemente vertikal in einer einzigen Spalte an.
- Raster: Ordnet die wiederholten Elemente in einem Rasterlayout mit mehreren Spalten an.

Zeilen pro Seite

Geben Sie die Anzahl der Zeilen an, die pro Seite im Listenlayout angezeigt werden sollen. Die Seitennummerierung ist für Elemente vorgesehen, bei denen die angegebene Zeilenanzahl überschritten wird.

Spalten und Zeilen pro Seite (Raster)

- Spalten: Geben Sie die Anzahl der Spalten im Rasterlayout an.

- Zeilen pro Seite: Geben Sie die Anzahl der Zeilen an, die pro Seite im Rasterlayout angezeigt werden sollen. Die Seitennummerierung ist für Elemente vorgesehen, die die angegebenen Rastermaße überschreiten.

Ausdrücke und Beispiele

Die Repeater-Komponente bietet verschiedene Ausdrucksfelder, mit denen Sie Daten innerhalb der Komponente dynamisch referenzieren und anzeigen können. Mit diesen Ausdrücken können Sie benutzerdefinierte und interaktive Repeater-Komponenten erstellen, die sich nahtlos mit den Daten und der Logik Ihrer Anwendung verbinden.

Beispiel: Elemente referenzieren

- `{{currentItem.propertyName}}`: Referenzeigenschaften oder Werte des aktuellen Elements in der Artikelvorlage.
- `{{ui.repeaterID[index]}}`: Verweisen Sie anhand seines Indexes auf ein bestimmtes Element in der Repeater-Komponente.

Beispiel: Rendern einer Produktliste

- Quelle: Wählen Sie die *`Products`* Entität als Datenquelle aus.
- Artikelvorlage: Fügen Sie eine Container-Komponente mit einer Textkomponente hinzu, um den Produktnamen (`{{currentItem.productName}}`) anzuzeigen, und eine Bildkomponente, um das Produktbild anzuzeigen (`{{currentItem.productImageUrl}}`).
- Layout: Stellen Sie den Wert `Orientation` auf ein `List` und passen Sie ihn `Rows per Page` wie gewünscht an.

Beispiel: Generieren eines Rasters mit Benutzer-Avataren

- Quelle: Verwenden Sie einen Ausdruck, um ein Array von Benutzerdaten zu generieren (z. B.`[{name: 'John', avatarUrl: '...'}, {...}, {...}]`).
- Elementvorlage: Fügen Sie eine Image-Komponente hinzu und legen Sie ihre `Source` Eigenschaft auf fest`{{currentItem.avatarUrl}}`.
- Layout: Stellen Sie den Wert `Orientation` auf ein`Grid`, geben Sie die Anzahl von `Columns` und `Rows per Page` an und passen Sie den Wert `Space Between` und nach `Padding` Bedarf an.

Mithilfe der `Repeater` Komponente können Sie dynamische und datengesteuerte Benutzeroberflächen erstellen, wodurch das Rendern von Elementsammlungen rationalisiert und die Notwendigkeit manueller Wiederholungen oder hartes Codieren reduziert wird.

Formular

Die Formularkomponente wurde entwickelt, um Benutzereingaben zu erfassen und Dateneingabeaufgaben in Ihrer Anwendung zu erleichtern. Sie bietet ein strukturiertes Layout für die Anzeige von Eingabefeldern, Dropdowns, Kontrollkästchen und anderen Formularsteuerelementen, sodass Benutzer Daten nahtlos eingeben oder ändern können. Sie können andere Komponenten innerhalb einer Formularkomponente, z. B. einer Tabelle, verschachteln.

Eigenschaften von Formularen

Die Formularkomponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten gemeinsam, z. B. `NameSource`, und`Actions`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Formular generieren

Die Funktion „Formular generieren“ erleichtert das schnelle Erstellen von Formularfeldern, indem sie automatisch auf der Grundlage einer ausgewählten Datenquelle ausgefüllt werden. Dies kann Zeit und Mühe bei der Erstellung von Formularen sparen, die eine große Anzahl von Feldern anzeigen müssen.

So verwenden Sie die Funktion „Formular generieren“:

1. Suchen Sie in den Eigenschaften der Formularkomponente nach dem Abschnitt Formular generieren.
2. Wählen Sie die Datenquelle aus, die Sie zum Generieren der Formularfelder verwenden möchten. Dies kann eine Entität, ein Workflow oder eine andere in Ihrer Anwendung verfügbare Datenquelle sein.
3. Die Formularfelder werden automatisch auf der Grundlage der ausgewählten Datenquelle generiert, einschließlich der Feldbezeichnungen, Typen und Datenzuordnungen.
4. Überprüfen Sie die generierten Felder und nehmen Sie alle erforderlichen Anpassungen vor, z. B. das Hinzufügen von Validierungsregeln oder das Ändern der Feldreihenfolge.
5. Wenn Sie mit der Formularkonfiguration zufrieden sind, wählen Sie Senden aus, um die generierten Felder auf Ihre Formularkomponente anzuwenden.

Die Funktion „Formular generieren“ ist besonders nützlich, wenn Sie ein klar definiertes Datenmodell oder eine Reihe von Entitäten in Ihrer Anwendung haben, für die Sie Benutzereingaben erfassen müssen. Durch die automatische Generierung der Formularfelder können Sie Zeit sparen und die Konsistenz aller Formulare Ihrer Anwendung sicherstellen.

Nachdem Sie die Funktion „Formular generieren“ verwendet haben, können Sie das Layout, die Aktionen und Ausdrücke für die Formularkomponente weiter an Ihre spezifischen Anforderungen anpassen.

Ausdrücke und Beispiele

Wie bei anderen Komponenten können Sie Ausdrücke verwenden, um Daten innerhalb der Form-Komponente zu referenzieren und anzuzeigen. Zum Beispiel:

- `{{ui.userForm.data.email}}`: Verweist auf den Wert des `email` Felds aus der Datenquelle, die mit der Form-Komponente mit der ID verbunden ist`userForm`.

> **Note**
>
> Weitere Ausdrucksbeispiele Allgemeine Eigenschaften von Komponenten für die allgemeinen Eigenschaften finden Sie unter.

Durch die Konfiguration dieser Eigenschaften und die Nutzung von Ausdrücken können Sie benutzerdefinierte und interaktive Formularkomponenten erstellen, die sich nahtlos in die Datenquellen und die Logik Ihrer Anwendung integrieren lassen. Diese Komponenten können verwendet werden, um Benutzereingaben zu erfassen, vorab ausgefüllte Daten anzuzeigen und Aktionen auf der Grundlage von Formularübermittlungen oder Benutzerinteraktionen auszulösen.

Stepflow

Die Stepflow-Komponente wurde entwickelt, um Benutzer durch mehrstufige Prozesse oder Workflows innerhalb Ihrer Anwendung zu führen. Sie bietet eine strukturierte und intuitive Oberfläche zur Darstellung einer Abfolge von Schritten mit jeweils eigenen Eingaben, Validierungen und Aktionen.

Die Stepflow-Komponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten wie`Name`, und`Source`. `Actions` Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Die Stepflow-Komponente verfügt über zusätzliche Eigenschaften und Konfigurationsoptionen wie `Step Navigation`Validation, und`Expressions`.

KI-Komponenten

KI der Generation

Die Gen AI-Komponente ist ein Gruppierungscontainer, der verwendet wird, um Komponenten und ihre zugehörige Logik zu gruppieren, um sie mithilfe des Chats im Anwendungsstudio einfach mit KI zu bearbeiten. Wenn Sie den Chat verwenden, um Komponenten zu erstellen, werden diese in einem Gen AI-Container gruppiert. Informationen zur Bearbeitung oder Verwendung dieser Komponente finden Sie unterIhre App erstellen oder bearbeiten.

Text- und Zahlenkomponenten

Texteingabe

Die Texteingabekomponente ermöglicht es Benutzern, Textdaten in Ihrer Anwendung einzugeben und zu senden. Sie bietet eine einfache und intuitive Möglichkeit, Benutzereingaben wie Namen, Adressen oder andere Textinformationen zu erfassen.

- `{{ui.`*`inputTextID`*`.value}}`: Gibt den angegebenen Wert im Eingabefeld zurück.
- `{{ui.`*`inputTextID`*`.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Eingabefeld zurück.

Text

Die Textkomponente wird verwendet, um Textinformationen in Ihrer Anwendung anzuzeigen. Sie kann verwendet werden, um statischen Text, dynamische Werte oder aus Ausdrücken generierte Inhalte anzuzeigen.

Textbereich

Die Textbereichskomponente dient zur Erfassung mehrzeiliger Texteingaben von Benutzern. Sie bietet Benutzern einen größeren Eingabefeldbereich, in den sie längere Texteinträge wie Beschreibungen, Notizen oder Kommentare eingeben können.

- `{{ui.`*`textAreaID`*`.value}}`: Gibt den angegebenen Wert im Textbereich zurück.
- `{{ui.`*`textAreaID`*`.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Textbereich zurück.

Email

Die E-Mail-Komponente ist ein spezielles Eingabefeld zur Erfassung von E-Mail-Adressen von Benutzern. Es kann spezifische Validierungsregeln durchsetzen, um sicherzustellen, dass der eingegebene Wert dem richtigen E-Mail-Format entspricht.

- `{{ui.emailID.value}}`: Gibt den angegebenen Wert im E-Mail-Eingabefeld zurück.
- `{{ui.emailID.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im E-Mail-Eingabefeld zurück.

Passwort

Die Passwort-Komponente ist ein Eingabefeld, das speziell für Benutzer zur Eingabe vertraulicher Informationen wie Passwörter oder PIN-Codes konzipiert wurde. Es maskiert die eingegebenen Zeichen, um Datenschutz und Sicherheit zu gewährleisten.

- `{{ui.passwordID.value}}`: Gibt den angegebenen Wert im Passwort-Eingabefeld zurück.
- `{{ui.passwordID.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Passworteingabefeld zurück.

Suche

Die Suchkomponente bietet Benutzern ein spezielles Eingabefeld für die Durchführung von Suchanfragen oder die Eingabe von Suchbegriffen innerhalb der ausgefüllten Daten innerhalb der Anwendung.

- `{{ui.searchID.value}}`: Gibt den angegebenen Wert im Suchfeld zurück.

Phone

Die Telefonkomponente ist ein Eingabefeld, das auf die Erfassung von Telefonnummern oder anderen Kontaktinformationen von Benutzern zugeschnitten ist. Es kann spezielle Validierungsregeln und Formatierungsoptionen enthalten, um sicherzustellen, dass der eingegebene Wert dem richtigen Telefonnummernformat entspricht.

- `{{ui.phoneID.value}}`: Gibt den angegebenen Wert im Telefoneingabefeld zurück.
- `{{ui.phoneID.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Telefoneingabefeld zurück.

Anzahl

Die Zahlenkomponente ist ein Eingabefeld, das speziell für Benutzer zur Eingabe numerischer Werte konzipiert wurde. Es kann Validierungsregeln durchsetzen, um sicherzustellen, dass der eingegebene Wert eine gültige Zahl innerhalb eines bestimmten Bereichs oder Formats ist.

- `{{ui.numberID.value}}`: Gibt den angegebenen Wert im Zahleneingabefeld zurück.
- `{{ui.numberID.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Zahleneingabefeld zurück.

Währung

Die Währungskomponente ist ein spezielles Eingabefeld zur Erfassung von Geldwerten oder Beträgen. Es kann Formatierungsoptionen zur Anzeige von Währungssymbolen und Dezimaltrennzeichen sowie zur Durchsetzung spezifischer Validierungsregeln für Währungseingaben enthalten.

- `{{ui.currencyID.value}}`: Gibt den angegebenen Wert im Währungseingabefeld zurück.
- `{{ui.currencyID.isValid}}`: Gibt die Gültigkeit des angegebenen Werts im Währungseingabefeld zurück.

Detailpaar

Die Detailpaar-Komponente wird verwendet, um Schlüssel-Wert-Paare oder Paare verwandter Informationen in einem strukturierten und lesbaren Format anzuzeigen. Sie wird häufig verwendet, um Details oder Metadaten zu präsentieren, die einem bestimmten Element oder einer bestimmten Entität zugeordnet sind.

Komponenten für die Auswahl

Switch

Bei der Switch-Komponente handelt es sich um ein Steuerelement auf der Benutzeroberfläche, mit dem Benutzer zwischen zwei Status oder Optionen wechseln können, z. B. on/off, true/false, or enabled/disabled Sie bietet eine visuelle Darstellung des aktuellen Status und ermöglicht es Benutzern, ihn mit einem einzigen Klick oder Tippen zu ändern.

Gruppe wechseln

Die Switch-Gruppenkomponente ist eine Sammlung einzelner Switch-Steuerelemente, mit denen Benutzer eine oder mehrere Optionen aus einem vordefinierten Satz auswählen können. Sie bietet eine visuelle Darstellung der ausgewählten und nicht ausgewählten Optionen, sodass Benutzer die verfügbaren Optionen leichter verstehen und mit ihnen interagieren können.

Zwischen Feldern für Gruppenausdrücke wechseln

- `{{ui.switchGroupID.value}}`: Gibt ein Array von Zeichenketten zurück, die den Wert jedes Schalters enthalten, der vom App-Benutzer aktiviert wurde.

Checkbox-Gruppe

Die Checkbox-Gruppenkomponente bietet Benutzern eine Gruppe von Kontrollkästchen, sodass sie mehrere Optionen gleichzeitig auswählen können. Dies ist nützlich, wenn Sie Benutzern die Möglichkeit geben möchten, ein oder mehrere Elemente aus einer Liste von Optionen auszuwählen.

Ausdrucksfelder für Checkbox-Gruppen

- `{{ui.checkboxGroupID.value}}`: Gibt ein Array von Zeichenketten zurück, die den Wert aller vom App-Benutzer ausgewählten Kontrollkästchen enthalten.

Radio-Gruppe

Die Optionsgruppenkomponente besteht aus einer Reihe von Optionsfeldern, mit denen Benutzer eine einzelne Option aus mehreren sich gegenseitig ausschließenden Optionen auswählen können. Sie stellt sicher, dass jeweils nur eine Option ausgewählt werden kann, und bietet Benutzern so eine klare und eindeutige Möglichkeit, eine Auswahl zu treffen.

Ausdrucksfelder für Funkgruppen

Die folgenden Felder können in Ausdrücken verwendet werden.

- `{{ui.radioGroupID.value}}`: Gibt den Wert des Optionsfeldes zurück, das vom App-Benutzer ausgewählt wurde.

Einzelauswahl

Die Einzelauswahl-Komponente bietet Benutzern eine Liste von Optionen, aus denen sie ein einzelnes Element auswählen können. Sie wird häufig in Szenarien verwendet, in denen Benutzer eine Auswahl aus einem vordefinierten Satz von Optionen treffen müssen, z. B. die Auswahl einer Kategorie, eines Standorts oder einer Präferenz.

Ausdrucksfelder mit einfacher Auswahl

- `{{ui.singleSelectID.value}}`: Gibt den Wert des Listenelements zurück, das vom App-Benutzer ausgewählt wurde.

Mehrfachauswahl

Die Mehrfachauswahl-Komponente ähnelt der Einzelauswahl-Komponente, ermöglicht es Benutzern jedoch, mehrere Optionen gleichzeitig aus einer Auswahlliste auszuwählen. Sie ist nützlich, wenn Benutzer mehrere Auswahlen aus einem vordefinierten Satz von Optionen treffen müssen, z. B. die Auswahl mehrerer Stichwörter, Interessen oder Präferenzen.

Ausdrucksfelder mit Mehrfachauswahl

- `{{ui.multiSelectID.value}}`: Gibt ein Array von Zeichenketten zurück, die den Wert jedes Listenelements enthalten, das vom App-Benutzer ausgewählt wurde.

Schaltflächen und Navigationskomponenten

Das Anwendungsstudio bietet eine Vielzahl von Schaltflächen und Navigationskomponenten, mit denen Benutzer Aktionen auslösen und innerhalb Ihrer Anwendung navigieren können.

Komponenten für Schaltflächen

Die verfügbaren Schaltflächenkomponenten sind:

- Button
- Umrissene Schaltfläche
- Icon-Schaltfläche
- Schaltfläche „Text“

Diese Schaltflächenkomponenten haben die folgenden gemeinsamen Eigenschaften:

Inhalt

- Tastenbezeichnung: Der Text, der auf der Schaltfläche angezeigt werden soll.

Typ

- Taste: Eine Standardtaste.
- Umrissen: Eine Schaltfläche mit einem konturierten Stil.
- Symbol: Eine Schaltfläche mit einem Symbol.
- Text: Eine Schaltfläche, die nur Text enthält.

Größe

Die Größe der Schaltfläche. Mögliche Werte sind `Small`, `Medium` und `Large`.

Symbol

Sie können aus einer Vielzahl von Symbolen wählen, die auf der Schaltfläche angezeigt werden sollen, darunter:

- Umschlag geschlossen
- Glocke
- Person
- Hamburger-Menü
- Suche
- Informationen eingekreist
- Ausrüstung
- Chevron links
- Chevron Rechts
- Horizontale Punkte
- Papierkorb
- Edit (Bearbeiten)
- Check
- Schließen
- Startseite

- Plus

Auslöser

Wenn auf die Schaltfläche geklickt wird, können Sie eine oder mehrere Aktionen konfigurieren, die ausgelöst werden sollen. Die verfügbaren Aktionstypen sind:

- Basic
  - Komponentenaktion ausführen: Führt eine bestimmte Aktion innerhalb einer Komponente aus.
  - Navigieren: Navigiert zu einer anderen Seite oder Ansicht.
  - Datenaktion aufrufen: Löst eine datenbezogene Aktion aus, z. B. das Erstellen, Aktualisieren oder Löschen eines Datensatzes.
- Advanced
  - JavaScript: Führt benutzerdefinierten JavaScript Code aus.
  - Automatisierung aufrufen: Startet eine bestehende Automatisierung oder einen bestehenden Workflow.

JavaScript Eigenschaften der Aktionsschaltfläche

Wählen Sie den `JavaScript` Aktionstyp aus, um benutzerdefinierten JavaScript Code auszuführen, wenn auf die Schaltfläche geklickt wird.

Quellcode

In das `Source code` Feld können Sie Ihren JavaScript Ausdruck oder Ihre Funktion eingeben. Zum Beispiel:

```
return "Hello World";
```

Dadurch wird einfach die Zeichenfolge zurückgegeben`Hello World`, wenn auf die Schaltfläche geklickt wird.

Bedingung: Führen Sie aus, wenn

Sie können auch einen booleschen Ausdruck angeben, der bestimmt, ob die JavaScript Aktion ausgeführt werden soll oder nicht. Für dieses Tag gilt folgende Syntax:

```
{{ui.textinput1.value !== ""}}
```

In diesem Beispiel wird die JavaScript Aktion nur ausgeführt, wenn der Wert der `textinput1` Komponente keine leere Zeichenfolge ist.

Mithilfe dieser erweiterten Triggeroptionen können Sie hochgradig angepasste Tastenverhaltensweisen erstellen, die sich direkt in die Logik und die Daten Ihrer Anwendung integrieren lassen. Auf diese Weise können Sie die integrierte Funktionalität der Schaltflächen erweitern und die Benutzererfahrung an Ihre spezifischen Anforderungen anpassen.

> **Note**
>
> Testen Sie Ihre JavaScript Aktionen immer gründlich, um sicherzustellen, dass sie wie erwartet funktionieren.

Hyperlink

Die Hyperlink-Komponente bietet einen anklickbaren Link für die Navigation zu externen URLs oder internen Anwendungsrouten.

Eigenschaften von Hyperlinks

Inhalt

- Hyperlink-Bezeichnung: Der Text, der als Hyperlink-Label angezeigt werden soll.

URL

Die Ziel-URL für den Hyperlink, bei der es sich um eine externe Website oder eine interne Anwendungsroute handeln kann.

Auslöser

Wenn auf den Hyperlink geklickt wird, können Sie konfigurieren, dass eine oder mehrere Aktionen ausgelöst werden. Die verfügbaren Aktionstypen sind dieselben wie für die Schaltflächenkomponenten.

Komponenten für Datum und Uhrzeit

Datum

Die Datumskomponente ermöglicht es Benutzern, Daten auszuwählen und einzugeben.

Die Datumskomponente hat mehrere gemeinsame Eigenschaften mit anderen Komponenten gemeinsam, z. B. `NameSource`, und`Validation`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Zusätzlich zu den allgemeinen Eigenschaften hat die Date-Komponente die folgenden spezifischen Eigenschaften:

Eigenschaften des Datums

Format

- YYYY/MM/DD, DD/MM/YYYY,, YYYY/MM/DDYYYY/DD/MM, MM/DD, TT/MM: Das Format, in dem das Datum angezeigt werden soll.

Wert

- YYYY-MM-DD: Das Format, in dem der Datumswert intern gespeichert wird.

Minimales Datum

- YYYY-MM-DD: Das Mindestdatum, das ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Maximales Datum

- YYYY-MM-DD: Das maximale Datum, das ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Typ des Kalenders

- 1 Monat, 2 Monate: Die Art der anzuzeigenden Kalenderbenutzeroberfläche.

Deaktivierte Daten

- Quelle: Die Datenquelle für die Daten, die deaktiviert werden sollten. Zum Beispiel: Keine, Ausdruck.
- Deaktivierte Daten: Ein Ausdruck, der bestimmt, welche Daten deaktiviert werden sollen, z. B.:
  - `{{currentRow.column}}`: Deaktiviert Daten, die dem entsprechen, was dieser Ausdruck ergibt.
  - `{{new Date(currentRow.dateColumn) < new Date("2023-01-01")}}`: Deaktiviert Daten vor dem 1. Januar 2023
  - `{{new Date(currentRow.dateColumn).getDay() === 0 || new Date(currentRow.dateColumn).getDay() === 6}}`: Deaktiviert Wochenenden.

Behavior

- Sichtbar wenn: Ein Ausdruck, der die Sichtbarkeit der Date-Komponente bestimmt.
- Deaktivieren wenn: Ein Ausdruck, der bestimmt, ob die Datumskomponente deaktiviert werden soll.

Validierung

Im Abschnitt Validierung können Sie zusätzliche Regeln und Einschränkungen für die Datumseingabe definieren. Durch die Konfiguration dieser Validierungsregeln können Sie sicherstellen, dass die von Benutzern eingegebenen Datumswerte den spezifischen Anforderungen Ihrer Anwendung entsprechen. Sie können die folgenden Arten von Validierungen hinzufügen:

- Erforderlich: Dieser Schalter stellt sicher, dass der Benutzer vor dem Absenden des Formulars einen Datumswert eingeben muss.
- Benutzerdefiniert: Sie können mithilfe von JavaScript Ausdrücken benutzerdefinierte Validierungsregeln erstellen. Zum Beispiel:

  ```
  {{new Date(ui.dateInput.value) < new Date("2023-01-01")}}
  ```

  Dieser Ausdruck überprüft, ob das eingegebene Datum vor dem 1. Januar 2023 liegt. Wenn die Bedingung erfüllt ist, schlägt die Überprüfung fehl.

  Sie können auch eine benutzerdefinierte Bestätigungsnachricht angeben, die angezeigt wird, wenn die Überprüfung nicht erfüllt ist:

```
"Validation not met. The date must be on or after January 1, 2023."
```

Durch die Konfiguration dieser Validierungsregeln können Sie sicherstellen, dass die von Benutzern eingegebenen Datumswerte den spezifischen Anforderungen Ihrer Anwendung entsprechen.

Ausdrücke und Beispiele

Die Date-Komponente stellt das folgende Ausdrucksfeld bereit:

- `{{ui.dateID.value}}`: Gibt den vom Benutzer eingegebenen Datumswert im Format zurück`YYYY-MM-DD`.

Zeit

Die Zeitkomponente ermöglicht es Benutzern, Zeitwerte auszuwählen und einzugeben. Durch die Konfiguration der verschiedenen Eigenschaften der Zeitkomponente können Sie Zeiteingabefelder erstellen, die den spezifischen Anforderungen Ihrer Anwendung entsprechen, z. B. die Einschränkung des auswählbaren Zeitbereichs, die Deaktivierung bestimmter Zeiten und die Steuerung der Sichtbarkeit und Interaktivität der Komponente.

Eigenschaften der Uhrzeit

Die Zeitkomponente weist mehrere gemeinsame Eigenschaften mit anderen Komponenten auf, z. B. `NameSource`, und`Validation`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Zusätzlich zu den allgemeinen Eigenschaften hat die Time-Komponente die folgenden spezifischen Eigenschaften:

Zeitintervalle

- 5 Minuten, 10 Minuten, 15 Minuten, 20 Minuten, 25 Minuten, 30 Minuten, 60 Minuten: Die Intervalle, die für die Auswahl der Uhrzeit zur Verfügung stehen.

Wert

- HH:MM AA: Das Format, in dem der Zeitwert intern gespeichert wird.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `HH:MM AA`

Placeholder

- Kalendereinstellungen: Der Platzhaltertext, der angezeigt wird, wenn das Zeitfeld leer ist.

Minimale Uhrzeit

- HH:MM AA: Die Mindestzeit, die ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `HH:MM AA`

Max. Zeit

- HH:MM AA: Die maximale Zeit, die ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `HH:MM AA`

Deaktivierte Zeiten

- Quelle: Die Datenquelle für die Zeiten, die deaktiviert werden sollen (z. B. None, Expression).
- Deaktivierte Zeiten: Ein Ausdruck, der bestimmt, welche Zeiten deaktiviert werden sollen, z. `{{currentRow.column}}` B.

Konfiguration für deaktivierte Zeiten

Im Bereich Deaktivierte Zeiten können Sie angeben, welche Zeitwerte nicht zur Auswahl stehen sollen.

Quelle

- Keine: Es sind keine Zeiten deaktiviert.
- Ausdruck: Sie können einen JavaScript Ausdruck verwenden, um zu bestimmen, welche Zeiten deaktiviert werden sollen, z. `{{currentRow.column}}` B.

Beispielausdruck:

```
{{currentRow.column === "Lunch Break"}}
```

Dieser Ausdruck würde jedes Mal deaktivieren, wenn die Spalte „Lunch Break“ für die aktuelle Zeile den Wert true hat.

Durch die Konfiguration dieser Validierungsregeln und die Deaktivierung von Zeitausdrücken können Sie sicherstellen, dass die von den Benutzern eingegebenen Zeitwerte den spezifischen Anforderungen Ihrer Anwendung entsprechen.

Behavior

- Sichtbar wenn: Ein Ausdruck, der die Sichtbarkeit der Zeitkomponente bestimmt.
- Deaktivieren wenn: Ein Ausdruck, der bestimmt, ob die Zeitkomponente deaktiviert werden soll.

Validierung

- Erforderlich: Ein Schalter, der sicherstellt, dass der Benutzer vor dem Absenden des Formulars einen Zeitwert eingeben muss.
- Benutzerdefiniert: Ermöglicht das Erstellen benutzerdefinierter Validierungsregeln mithilfe von JavaScript Ausdrücken.

  Benutzerdefinierte Bestätigungsnachricht: Die Meldung, die angezeigt werden soll, wenn die benutzerdefinierte Überprüfung nicht erfüllt ist.

Zum Beispiel:

```
{{ui.timeInput.value === "09:00 AM" || ui.timeInput.value === "09:30 AM"}}
```

Dieser Ausdruck überprüft, ob die eingegebene Uhrzeit 9:00 Uhr oder 9:30 Uhr ist. Wenn die Bedingung erfüllt ist, schlägt die Überprüfung fehl.

Sie können auch eine benutzerdefinierte Bestätigungsnachricht angeben, die angezeigt wird, wenn die Überprüfung nicht erfüllt ist:

```
Validation not met. The time must be 9:00 AM or 9:30 AM.
```

Ausdrücke und Beispiele

Die Time-Komponente stellt das folgende Ausdrucksfeld bereit:

- `{{ui.timeID.value}}`: Gibt den vom Benutzer eingegebenen Zeitwert im Format HH:MM AA zurück.

Beispiel: Zeitwert

- `{{ui.timeID.value}}`: Gibt den vom Benutzer eingegebenen Zeitwert im Format zurück`HH:MM` AA.

Beispiel: Zeitvergleich

- `{{ui.timeInput.value > "10:00 AM"}}`: Prüft, ob der Zeitwert größer als 10:00 Uhr ist.
- `{{ui.timeInput.value < "05:00 pM"}}`: Prüft, ob der Zeitwert unter 17:00 Uhr liegt.

Datumsbereich

Die Komponente „Datumsbereich“ ermöglicht es Benutzern, einen Datumsbereich auszuwählen und einzugeben. Durch die Konfiguration der verschiedenen Eigenschaften der Komponente „Datumsbereich“ können Sie Eingabefelder für den Datumsbereich erstellen, die den spezifischen Anforderungen Ihrer Anwendung entsprechen, z. B. die Einschränkung des auswählbaren Datumsbereichs, die Deaktivierung bestimmter Daten und die Steuerung der Sichtbarkeit und Interaktivität der Komponente.

Eigenschaften des Datumsbereichs

Die Komponente „Datumsbereich“ hat mehrere gemeinsame Eigenschaften mit anderen Komponenten gemeinsam`Name`, z. B.`Source`, und`Validation`. Weitere Informationen zu diesen Eigenschaften finden Sie unterAllgemeine Eigenschaften von Komponenten.

Zusätzlich zu den allgemeinen Eigenschaften hat die Komponente Date Range die folgenden spezifischen Eigenschaften:

Format

- MM/DD/YYYY: Das Format, in dem der Datumsbereich angezeigt werden soll.

Startdatum

- YYYY-MM-DD: Das Mindestdatum, das als Anfang des Bereichs ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Enddatum

- YYYY-MM-DD: Das maximale Datum, das als Ende des Bereichs ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Placeholder

- Kalendereinstellungen: Der Platzhaltertext, der angezeigt wird, wenn das Datumsbereichsfeld leer ist.

Minimales Datum

- YYYY-MM-DD: Das Mindestdatum, das ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Maximales Datum

- YYYY-MM-DD: Das maximale Datum, das ausgewählt werden kann.

> **Note**
>
> Dieser Wert muss dem Format von entsprechen. `YYYY-MM-DD`

Typ des Kalenders

- 1 Monat: Die Art der anzuzeigenden Kalenderbenutzeroberfläche. Zum Beispiel ein einzelner Monat.
- 2 Monate: Der Typ der anzuzeigenden Kalenderbenutzeroberfläche. Zum Beispiel zwei Monate.

Obligatorische Tage ausgewählt

- 0: Die Anzahl der obligatorischen Tage, die innerhalb des Datumsbereichs ausgewählt werden müssen.

Deaktivierte Daten

- Quelle: Die Datenquelle für die Daten, die deaktiviert werden sollen (z. B. Keine, Ausdruck, Entität oder Automatisierung).
- Deaktivierte Daten: Ein Ausdruck, der bestimmt, welche Daten deaktiviert werden sollen, z. `{{currentRow.column}}` B.

Validierung

Im Abschnitt Validierung können Sie zusätzliche Regeln und Einschränkungen für die Eingabe des Datumsbereichs definieren.

Ausdrücke und Beispiele

Die Komponente „Datumsbereich“ stellt die folgenden Ausdrucksfelder bereit:

- `{{ui.dateRangeID.startDate}}`: Gibt das Startdatum des ausgewählten Bereichs im Format zurück`YYYY-MM-DD`.
- `{{ui.dateRangeID.endDate}}`: Gibt das Enddatum des ausgewählten Bereichs im Format zurück`YYYY-MM-DD`.

Beispiel: Berechnung der Datumsdifferenz

- `{(new Date(ui.dateRangeID.endDate) - new Date(ui.dateRangeID.startDate)) / (1000 * 60 * 60 * 24)}}`Berechnet die Anzahl der Tage zwischen dem Start- und dem Enddatum.

Beispiel: Bedingte Sichtbarkeit basierend auf dem Datumsbereich

- `{{new Date(ui.dateRangeID.startDate) < new Date("2023-01-01") || new Date(ui.dateRangeID.endDate) > new Date("2023-12-31")}}`Prüft, ob der ausgewählte Datumsbereich außerhalb des Jahres 2023 liegt.

Beispiel: Deaktivierte Datumsangaben basierend auf aktuellen Zeilendaten

- `{{currentRow.isHoliday}}`Deaktiviert Daten, bei denen die Spalte „IsHoliday“ in der aktuellen Zeile den Wert true hat.
- `{{new Date(currentRow.dateColumn) < new Date("2023-01-01")}}`Deaktiviert Daten vor dem 1. Januar 2023 basierend auf der „DateColumn“ in der aktuellen Zeile.
- `{{new Date(currentRow.dateColumn).getDay() === 0 || new Date(currentRow.dateColumn).getDay() === 6}}`Deaktiviert Wochenenden basierend auf der „DateColumn“ in der aktuellen Zeile.

Benutzerdefinierte Validierung

- `{{new Date(ui.dateRangeID.startDate) > new Date(ui.dateRangeID.endDate)}}`Prüft, ob das Startdatum nach dem Enddatum liegt. Andernfalls würde die benutzerdefinierte Validierung fehlschlagen.

Medienkomponenten

Das Anwendungsstudio bietet mehrere Komponenten zum Einbetten und Anzeigen verschiedener Medientypen in Ihrer Anwendung.

iFrame einbetten

Mit der iFrame-Einbettungskomponente können Sie externe Webinhalte oder Anwendungen mithilfe eines iFrames in Ihre Anwendung einbetten.

Eigenschaften zum Einbetten von iFrames

URL

> **Note**
>
> Die Quelle der in dieser Komponente angezeigten Medien muss in den Inhaltssicherheitseinstellungen Ihrer Anwendung zulässig sein. Weitere Informationen finden Sie unter Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren.

Die URL des externen Inhalts oder der Anwendung, die Sie einbetten möchten.

Layout

- Breite: Die Breite des iFrames, angegeben als Prozentsatz (%) oder als fester Pixelwert (z. B. 300 px).
- Höhe: Die Höhe des iFrames, angegeben als Prozentsatz (%) oder als fester Pixelwert.

S3-Upload

Die S3-Upload-Komponente ermöglicht es Benutzern, Dateien in einen Amazon S3 S3-Bucket hochzuladen. Durch die Konfiguration der S3-Upload-Komponente können Sie es Benutzern ermöglichen, Dateien einfach in den Amazon S3 S3-Speicher Ihrer Anwendung hochzuladen und dann die hochgeladenen Dateiinformationen innerhalb der Logik und Benutzeroberfläche Ihrer Anwendung zu nutzen.

> **Note**
>
> Denken Sie daran, sicherzustellen, dass die erforderlichen Berechtigungen und Amazon S3 S3-Bucket-Konfigurationen vorhanden sind, um die Datei-Uploads und Speicheranforderungen Ihrer Anwendung zu unterstützen.

Eigenschaften des S3-Uploads

S3-Konfiguration

- Connector: Wählen Sie den vorkonfigurierten Amazon S3 S3-Connector aus, der für die Datei-Uploads verwendet werden soll.

- Bucket: Der Amazon S3 S3-Bucket, in den die Dateien hochgeladen werden.
- Ordner: Der Ordner innerhalb des Amazon S3 S3-Buckets, in dem die Dateien gespeichert werden.
- Dateiname: Die Benennungskonvention für die hochgeladenen Dateien.

Konfiguration für das Hochladen von Dateien

- Bezeichnung: Das Etikett oder die Anweisungen, die über dem Datei-Upload-Bereich angezeigt werden.
- Beschreibung: Zusätzliche Anweisungen oder Informationen zum Datei-Upload.
- Dateityp: Der Dateityp, der hochgeladen werden darf. Zum Beispiel: Bild, Dokument oder Video.
- Größe: Die maximale Größe der einzelnen Dateien, die hochgeladen werden können.
- Schaltflächenbezeichnung: Der Text, der auf der Schaltfläche zur Dateiauswahl angezeigt wird.
- Schaltflächenstil: Der Stil der Dateiauswahlschaltfläche. Zum Beispiel umrandet oder gefüllt.
- Größe der Schaltfläche: Die Größe der Schaltfläche zur Dateiauswahl.

Validierung

- Maximale Anzahl von Dateien: Die maximale Anzahl von Dateien, die gleichzeitig hochgeladen werden können.
- Maximale Dateigröße: Die maximal zulässige Größe für jede einzelne Datei.

Auslöser

- Bei Erfolg: Aktionen, die ausgelöst werden sollen, wenn ein Datei-Upload erfolgreich war.
- Bei einem Fehler: Aktionen, die ausgelöst werden sollen, wenn ein Datei-Upload fehlschlägt.

Felder für S3-Upload-Ausdrücke

Die S3-Upload-Komponente bietet die folgenden Ausdrucksfelder:

- `{{ui.s3uploadID.files}}`: Gibt ein Array der hochgeladenen Dateien zurück.
- `{{ui.s3uploadID.files[0]?.size}}`: Gibt die Größe der Datei am angegebenen Index zurück.
- `{{ui.s3uploadID.files[0]?.type}}`: Gibt den Typ der Datei am angegebenen Index zurück.

- `{{ui.s3uploadID.files[0]?.nameOnly}}`: Gibt den Namen der Datei ohne Erweiterungssuffix am angegebenen Index zurück.
- `{{ui.s3uploadID.files[0]?.nameWithExtension}}`: Gibt den Namen der Datei mit ihrem Erweiterungssuffix am angegebenen Index zurück.

Ausdrücke und Beispiele

Beispiel: Zugriff auf hochgeladene Dateien

- `{{ui.s3uploadID.files.length}}`: Gibt die Anzahl der hochgeladenen Dateien zurück.
- `{{ui.s3uploadID.files.map(f => f.name).join(', ')}}`: Gibt eine durch Kommas getrennte Liste der Dateinamen zurück, die hochgeladen wurden.
- `{{ui.s3uploadID.files.filter(f => f.type.startsWith('image/'))}}`: Gibt ein Array zurück, das nur die hochgeladenen Bilddateien enthält.

Beispiel: Datei-Uploads validieren

- `{{ui.s3uploadID.files.some(f => f.size > 5 * 1024 * 1024)}}`: Prüft, ob eine der hochgeladenen Dateien mehr als 5 MB groß ist.
- `{{ui.s3uploadID.files.every(f => f.type === 'image/png')}}`: Prüft, ob es sich bei allen hochgeladenen Dateien um PNG-Bilder handelt.
- `{{ui.s3uploadID.files.length > 3}}`: Prüft, ob mehr als 3 Dateien hochgeladen wurden.

Beispiel: Aktionen auslösen

- `{{ui.s3uploadID.files.length > 0 ? 'Upload Successful' : 'No files uploaded'}}`: Zeigt eine Erfolgsmeldung an, wenn mindestens eine Datei hochgeladen wurde.
- `{{ui.s3uploadID.files.some(f => f.type.startsWith('video/')) ? triggerVideoProcessing() : null}}`: Löst eine Automatisierung der Videoverarbeitung aus, wenn Videodateien hochgeladen wurden.
- `{{ui.s3uploadID.files.map(f => f.url)}}`: Ruft die URLs hochgeladenen Dateien ab, die zur Anzeige oder Weiterverarbeitung der Dateien verwendet werden können.

Mit diesen Ausdrücken können Sie auf die hochgeladenen Dateien zugreifen, die Datei-Uploads validieren und Aktionen auslösen, die auf den Ergebnissen des Datei-Uploads basieren. Durch

die Verwendung dieser Ausdrücke können Sie innerhalb der Datei-Upload-Funktionalität Ihrer Anwendung ein dynamischeres und intelligenteres Verhalten erzeugen.

> **Note**
>
> `s3uploadID`Ersetzen Sie es durch die ID Ihrer S3-Upload-Komponente.

Komponente „PDF-Viewer“

Mit der PDF-Viewer-Komponente können Benutzer PDF-Dokumente in Ihrer Anwendung anzeigen und mit ihnen interagieren. App Studio unterstützt diese verschiedenen Eingabearten für die PDF-Quelle. Die PDF-Viewer-Komponente bietet Flexibilität bei der Integration von PDF-Dokumenten in Ihre Anwendung, unabhängig davon, ob es sich um eine statische URL, einen Inline-Daten-URI oder dynamisch generierten Inhalt handelt.

Eigenschaften des PDF-Viewers

Quelle

> **Note**
>
> Die Quelle der in dieser Komponente angezeigten Medien muss in den Inhaltssicherheitseinstellungen Ihrer Anwendung zulässig sein. Weitere Informationen finden Sie unter Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren.

Die Quelle des PDF-Dokuments, bei der es sich um einen Ausdruck, eine Entität, eine URL oder eine Automatisierung handeln kann.

Expression

Verwenden Sie einen Ausdruck, um die PDF-Quelle dynamisch zu generieren.

Entität

Connect die PDF-Viewer-Komponente mit einer Datenentität, die das PDF-Dokument enthält.

URL

Geben Sie die URL des PDF-Dokuments an.

URL

Sie können eine URL eingeben, die auf das PDF-Dokument verweist, das Sie anzeigen möchten. Dies kann eine öffentliche Web-URL oder eine URL in Ihrer eigenen Anwendung sein.

Beispiel: `https://example.com/document.pdf`

Daten-URI

Ein Daten-URI ist eine kompakte Methode, um kleine Datendateien (wie Bilder oder PDFs) direkt in Ihre Anwendung einzubinden. Das PDF-Dokument ist als Base64-Zeichenfolge codiert und direkt in der Konfiguration der Komponente enthalten.

Blob oder ArrayBuffer

Sie können das PDF-Dokument auch als Blob oder ArrayBuffer Objekt bereitstellen, sodass Sie die PDF-Daten dynamisch aus verschiedenen Quellen in Ihrer Anwendung generieren oder abrufen können.

Automatisierung

Connect die PDF-Viewer-Komponente mit einer Automatisierung, die das PDF-Dokument bereitstellt.

Aktionen

- Herunterladen: Fügt eine Schaltfläche oder einen Link hinzu, über den Benutzer das PDF-Dokument herunterladen können.

Layout

- Breite: Die Breite des PDF-Viewers, angegeben als Prozentsatz (%) oder als fester Pixelwert (z. B. 600 Pixel).
- Höhe: Die Höhe des PDF-Viewers, angegeben als fester Pixelwert.

Bildbetrachter

Mit der Image Viewer-Komponente können Benutzer Bilddateien in Ihrer Anwendung anzeigen und mit ihnen interagieren.

Eigenschaften des Bildbetrachters

Quelle

> **Note**
>
> Die Quelle der in dieser Komponente angezeigten Medien muss in den Inhaltssicherheitseinstellungen Ihrer Anwendung zulässig sein. Weitere Informationen finden Sie unter Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren.

- Entität: Connect die Image Viewer-Komponente mit einer Datenentität, die die Bilddatei enthält.
- URL: Geben Sie die URL der Bilddatei an.
- Ausdruck: Verwenden Sie einen Ausdruck, um die Bildquelle dynamisch zu generieren.
- Automatisierung: Connect die Image Viewer-Komponente mit einer Automatisierung, die die Bilddatei bereitstellt.

Alternativer Text

Die alternative Textbeschreibung des Bildes, die aus Gründen der Barrierefreiheit verwendet wird.

Layout

- Bildanpassung: Legt fest, wie das Bild in der Größe geändert und innerhalb der Komponente angezeigt werden soll. Beispiel: `Contain`, `Cover` oder `Fill`.
- Breite: Die Breite der Bildbetrachter-Komponente, angegeben als Prozentsatz (%) oder als fester Pixelwert (z. B. 300 Pixel).
- Höhe: Die Höhe der Image Viewer-Komponente, angegeben als fester Pixelwert.
- Hintergrund: Ermöglicht es Ihnen, eine Hintergrundfarbe oder ein Hintergrundbild für die Image Viewer-Komponente festzulegen.

Automatisierungen und Aktionen: Definieren Sie die Geschäftslogik Ihrer App

Mit Automatisierungen definieren Sie die Geschäftslogik Ihrer Anwendung. Die Hauptkomponenten einer Automatisierung sind: Auslöser, die die Automatisierung starten, eine Abfolge von einer oder

mehreren Aktionen, Eingabeparameter, mit denen Daten an die Automatisierung übergeben werden, und eine Ausgabe.

Themen

- Automationskonzepte
- Automatisierungen erstellen, bearbeiten und löschen
- Automatisierungsaktionen hinzufügen, bearbeiten und löschen
- Referenz zu Automation-Aktionen

Automationskonzepte

Im Folgenden finden Sie einige Konzepte und Begriffe, die Sie kennen sollten, wenn Sie die Geschäftslogik Ihrer App mithilfe von Automatisierungen in App Studio definieren und konfigurieren.

Automatisierungen

Mit Automatisierungen definieren Sie die Geschäftslogik Ihrer Anwendung. Die Hauptkomponenten einer Automatisierung sind: Auslöser, die die Automatisierung starten, eine Abfolge von einer oder mehreren Aktionen, Eingabeparameter, mit denen Daten an die Automatisierung übergeben werden, und eine Ausgabe.

Aktionen

Eine Automatisierungsaktion, allgemein als Aktion bezeichnet, ist ein einzelner Logikschritt, aus dem eine Automatisierung besteht. Jede Aktion führt eine bestimmte Aufgabe aus, sei es das Senden einer E-Mail, das Erstellen eines Datensatzes, das Aufrufen einer Lambda-Funktion oder das Aufrufen. APIs Aktionen werden zu Automatisierungen aus der Aktionsbibliothek hinzugefügt und können in bedingte Anweisungen oder Schleifen gruppiert werden.

Eingabeparameter für die Automatisierung

Automatisierungs-Eingabeparameter sind dynamische Eingabewerte, die Sie von Komponenten an Automatisierungen übergeben können, um sie flexibel und wiederverwendbar zu machen. Stellen Sie sich Parameter als Variablen für Ihre Automatisierung vor. Anstatt Werte fest in eine Automatisierung zu codieren, können Sie Parameter definieren und bei Bedarf unterschiedliche Werte angeben. Parameter ermöglichen es Ihnen, dieselbe Automatisierung bei jeder Ausführung mit unterschiedlichen Eingaben zu verwenden.

Verspottete Ausgabe

Einige Aktionen interagieren mithilfe von Konnektoren mit externen Ressourcen oder Diensten. Bei Verwendung der Vorschauumgebung interagieren Anwendungen nicht mit externen Diensten. Um Aktionen zu testen, die Konnektoren in der Vorschauumgebung verwenden, können Sie das Verhalten und die Ausgabe des Konnektors mithilfe von Mocked-Output simulieren. Die simulierte Ausgabe wird mit konfiguriert JavaScript, und das Ergebnis wird in den Ergebnissen einer Aktion gespeichert, genau wie die Antwort des Connectors in einer veröffentlichten App gespeichert wird.

Mithilfe von Mocking können Sie die Vorschauumgebung verwenden, um verschiedene Szenarien und deren Auswirkungen auf andere automatisierte Aktionen zu testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unglücklicher Pfade, ohne den externen Service über Konnektoren aufrufen zu müssen.

Ausgabe durch Automatisierung

Eine Automatisierungsausgabe wird verwendet, um Werte von einer Automatisierung an andere Ressourcen einer App zu übergeben, z. B. an Komponenten oder andere Automatisierungen. Automatisierungsausgaben werden als Ausdrücke konfiguriert, und der Ausdruck kann einen statischen Wert oder einen dynamischen Wert zurückgeben, der aus Automatisierungsparametern und Aktionen berechnet wird. Standardmäßig geben Automatisierungen keine Daten zurück, einschließlich der Ergebnisse von Aktionen innerhalb der Automatisierung.

Ein paar Beispiele dafür, wie Automatisierungsausgaben verwendet werden können:

- Sie können eine Automatisierungsausgabe so konfigurieren, dass sie ein Array zurückgibt, und dieses Array zum Auffüllen einer Datenkomponente übergeben.
- Sie können eine Automatisierung verwenden, um einen Wert zu berechnen und diesen Wert an mehrere andere Automatisierungen zu übergeben, um die Geschäftslogik zu zentralisieren und wiederzuverwenden.

Auslöser

Ein Trigger bestimmt, wann und unter welchen Bedingungen eine Automatisierung ausgeführt wird. Einige Beispiele für Trigger sind `On click` für Schaltflächen und `On select` für Texteingaben. Der Typ der Komponente bestimmt die Liste der verfügbaren Trigger für diese Komponente. Trigger werden den Komponenten hinzugefügt und im Application Studio konfiguriert.

Automatisierungen erstellen, bearbeiten und löschen

Inhalt

- Eine Automatisierung erstellen
- Automatisierungseigenschaften anzeigen oder bearbeiten
- Eine Automatisierung löschen

Eine Automatisierung erstellen

Gehen Sie wie folgt vor, um eine Automatisierung in einer App Studio-Anwendung zu erstellen. Nach der Erstellung muss eine Automatisierung konfiguriert werden, indem ihre Eigenschaften bearbeitet und Aktionen hinzugefügt werden.

Um eine Automatisierung zu erstellen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wenn Sie keine Automatisierungen haben, wählen Sie + Automatisierung hinzufügen auf der Arbeitsfläche. Andernfalls wählen Sie im Automationsmenü auf der linken Seite + Hinzufügen.
4. Eine neue Automatisierung wird erstellt, und Sie können damit beginnen, ihre Eigenschaften zu bearbeiten oder Aktionen hinzuzufügen und zu konfigurieren, um die Geschäftslogik Ihrer Anwendung zu definieren.

Automatisierungseigenschaften anzeigen oder bearbeiten

Gehen Sie wie folgt vor, um Automatisierungseigenschaften anzuzeigen oder zu bearbeiten.

Um Automatisierungseigenschaften anzuzeigen oder zu bearbeiten

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wählen Sie im Automationsmenü auf der linken Seite die Automatisierung aus, für die Sie Eigenschaften anzeigen oder bearbeiten möchten, um das Eigenschaftenmenü auf der rechten Seite zu öffnen.
4. Im Eigenschaftenmenü können Sie die folgenden Eigenschaften anzeigen:

- Automatisierungs-ID: Der eindeutige Name der Automatisierung. Um ihn zu bearbeiten, geben Sie einen neuen Bezeichner in das Textfeld ein.
- Automatisierungsparameter: Automatisierungsparameter werden verwendet, um dynamische Werte von der Benutzeroberfläche Ihrer App an Automatisierungs- und Datenaktionen zu übergeben. Um einen Parameter hinzuzufügen, wählen Sie + Hinzufügen. Wählen Sie das Stiftsymbol, um den Namen, die Beschreibung oder den Typ des Parameters zu ändern. Um einen Parameter zu entfernen, wählen Sie das Papierkorbsymbol.

  > **Tip**
  >
  > Sie können Automatisierungsparameter auch direkt von der Arbeitsfläche aus hinzufügen.

- Automatisierungsausgabe: Die Automatisierungsausgabe wird verwendet, um zu konfigurieren, auf welche Daten aus der Automatisierung in anderen Automatisierungen oder Komponenten verwiesen werden kann. Standardmäßig erstellen Automatisierungen keine Ausgabe. Um eine Automatisierungsausgabe hinzuzufügen, wählen Sie + Hinzufügen. Um die Ausgabe zu entfernen, wählen Sie das Papierkorbsymbol.

5. Sie definieren, was eine Automatisierung bewirkt, indem Sie Aktionen hinzufügen und konfigurieren. Weitere Informationen zu Aktionen finden Sie unter Automatisierungsaktionen hinzufügen, bearbeiten und löschen undReferenz zu Automation-Aktionen.

Eine Automatisierung löschen

Gehen Sie wie folgt vor, um eine Automatisierung in einer App Studio-Anwendung zu löschen.

Um eine Automatisierung zu löschen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wählen Sie im Automationsmenü auf der linken Seite das Ellipsenmenü der Automatisierung, die Sie löschen möchten, und wählen Sie Löschen. Alternativ können Sie das Papierkorbsymbol aus dem Eigenschaftenmenü der Automatisierung auf der rechten Seite auswählen.
4. Wählen Sie im Bestätigungs-Dialogfeld die Option Delete (Löschen).

Automatisierungsaktionen hinzufügen, bearbeiten und löschen

Eine Automatisierungsaktion, allgemein als Aktion bezeichnet, ist ein einzelner Logikschritt, aus dem eine Automatisierung besteht. Jede Aktion führt eine bestimmte Aufgabe aus, sei es das Senden einer E-Mail, das Erstellen eines Datensatzes, das Aufrufen einer Lambda-Funktion oder das Aufrufen. APIs Aktionen werden zu Automatisierungen aus der Aktionsbibliothek hinzugefügt und können in bedingte Anweisungen oder Schleifen gruppiert werden.

Inhalt

- Eine Automatisierungsaktion hinzufügen
- Eigenschaften von Automatisierungsaktionen anzeigen und bearbeiten
- Löschen einer Automatisierungsaktion

Eine Automatisierungsaktion hinzufügen

Gehen Sie wie folgt vor, um einer Automatisierung in einer App Studio-Anwendung eine Aktion hinzuzufügen.

Um eine Automatisierungsaktion hinzuzufügen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wählen Sie im Automationsmenü auf der linken Seite die Automatisierung aus, zu der Sie eine Aktion hinzufügen möchten.
4. Wählen Sie im Aktionsmenü auf der rechten Seite die Aktion aus, die Sie hinzufügen möchten, oder ziehen Sie die Aktion per Drag & Drop auf die Leinwand. Nachdem die Aktion erstellt wurde, können Sie die Aktion auswählen, um die Aktionseigenschaften zu konfigurieren und so die Funktionalität der Aktion zu definieren. Weitere Informationen zu Aktionseigenschaften und deren Konfiguration finden Sie unterReferenz zu Automation-Aktionen.

Eigenschaften von Automatisierungsaktionen anzeigen und bearbeiten

Gehen Sie wie folgt vor, um die Eigenschaften einer Automatisierungsaktion in einer App Studio-Anwendung anzuzeigen oder zu bearbeiten.

Um die Eigenschaften einer Automatisierungsaktion anzuzeigen oder zu bearbeiten

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wählen Sie im Automationsmenü auf der linken Seite die Aktion aus, deren Eigenschaften Sie anzeigen oder bearbeiten möchten. Alternativ können Sie die Aktion auf der Arbeitsfläche auswählen, wenn Sie sich die Automatisierung ansehen, die sie enthält.
4. Sie können die Aktionseigenschaften im Eigenschaftenmenü auf der rechten Seite anzeigen oder bearbeiten. Die Eigenschaften einer Aktion sind für jeden Aktionstyp unterschiedlich. Weitere Informationen zu Aktionseigenschaften und deren Konfiguration finden Sie unterReferenz zu Automation-Aktionen.

Löschen einer Automatisierungsaktion

Gehen Sie wie folgt vor, um eine Aktion aus einer Automatisierung in einer App Studio-Anwendung zu löschen.

Um eine Automatisierungsaktion zu löschen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie die Registerkarte Automatisierungen.
3. Wählen Sie im Automationsmenü auf der linken Seite die Automatisierung aus, die die Aktion enthält, die Sie löschen möchten.
4. Wählen Sie auf der Leinwand das Papierkorbsymbol in der Aktion, die Sie löschen möchten, und wählen Sie Löschen.

Referenz zu Automation-Aktionen

Im Folgenden finden Sie die Referenzdokumentation für Automatisierungsaktionen, die in App Studio verwendet werden.

Eine Automatisierungsaktion, allgemein als Aktion bezeichnet, ist ein einzelner Logikschritt, aus dem eine Automatisierung besteht. Jede Aktion führt eine bestimmte Aufgabe aus, sei es das Senden einer E-Mail, das Erstellen eines Datensatzes, das Aufrufen einer Lambda-Funktion oder das Aufrufen. APIs Aktionen werden zu Automatisierungen aus der Aktionsbibliothek hinzugefügt und können in bedingte Anweisungen oder Schleifen gruppiert werden.

Informationen zum Erstellen und Konfigurieren von Automatisierungen und deren Aktionen finden Sie in den Themen unter. Automatisierungen und Aktionen: Definieren Sie die Geschäftslogik Ihrer App

Rufen Sie die API auf

Ruft eine HTTP-REST-API-Anfrage auf. Builder können diese Aktion verwenden, um Anfragen von App Studio an andere Systeme oder Dienste mit APIs zu senden. Sie können damit beispielsweise eine Verbindung zu Systemen von Drittanbietern oder selbst entwickelten Anwendungen herstellen, um auf geschäftskritische Daten zuzugreifen, oder API-Endpunkte aufrufen, die nicht durch spezielle App Studio-Aktionen aufgerufen werden können.

Weitere Informationen zu REST APIs finden Sie unter Was ist eine API? RESTful .

Eigenschaften

Konnektor

Der Connector, der für die API-Anfragen verwendet werden soll, die durch diese Aktion gestellt werden. Die Connector-Dropdown-Liste enthält nur Konnektoren der folgenden Typen: `API Connector` und`OpenAPI Connector`. Je nachdem, wie der Connector konfiguriert ist, kann er wichtige Informationen wie Anmeldeinformationen und Standardheader oder Abfrageparameter enthalten.

Weitere Informationen zu API-Konnektoren, einschließlich eines Vergleichs zwischen der Verwendung von `API Connector` und`OpenAPI Connector`, finden Sie unterConnect zu Diensten von Drittanbietern her.

Eigenschaften der Konfiguration von API-Anfragen

Wählen Sie im Eigenschaftenbereich die Option API-Anfrage konfigurieren aus, um das Dialogfeld zur Anforderungskonfiguration zu öffnen. Wenn ein API-Konnektor ausgewählt ist, enthält das Dialogfeld Konnektorinformationen.

Methode: Die Methode für den API-Aufruf. Die möglichen Werte lauten wie folgt:

- `DELETE`: Löscht eine angegebene Ressource.
- `GET`: Ruft Informationen oder Daten ab.
- `HEAD`: Ruft nur die Kopfzeilen einer Antwort ohne den Hauptteil ab.
- `POST`: Sendet Daten zur Verarbeitung.
- `PUSH`: Sendet Daten zur Verarbeitung ein.

- `PATCH`: Aktualisiert eine angegebene Ressource teilweise.

Pfad: Der relative Pfad zur Ressource.

Header: Alle Header in Form von Schlüssel-Wert-Paaren, die mit der API-Anfrage gesendet werden sollen. Wenn ein Connector ausgewählt wird, werden die konfigurierten Header automatisch hinzugefügt und können nicht entfernt werden. Die konfigurierten Header können nicht bearbeitet werden, aber Sie können sie überschreiben, indem Sie einen weiteren Header mit demselben Namen hinzufügen.

Abfrageparameter: Alle Abfrageparameter in Form von Schlüssel-Wert-Paaren, die mit der API-Anfrage gesendet werden sollen. Wenn ein Konnektor ausgewählt wird, werden seine konfigurierten Abfrageparameter automatisch hinzugefügt und können nicht bearbeitet oder entfernt werden.

Hauptteil: Informationen, die mit der API-Anfrage im JSON-Format gesendet werden sollen. Es gibt keinen Hauptteil für `GET` Anfragen.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Aufrufen AWS

Ruft einen Vorgang von einem Dienst aus auf AWS . Dies ist eine allgemeine Aktion zum Aufrufen von AWS Diensten oder Vorgängen. Sie sollte verwendet werden, wenn es keine spezielle Aktion für den gewünschten AWS Dienst oder Vorgang gibt.

Eigenschaften

Service

Der AWS Dienst, der den auszuführenden Vorgang enthält.

Operation

Die Operation, die ausgeführt werden soll.

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Die JSON-Eingabe, die bei der Ausführung des angegebenen Vorgangs erfolgen soll. Weitere Informationen zur Konfiguration von Eingaben für AWS Operationen finden Sie unter AWS SDK für JavaScript.

Lambda aufrufen

Ruft eine bestehende Lambda-Funktion auf.

Eigenschaften

Konnektor

Der Konnektor, der für die Lambda-Funktionen verwendet werden soll, die von dieser Aktion ausgeführt werden. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für den Zugriff auf die Lambda-Funktion und anderen Konfigurationsinformationen, z. B. der AWS Region, die die Lambda-Funktion enthält, eingerichtet werden. Weitere Informationen zur Konfiguration eines Connectors für Lambda finden Sie unterSchritt 3: Lambda-Konnektor erstellen.

Funktionsname

Der Name der auszuführenden Lambda-Funktion. Beachten Sie, dass dies der Funktionsname und nicht der Funktions-ARN (Amazon Resource Name) ist.

Funktionsereignis

Schlüssel-Wert-Paare, die als Event-Payload an Ihre Lambda-Funktion weitergegeben werden.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen,

der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Loop

Führt verschachtelte Aktionen wiederholt aus, um eine Liste von Elementen nacheinander durchzugehen. Fügen Sie die Datensatz erstellen Aktion beispielsweise einer Schleifenaktion hinzu, um mehrere Datensätze zu erstellen.

Die Schleifenaktion kann innerhalb anderer Schleifen oder Bedingungsaktionen verschachtelt sein. Die Loop-Aktionen werden sequentiell und nicht parallel ausgeführt. Auf die Ergebnisse der einzelnen Aktionen innerhalb der Schleife können nur nachfolgende Aktionen innerhalb derselben Schleifeniteration zugreifen. Auf sie kann nicht außerhalb der Schleife oder in verschiedenen Iterationen der Schleife zugegriffen werden.

Eigenschaften

Quelle

Die Liste der Elemente, durch die iteriert werden soll, ein Element nach dem anderen. Die Quelle kann das Ergebnis einer vorherigen Aktion oder eine statische Liste von Zeichenfolgen, Zahlen oder Objekten sein, die Sie mithilfe eines JavaScript Ausdrucks bereitstellen können.

Beispiele

Die folgende Liste enthält Beispiele für Quelleingaben.

- Ergebnisse einer früheren Aktion: `{{results.actionName.data}}`
- Eine Liste von Zahlen: `{{[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}}`
- Eine Liste von Zeichenketten: `{{["apple", "banana", "orange", "grape", "kiwi"]}}`
- Ein berechneter Wert: `{{params.actionName.split("\n")}}`

Aktueller Artikelname

Der Name der Variablen, die verwendet werden kann, um auf das aktuelle Element zu verweisen, das gerade iteriert wird. Der aktuelle Elementname ist konfigurierbar, sodass Sie zwei oder mehr Schleifen verschachteln und von jeder Schleife aus auf Variablen zugreifen können. Wenn Sie beispielsweise Länder und Städte mit zwei Schleifen durchqueren, können Sie und konfigurieren `currentCountry` und `currentCity` referenzieren.

Bedingung

Führt Aktionen auf der Grundlage des Ergebnisses einer oder mehrerer bestimmter logischer Bedingungen aus, die bei der Ausführung der Automatisierung ausgewertet werden. Die Bedingungsaktion besteht aus den folgenden Komponenten:

- Ein Bedingungsfeld, das verwendet wird, um einen JavaScript Ausdruck bereitzustellen, der als `true` oder `false` ausgewertet wird.
- Ein echter Zweig, der Aktionen enthält, die ausgeführt werden, wenn die Bedingung als ausgewertet wird. `true`
- Ein falscher Zweig, der Aktionen enthält, die ausgeführt werden, wenn die Bedingung als erfüllt gilt. `false`

Fügen Sie den Verzweigungen „wahr“ und „falsch“ Aktionen hinzu, indem Sie sie in die Bedingungsaktion ziehen.

Eigenschaften

Bedingung

Der JavaScript Ausdruck, der ausgewertet werden soll, wenn die Aktion ausgeführt wird.

Datensatz erstellen

Erstellt einen Datensatz in einer vorhandenen App Studio-Entität.

Eigenschaften

Entität

Die Entität, in der ein Datensatz erstellt werden soll. Sobald eine Entität ausgewählt wurde, müssen den Feldern der Entität Werte hinzugefügt werden, damit der Datensatz erstellt werden kann. Die Typen der Felder und ob die Felder erforderlich oder optional sind, sind in der Entität definiert.

Datensatz aktualisieren

Aktualisiert einen vorhandenen Datensatz in einer App Studio-Entität.

Eigenschaften

Entität

Die Entität, die die zu aktualisierenden Datensätze enthält.

Bedingungen

Die Kriterien, die definieren, welche Datensätze durch die Aktion aktualisiert werden. Sie können Bedingungen gruppieren, um eine logische Aussage zu erstellen. Sie können Gruppen oder Bedingungen mit `AND` `OR` Oder-Anweisungen kombinieren.

Felder

Die Felder, die in den durch die Bedingungen angegebenen Datensätzen aktualisiert werden sollen.

Werte

Die Werte, die in den angegebenen Feldern aktualisiert werden sollen.

Datensatz löschen

Löscht einen Datensatz aus einer App Studio-Entität.

Eigenschaften

Entität

Die Entität, die die zu löschenden Datensätze enthält.

Bedingungen

Die Kriterien, die definieren, welche Datensätze durch die Aktion gelöscht werden. Sie können Bedingungen gruppieren, um eine logische Anweisung zu erstellen. Sie können Gruppen oder Bedingungen mit `AND` `OR` Oder-Anweisungen kombinieren.

Datenaktion aufrufen

Führt eine Datenaktion mit optionalen Parametern aus.

Eigenschaften

Datenaktion

Die Datenaktion, die von der Aktion ausgeführt werden soll.

Parameter

Datenaktionsparameter, die von der Datenaktion verwendet werden sollen. Datenaktionsparameter werden verwendet, um Werte zu senden, die als Eingaben für Datenaktionen verwendet werden. Datenaktionsparameter können bei der Konfiguration der Automatisierungsaktion hinzugefügt werden, müssen jedoch auf der Registerkarte Daten bearbeitet werden.

Erweiterte Einstellungen

Die `Invoke data action` Aktion enthält die folgenden erweiterten Einstellungen:

- Seitengröße: Die maximale Anzahl von Datensätzen, die in jeder Abfrage abgerufen werden können. Der Standardwert ist 500 und der Höchstwert ist 3000.
- Paginierungstoken: Das Token, das verwendet wird, um zusätzliche Datensätze aus einer Abfrage abzurufen. Wenn das beispielsweise auf 500 gesetzt `Page size` ist, es aber mehr als 500 Datensätze gibt, werden durch die Übergabe des Paginierungstokens an eine nachfolgende Abfrage die nächsten 500 abgerufen. Das Token ist undefiniert, wenn keine Datensätze oder Seiten mehr existieren.

Amazon S3: Objekt platzieren

Verwendet den `Amazon S3 PutObject` Vorgang, um ein Objekt, das durch einen Schlüssel (Dateipfad) identifiziert wird, zu einem angegebenen Amazon S3 S3-Bucket hinzuzufügen.

Eigenschaften

Konnektor

Der Connector, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den entsprechenden Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Die erforderlichen Optionen, die im `PutObject` Befehl verwendet werden sollen. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon S3 PutObject` Vorgang finden Sie PutObjectin der Amazon Simple Storage Service API-Referenz.

- Bucket: Der Name des Amazon S3 S3-Buckets, in den ein Objekt eingefügt werden soll.
- Schlüssel: Der eindeutige Name des Objekts, das in den Amazon S3 S3-Bucket aufgenommen werden soll.
- Hauptteil: Der Inhalt des Objekts, das in den Amazon S3-Bucket aufgenommen werden soll.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon S3: Objekt löschen

Verwendet den `Amazon S3 DeleteObject` Vorgang, um ein durch einen Schlüssel (Dateipfad) identifiziertes Objekt aus einem angegebenen Amazon S3 S3-Bucket zu löschen.

Eigenschaften

Konnektor

Der Connector, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des

Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Die erforderlichen Optionen, die im `DeleteObject` Befehl verwendet werden sollen. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon S3 DeleteObject` Vorgang finden Sie DeleteObjectin der Amazon Simple Storage Service API-Referenz.

- Bucket: Der Name des Amazon S3 S3-Buckets, aus dem ein Objekt gelöscht werden soll.
- Schlüssel: Der eindeutige Name des Objekts, das aus dem Amazon S3 S3-Bucket gelöscht werden soll.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon S3: Objekt abrufen

Verwendet den `Amazon S3 GetObject` Vorgang, um ein durch einen Schlüssel (Dateipfad) identifiziertes Objekt aus einem angegebenen Amazon S3 S3-Bucket abzurufen.

Eigenschaften

Konnektor

Der Connector, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Die erforderlichen Optionen, die im `GetObject` Befehl verwendet werden sollen. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon S3 GetObject` Vorgang finden Sie GetObjectin der Amazon Simple Storage Service API-Referenz.

- Bucket: Der Name des Amazon S3 S3-Buckets, aus dem ein Objekt abgerufen werden soll.
- Schlüssel: Der eindeutige Name des Objekts, das aus dem Amazon S3 S3-Bucket abgerufen werden soll.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon S3: Objekte auflisten

Verwendet den `Amazon S3 ListObjects` Vorgang, um Objekte in einem angegebenen Amazon S3 S3-Bucket aufzulisten.

Eigenschaften

Konnektor

Der Connector, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Die erforderlichen Optionen, die im `ListObjects` Befehl verwendet werden sollen. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon S3 ListObjects` Vorgang finden Sie ListObjectsin der Amazon Simple Storage Service API-Referenz.

- Bucket: Der Name des Amazon S3 S3-Buckets, aus dem Objekte aufgelistet werden sollen.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Textract: Dokument analysieren

Verwendet den `Amazon Textract AnalyzeDocument` Vorgang, um ein Eingabedokument auf Beziehungen zwischen erkannten Elementen zu analysieren.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `AnalyzeDocument` Befehl verwendet werden soll. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zur `Amazon Textract AnalyzeDocument` Bedienung finden Sie AnalyzeDocumentim Amazon Textract Developer Guide.

- Dokument//S3Object//Bucket: Der Name des Amazon S3 S3-Buckets. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Dokument//S3Object//Name: Der Dateiname des Eingabedokuments. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Document//S3Object//Version: Wenn für den Amazon S3 S3-Bucket die Versionierung aktiviert ist, können Sie die Version des Objekts angeben. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- FeatureTypes: Eine Liste der durchzuführenden Analysetypen. Gültige Werte: `TABLES`, `FORMS`, `QUERIES`, `SIGNATURES` und `LAYOUT`.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen,

der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Textract: Kosten analysieren

Verwendet den `Amazon Textract AnalyzeExpense` Vorgang, um ein Eingabedokument auf finanzielle Beziehungen zwischen Text zu analysieren.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `AnalyzeExpense` Befehl verwendet werden soll. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zur `Amazon Textract AnalyzeExpense` Bedienung finden Sie AnalyzeExpenseim Amazon Textract Developer Guide.

- Dokument//S3Object//Bucket: Der Name des Amazon S3 S3-Buckets. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Dokument//S3Object//Name: Der Dateiname des Eingabedokuments. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.

- Document//S3Object//Version: Wenn für den Amazon S3 S3-Bucket die Versionierung aktiviert ist, können Sie die Version des Objekts angeben. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Textract: ID analysieren

Verwendet den `Amazon Textract AnalyzeID` Vorgang, um ein Ausweisdokument auf relevante Informationen zu analysieren.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `AnalyzeID` Befehl verwendet werden soll. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon Textract AnalyzeID` Vorgang finden Sie unter AnalyzeId im Amazon Textract Developer Guide.

- Dokument//S3Object//Bucket: Der Name des Amazon S3 S3-Buckets. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Dokument//S3Object//Name: Der Dateiname des Eingabedokuments. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Document//S3Object//Version: Wenn für den Amazon S3 S3-Bucket die Versionierung aktiviert ist, können Sie die Version des Objekts angeben. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Textract: Dokumenttext erkennen

Verwendet den `Amazon Textract DetectDocumentText` Vorgang, um Textzeilen und die Wörter, aus denen eine Textzeile besteht, in einem Eingabedokument zu erkennen.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `DetectDocumentText` Befehl verwendet werden soll. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zur `Amazon Textract DetectDocumentText` Bedienung finden Sie DetectDocumentTextim Amazon Textract Developer Guide.

- Dokument//S3Object//Bucket: Der Name des Amazon S3 S3-Buckets. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Dokument//S3Object//Name: Der Dateiname des Eingabedokuments. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.
- Document//S3Object//Version: Wenn für den Amazon S3 S3-Bucket die Versionierung aktiviert ist, können Sie die Version des Objekts angeben. Dieser Parameter kann leer gelassen werden, wenn eine Datei an die Aktion mit der S3-Upload-Komponente übergeben wird.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Bedrock: GenAI-Aufforderung

Verwendet den Amazon InvokeModel Bedrock-Vorgang, um Inferenzen mithilfe der in den Aktionseigenschaften angegebenen Eingabeaufforderungs- und Inferenzparameter auszuführen. Die Aktion kann Text, Bilder und Einbettungen generieren.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Um diese Aktion erfolgreich zu verwenden, muss der Connector mit Amazon Bedrock

Runtime als Service konfiguriert werden. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Modell

Das Basismodell, das von Amazon Bedrock zur Bearbeitung der Anfrage verwendet werden soll. Weitere Informationen zu Modellen in Amazon Bedrock finden Sie unter Amazon Bedrock Foundation-Modellinformationen im Amazon Bedrock-Benutzerhandbuch.

Eingabetyp

Der Eingangstyp der Eingabe, die an das Amazon Bedrock-Modell gesendet wird. Die möglichen Werte sind Text, Dokument und Bild. Wenn ein Eingabetyp nicht zur Auswahl steht, wird er wahrscheinlich vom konfigurierten Modell nicht unterstützt.

Benutzeraufforderung

Die Aufforderung, an das Amazon Bedrock-Modell gesendet zu werden, um verarbeitet zu werden, um eine Antwort zu generieren. Sie können statischen Text eingeben oder eine Eingabe aus einem anderen Teil Ihrer Anwendung übergeben, z. B. von einer Komponente, die Parameter verwendet, einer vorherigen Aktion in der Automatisierung oder einer anderen Automatisierung. Die folgenden Beispiele zeigen, wie Sie einen Wert aus einer Komponente oder einer vorherigen Aktion übergeben:

- Um einen Wert aus einer Komponente mithilfe von Parametern zu übergeben: `{{params.paramName}}`
- Um einen Wert aus einer vorherigen Aktion zu übergeben: `{{results.actionName}}`

Systemaufforderung (Claude-Modelle)

Die Systemaufforderung, bei der Bearbeitung der Anfrage vom Amazon Bedrock-Modell verwendet zu werden. Die Systemaufforderung wird verwendet, um Claude-Modellen Kontext, Anweisungen oder Richtlinien bereitzustellen.

Einstellungen anfordern

Konfigurieren Sie verschiedene Anforderungseinstellungen und Modellinferenzparameter. Sie können die folgenden Einstellungen konfigurieren:

- Temperatur: Die Temperatur, die vom Amazon Bedrock-Modell bei der Bearbeitung der Anfrage verwendet werden soll. Die Temperatur bestimmt die Zufälligkeit oder Kreativität der Ausgabe des Bedrock-Modells. Je höher die Temperatur, desto kreativer und weniger analytisch ist die Reaktion. Mögliche Werte sind`[0-10]`.
- Max. Tokens: Begrenzen Sie die Länge der Ausgabe des Amazon Bedrock-Modells.
- TopP: Beim Nucleus-Sampling berechnet das Modell die kumulative Verteilung über alle Optionen für jedes nachfolgende Token in absteigender Wahrscheinlichkeitsreihenfolge und schneidet sie ab, sobald eine bestimmte, durch TopP angegebene Wahrscheinlichkeit erreicht wird. Sie sollten entweder die Temperatur oder den TopP ändern, aber nicht beide
- Stopp-Sequenzen: Sequenzen, die dazu führen, dass das Modell die Verarbeitung der Anforderung und die Generierung der Ausgabe beendet.

Weitere Informationen finden Sie unter Inferenz-Anforderungsparameter und Antwortfelder für Foundation-Modelle im Amazon Bedrock-Benutzerhandbuch.

Sequenzen beenden

Geben Sie eine Amazon Bedrock Guardrail ID und Version ein. Guardrails werden verwendet, um Schutzmaßnahmen zu implementieren, die auf Ihren Anwendungsfällen und verantwortungsvollen KI-Richtlinien basieren. Weitere Informationen finden Sie unter Abwehr schädlicher Inhalte in Modellen mit Amazon Bedrock Guardrails im Amazon Bedrock-Benutzerhandbuch.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Amazon Bedrock: Modell aufrufen

Verwendet den Amazon InvokeModel Bedrock-Vorgang, um Inferenzen mithilfe der im Anforderungstext angegebenen Eingabeaufforderungs- und Inferenzparameter auszuführen. Sie verwenden Modellinferenz, um Text, Bilder und Einbettungen zu generieren.

Eigenschaften

Konnektor

Der Konnektor, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Um diese Aktion erfolgreich zu verwenden, muss der Connector mit Amazon Bedrock Runtime als Service konfiguriert werden. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `InvokeModel` Befehl verwendet werden soll.

> **Note**
>
> Weitere Informationen zu diesem `Amazon Bedrock InvokeModel` Vorgang, einschließlich Beispielbefehlen, finden Sie InvokeModelin der Amazon Bedrock API-Referenz.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

JavaScript

Führt eine benutzerdefinierte JavaScript Funktion aus, um einen bestimmten Wert zurückzugeben.

> **Important**
>
> App Studio unterstützt nicht die Verwendung von Bibliotheken von Drittanbietern oder benutzerdefinierten JavaScript Bibliotheken.

Eigenschaften

Quellcode

Der JavaScript Codeausschnitt, der von der Aktion ausgeführt werden soll.

> **Tip**
>
> Sie können KI verwenden, um die Generierung JavaScript für Sie zu unterstützen, indem Sie die folgenden Schritte ausführen:
>
> 1. Wählen Sie das Erweiterungssymbol, um den erweiterten JavaScript Editor zu öffnen.
> 2. (Optional): Aktivieren Sie den Schalter Code ändern, um einen vorhandenen JavaScript Code zu ändern. Andernfalls ersetzt KI alle vorhandenen JavaScript.
> 3. Beschreiben Sie in Generate JavaScript, was Sie damit machen möchten JavaScript, zum Beispiel:**`Add two numbers`**.
> 4. Wählen Sie das Sendesymbol, um Ihre zu generieren JavaScript.

Rufen Sie die Automatisierung auf

Führt eine angegebene Automatisierung aus.

Eigenschaften

Rufen Sie die Automatisierung auf

Die Automatisierung, die von der Aktion ausgeführt werden soll.

E-Mail senden

Verwendet den `Amazon SES SendEmail` Vorgang, um eine E-Mail zu senden.

Eigenschaften

Konnektor

Der Connector, der für die von dieser Aktion ausgeführten Operationen verwendet werden soll. Der konfigurierte Connector sollte mit den richtigen Anmeldeinformationen für die Ausführung des Vorgangs und anderen Konfigurationsinformationen eingerichtet werden, z. B. mit der AWS Region, in der sich alle Ressourcen befinden, auf die im Vorgang verwiesen wird.

Konfiguration

Der Inhalt der Anforderung, der im `SendEmail` Befehl verwendet werden soll. Es handelt sich um folgende Optionen:

> **Note**
>
> Weitere Informationen zu diesem `Amazon SES SendEmail` Vorgang finden Sie SendEmailin der Amazon Simple Email Service API-Referenz.

Verspottete Ausgabe

Aktionen interagieren nicht mit externen Diensten oder Ressourcen in der Vorschauumgebung. Das Ausgabefeld Mocked wird verwendet, um zu Testzwecken einen JSON-Ausdruck bereitzustellen, der das Verhalten eines Konnektors in der Vorschauumgebung simuliert. Dieses Snippet wird in der `results` Map der Aktion gespeichert, genau wie die Antwort des Connectors für eine veröffentlichte App in der Live-Umgebung.

Mit diesem Feld können Sie verschiedene Szenarien und ihre Auswirkungen auf andere Aktionen innerhalb der Automatisierung testen, z. B. die Simulation verschiedener Ergebniswerte, Fehlerszenarien, Grenzfälle oder unpassender Pfade, ohne mit externen Diensten über Konnektoren zu kommunizieren.

Entitäten und Datenaktionen: Konfigurieren Sie das Datenmodell Ihrer App

Entitäten sind Datentabellen in App Studio. Entitäten interagieren direkt mit Tabellen in Datenquellen. Entitäten umfassen Felder zur Beschreibung der darin enthaltenen Daten, Abfragen zum Suchen und Zurückgeben von Daten sowie Zuordnungen, um die Felder der Entität mit den Spalten einer Datenquelle zu verbinden.

Themen

- Bewährte Methoden beim Entwerfen von Datenmodellen
- Eine Entität in einer App Studio-App erstellen
- Konfiguration oder Bearbeitung einer Entität in einer App Studio-App
- Eine Entität löschen
- Verwaltete Datenentitäten in AWS App Studio

Bewährte Methoden beim Entwerfen von Datenmodellen

Verwenden Sie die folgenden bewährten Methoden, um ein robustes, skalierbares und sicheres relationales Datenmodell AWS für die Verwendung in Ihrer App Studio-Anwendung zu erstellen, das die Anforderungen Ihrer Anwendung erfüllt und die langfristige Zuverlässigkeit und Leistung Ihrer Dateninfrastruktur gewährleistet.

- Wählen Sie den richtigen AWS Datendienst: Wählen Sie je nach Ihren Anforderungen den entsprechenden AWS Datendienst aus. Für eine OLTP-Anwendung (Online Transaction Processing) könnten Sie beispielsweise eine Datenbank (DB) wie Amazon Aurora in Betracht ziehen, bei der es sich um einen cloudnativen, relationalen und vollständig verwalteten Datenbankservice handelt, der verschiedene Datenbank-Engines wie MySQL und PostgreSQL unterstützt. Eine vollständige Liste der von App Studio unterstützten Aurora-Versionen finden Sie unterConnect zu Amazon Aurora her. Andererseits sollten Sie für OLAP-Anwendungsfälle (Online Analytical Processing) die Verwendung von Amazon Redshift in Betracht ziehen, einem Cloud-Data Warehouse, mit dem Sie komplexe Abfragen für sehr große Datensätze ausführen können. Diese Abfragen können oft einige Zeit (viele Sekunden) in Anspruch nehmen, sodass Amazon Redshift für OLTP-Anwendungen, die Datenzugriff mit niedriger Latenz benötigen, weniger geeignet ist.
- Skalierbares Design: Planen Sie Ihr Datenmodell mit Blick auf future Wachstum und Skalierbarkeit. Berücksichtigen Sie Faktoren wie das erwartete Datenvolumen, Zugriffsmuster und

Leistungsanforderungen bei der Auswahl eines geeigneten Datendienstes und einer geeigneten Konfiguration der Datenbankinstanz (z. B. bereitgestellte Kapazität).

  - Weitere Informationen zur Skalierung mit Aurora Serverless finden Sie unter Leistung und Skalierung für Aurora Serverless V2.

- Normalisieren Sie Ihre Daten: Folgen Sie den Prinzipien der Datenbanknormalisierung, um Datenredundanz zu minimieren und die Datenintegrität zu verbessern. Dazu gehören das Erstellen geeigneter Tabellen, das Definieren von Primär- und Fremdschlüsseln und das Herstellen von Beziehungen zwischen Entitäten. Wenn Sie in App Studio Daten von einer Entität abfragen, können Sie zugehörige Daten von einer anderen Entität abrufen, indem Sie eine `join` Klausel in der Abfrage angeben.
- Implementieren Sie eine angemessene Indizierung: Identifizieren Sie die wichtigsten Abfragen und Zugriffsmuster und erstellen Sie geeignete Indizes, um die Leistung zu optimieren.
- Nutzen Sie die Funktionen von AWS Datendiensten: Nutzen Sie die Funktionen, die der von Ihnen gewählte AWS Datendienst bietet, z. B. automatisierte Backups, Multi-AZ-Bereitstellungen und automatische Softwareupdates.
- Schützen Sie Ihre Daten: Implementieren Sie robuste Sicherheitsmaßnahmen wie IAM (AWS Identity and Access Management) -Richtlinien, erstellen Sie Datenbankbenutzer mit eingeschränkten Rechten für Tabellen und Schemas und setzen Sie Verschlüsselung im Ruhezustand und bei der Übertragung durch.
- Überwachen und optimieren Sie die Leistung: Überwachen Sie kontinuierlich die Leistung Ihrer Datenbank und nehmen Sie bei Bedarf Anpassungen vor, z. B. die Skalierung von Ressourcen, die Optimierung von Abfragen oder die Optimierung von Datenbankkonfigurationen.
- Automatisieren Sie das Datenbankmanagement: Nutzen Sie AWS Dienste wie Aurora Autoscaling, Performance Insights for Aurora und AWS Database Migration Service, um Datenbankverwaltungsaufgaben zu automatisieren und den betrieblichen Aufwand zu reduzieren.
- Implementieren Sie Disaster Recovery- und Backup-Strategien: Stellen Sie sicher, dass Sie über einen klar definierten Sicherungs- und Wiederherstellungsplan verfügen, der Funktionen wie Aurora Automated Backups, point-in-time Recovery und regionsübergreifende Replikatkonfigurationen nutzt.
- Halten Sie sich an AWS bewährte Verfahren und die Dokumentation: Halten up-to-date Sie sich an die neuesten AWS Best Practices, Richtlinien und Dokumentationen für den von Ihnen ausgewählten Datenservice, um sicherzustellen, dass Ihr Datenmodell und Ihre Implementierung den Empfehlungen entsprechen. AWS

Ausführlichere Anleitungen zu den einzelnen AWS Datendiensten finden Sie in den folgenden Themen:

- Bewährte Methoden mit Amazon Aurora
- Bewährte Methoden mit Amazon Aurora MySQL
- Leistungsoptimierung für Amazon Redshift Redshift-Abfragen
- Bewährte Methoden für das Abfragen und Scannen von Daten in Amazon DynamoDB

Eine Entität in einer App Studio-App erstellen

Es gibt vier Methoden zum Erstellen einer Entität in einer App Studio-App. Die folgende Liste enthält jede Methode, ihre Vorteile und einen Link zu den Anweisungen, wie Sie diese Methode verwenden, um die Entität zu erstellen und anschließend zu konfigurieren.

- Eine Entität aus einer vorhandenen Datenquelle erstellen: Erstellen Sie automatisch eine Entität und ihre Felder aus einer vorhandenen Datenquellentabelle und ordnen Sie die Felder den Spalten der Datenquellentabelle zu. Diese Option ist vorzuziehen, wenn Sie über eine bestehende Datenquelle verfügen, die Sie in Ihrer App Studio-App verwenden möchten.
- Eine Entität mit einer von App Studio verwalteten Datenquelle erstellen: Erstellen Sie eine Entität und eine DynamoDB-Tabelle, die App Studio für Sie verwaltet. Die DynamoDB-Tabelle wird automatisch aktualisiert, wenn Sie Ihre Entität aktualisieren. Mit dieser Option müssen Sie eine Datenquelle eines Drittanbieters nicht manuell erstellen, verwalten oder eine Verbindung herstellen oder die Zuordnung von Entitätsfeldern zu Tabellenspalten festlegen. Die gesamte Datenmodellierung und Konfiguration Ihrer App erfolgt in App Studio. Diese Option ist vorzuziehen, wenn Sie Ihre eigenen Datenquellen und eine DynamoDB-Tabelle nicht verwalten möchten und ihre Funktionalität für Ihre App ausreichend ist.
- Eine leere Entität erstellen: Erstellen Sie eine leere Entität von Grund auf neu. Diese Option ist vorzuziehen, wenn Sie keine vorhandenen Datenquellen oder Konnektoren haben, die von einem Administrator erstellt wurden, und Sie das Datenmodell Ihrer App flexibel gestalten möchten, ohne durch externe Datenquellen eingeschränkt zu werden. Sie können die Entität nach der Erstellung mit einer Datenquelle verbinden.
- Eine Entität mit KI erstellen: Generieren Sie eine Entität, Felder, Datenaktionen und Beispieldaten auf der Grundlage des angegebenen Entitätsnamens. Diese Option ist vorzuziehen, wenn Sie eine Vorstellung vom Datenmodell für Ihre App haben, aber Hilfe bei der Übersetzung in eine Entität benötigen.

Eine Entität aus einer vorhandenen Datenquelle erstellen

Verwenden Sie eine Tabelle aus einer Datenquelle, um automatisch eine Entität und ihre Felder zu erstellen und die Entitätsfelder den Spalten der Tabelle zuzuordnen. Diese Option ist vorzuziehen, wenn Sie über eine bestehende Datenquelle verfügen, die Sie in Ihrer App Studio-App verwenden möchten.

1. Navigieren Sie bei Bedarf zu Ihrer Anwendung.
2. Wählen Sie oben auf der Leinwand die Registerkarte Daten aus.
3. Wenn Ihre App keine Entitäten enthält, wählen Sie + Entität erstellen. Andernfalls wählen Sie im Menü Entitäten auf der linken Seite + Hinzufügen.
4. Wählen Sie Tabelle aus einer vorhandenen Datenquelle verwenden aus.
5. Wählen Sie unter Connector den Connector aus, der die Tabelle enthält, die Sie zum Erstellen Ihrer Entität verwenden möchten.
6. Wählen Sie unter Tabelle die Tabelle aus, die Sie zum Erstellen Ihrer Entität verwenden möchten.
7. Aktivieren Sie das Kontrollkästchen Datenaktionen erstellen, um Datenaktionen zu erstellen.
8. Klicken Sie auf Create entity (Entity erstellen). Ihre Entität ist jetzt erstellt und Sie können sie im Bereich Entitäten auf der linken Seite sehen.
9. Konfigurieren Sie Ihre neue Entität, indem Sie den Anweisungen unter folgen. Konfiguration oder Bearbeitung einer Entität in einer App Studio-App Beachten Sie, dass einige Eigenschaften oder Ressourcen bereits erstellt wurden, da Ihre Entität mit einer vorhandenen Datenquelle erstellt wurde, z. B. Felder, die verbundene Datenquelle und Feldzuordnung. Außerdem enthält Ihre Entität Datenaktionen, wenn Sie bei der Erstellung das Kontrollkästchen Datenaktionen erstellen aktiviert haben.

Eine Entität mit einer von App Studio verwalteten Datenquelle erstellen

Erstellen Sie eine verwaltete Entität und die entsprechende DynamoDB-Tabelle, die von App Studio verwaltet wird. Die DynamoDB-Tabelle ist zwar im zugehörigen AWS Konto vorhanden, aber wenn Änderungen an der Entität in der App Studio-App vorgenommen werden, wird die DynamoDB-Tabelle automatisch aktualisiert. Mit dieser Option müssen Sie eine Datenquelle eines Drittanbieters nicht manuell erstellen, verwalten oder eine Verbindung herstellen oder die Zuordnung von Entitätsfeldern zu Tabellenspalten festlegen. Diese Option ist vorzuziehen, wenn Sie Ihre eigenen Datenquellen und eine DynamoDB-Tabelle nicht verwalten möchten und ihre Funktionalität für Ihre App ausreichend ist.

Weitere Informationen zu verwalteten Entitäten finden Sie unter. Verwaltete Datenentitäten in AWS App Studio

Sie können dieselben verwalteten Entitäten in mehreren Anwendungen verwenden. Detaillierte Anweisungen finden Sie unter Eine Entität aus einer vorhandenen Datenquelle erstellen.

1. Navigieren Sie bei Bedarf zu Ihrer Anwendung.
2. Wählen Sie oben auf der Leinwand die Registerkarte Daten aus.
3. Wenn Ihre App keine Entitäten enthält, wählen Sie + Entität erstellen. Andernfalls wählen Sie im Menü Entitäten auf der linken Seite + Hinzufügen.
4. Wählen Sie Verwaltete App Studio-Entität erstellen aus.
5. Geben Sie im Feld Entitätsname einen Namen für Ihre Entität ein.
6. Geben Sie im Feld Primärschlüssel einen Namen für den Primärschlüssel Ihrer Entität ein. Der Primärschlüssel ist der eindeutige Bezeichner der Entität und kann nach der Erstellung der Entität nicht geändert werden.
7. Wählen Sie unter Primärschlüssel-Datentyp den Datentyp des Primärschlüssels Ihrer Entität aus. Der Datentyp kann nach der Erstellung der Entität nicht geändert werden.
8. Klicken Sie auf Create entity (Entity erstellen). Ihre Entität ist jetzt erstellt und Sie können sie im Bereich Entitäten auf der linken Seite sehen.
9. Konfigurieren Sie Ihre neue Entität, indem Sie den Anweisungen unter folgen. Konfiguration oder Bearbeitung einer Entität in einer App Studio-App Beachten Sie, dass einige Eigenschaften oder Ressourcen bereits erstellt wurden, da Ihre Entität mit verwalteten Daten erstellt wurde, z. B. das Primärschlüsselfeld und die verbundene Datenquelle.

Eine leere Entität erstellen

Erstellen Sie eine leere Entität von Grund auf neu. Diese Option ist vorzuziehen, wenn Sie keine vorhandenen Datenquellen oder Konnektoren haben, die von einem Administrator erstellt wurden. Das Erstellen einer leeren Entität bietet Flexibilität, da Sie Ihre Entität in Ihrer App Studio-App entwerfen können, ohne durch externe Datenquellen eingeschränkt zu sein. Nachdem Sie das Datenmodell Ihrer App entworfen und die Entität entsprechend konfiguriert haben, können Sie sie später immer noch mit einer externen Datenquelle verbinden.

1. Navigieren Sie bei Bedarf zu Ihrer Anwendung.
2. Wählen Sie oben auf der Leinwand die Registerkarte Daten aus.

3. Wenn Ihre App keine Entitäten enthält, wählen Sie + Entität erstellen. Andernfalls wählen Sie im Menü Entitäten auf der linken Seite + Hinzufügen.
4. Wählen Sie Entität erstellen aus.
5. Klicken Sie auf Create entity (Entity erstellen). Ihre Entität ist jetzt erstellt und Sie können sie im Bereich Entitäten auf der linken Seite sehen.
6. Konfigurieren Sie Ihre neue Entität, indem Sie den Anweisungen unter folgen. Konfiguration oder Bearbeitung einer Entität in einer App Studio-App

Eine Entität mit KI erstellen

Generieren Sie eine Entität, Felder, Datenaktionen und Beispieldaten auf der Grundlage des angegebenen Entitätsnamens. Diese Option ist vorzuziehen, wenn Sie eine Vorstellung vom Datenmodell für Ihre App haben, aber Hilfe bei der Übersetzung in eine Entität benötigen.

1. Navigieren Sie bei Bedarf zu Ihrer Anwendung.
2. Wählen Sie oben auf der Leinwand die Registerkarte Daten aus.
3. Wenn Ihre App keine Entitäten enthält, wählen Sie + Entität erstellen. Andernfalls wählen Sie im Menü Entitäten auf der linken Seite + Hinzufügen.
4. Wählen Sie Eine Entität mit KI erstellen aus.
5. Geben Sie im Feld Entitätsname einen Namen für Ihre Entität ein. Dieser Name wird verwendet, um die Felder, Datenaktionen und Beispieldaten Ihrer Entität zu generieren.
6. Aktivieren Sie das Kontrollkästchen Datenaktionen erstellen, um Datenaktionen zu erstellen.
7. Wählen Sie Entität generieren aus. Ihre Entität ist jetzt erstellt und Sie können sie im Bereich Entitäten auf der linken Seite sehen.
8. Konfigurieren Sie Ihre neue Entität, indem Sie den Anweisungen unter folgen. Konfiguration oder Bearbeitung einer Entität in einer App Studio-App Beachten Sie, dass Ihre Entität bereits generierte Felder enthält, da Ihre Entität mit KI erstellt wurde. Außerdem wird Ihre Entität Datenaktionen enthalten, wenn Sie bei der Erstellung das Kontrollkästchen Datenaktionen erstellen aktiviert haben.

Konfiguration oder Bearbeitung einer Entität in einer App Studio-App

Verwenden Sie die folgenden Themen, um eine Entität in einer App Studio-Anwendung zu konfigurieren.

Themen

- Den Namen der Entität bearbeiten
- Entitätsfelder hinzufügen, bearbeiten oder löschen
- Datenaktionen erstellen, bearbeiten oder löschen
- Beispieldaten hinzufügen oder löschen
- Fügen Sie verbundene Datenquellen hinzu oder bearbeiten Sie sie und ordnen Sie Felder zu

Den Namen der Entität bearbeiten

1. Navigieren Sie bei Bedarf zu der Entität, die Sie bearbeiten möchten.
2. Aktualisieren Sie auf der Registerkarte Konfiguration unter Entitätsname den Entitätsnamen und wählen Sie außerhalb des Textfeldes, um Ihre Änderungen zu speichern.

Entitätsfelder hinzufügen, bearbeiten oder löschen

> **Tip**
>
> Sie können STRG+Z drücken, um die letzte Änderung an Ihrer Entität rückgängig zu machen.

1. Navigieren Sie bei Bedarf zu der Entität, die Sie bearbeiten möchten.
2. Auf der Registerkarte Konfiguration sehen Sie sich unter Felder eine Tabelle mit den Feldern Ihrer Entität an. Entitätsfelder haben die folgenden Spalten:
   - Anzeigename: Der Anzeigename ähnelt einer Tabellenüberschrift oder einem Formularfeld und ist für Anwendungsbenutzer sichtbar. Er kann Leerzeichen und Sonderzeichen enthalten, muss jedoch innerhalb einer Entität eindeutig sein.
   - Systemname: Der Systemname ist eine eindeutige Kennung, die im Code verwendet wird, um auf ein Feld zu verweisen. Bei der Zuordnung zu einer Spalte in einer Amazon Redshift Redshift-Tabelle muss diese mit dem Spaltennamen der Amazon Redshift Redshift-Tabelle übereinstimmen.
   - Datentyp: Der Datentyp, der in diesem Feld gespeichert wird, z. B. `IntegerBoolean`, oder. `String`
3. Um Felder hinzuzufügen:

a. Um mithilfe von KI Felder auf der Grundlage des Entitätsnamens und der verbundenen Datenquelle zu generieren, wählen Sie Weitere Felder generieren aus.

b. Um ein einzelnes Feld hinzuzufügen, wählen Sie + Feld hinzufügen.

4. Um ein Feld zu bearbeiten:

a. Um den Anzeigenamen zu bearbeiten, geben Sie den gewünschten Wert in das Textfeld Anzeigename ein. Wenn der Systemname des Felds nicht bearbeitet wurde, wird er auf den neuen Wert des Anzeigenamens aktualisiert.

b. Um den Systemnamen zu bearbeiten, geben Sie den gewünschten Wert in das Textfeld Systemname ein.

c. Um den Datentyp zu bearbeiten, wählen Sie das Dropdownmenü Datentyp und wählen Sie den gewünschten Typ aus der Liste aus.

d. Um die Eigenschaften des Felds zu bearbeiten, wählen Sie das Zahnradsymbol des Felds. In der folgenden Liste werden die Feldeigenschaften detailliert beschrieben:

- Erforderlich: Aktivieren Sie diese Option, wenn das Feld für Ihre Datenquelle erforderlich ist.
- Primärschlüssel: Aktivieren Sie diese Option, wenn das Feld einem Primärschlüssel in Ihrer Datenquelle zugeordnet ist.
- Eindeutig: Aktivieren Sie diese Option, wenn der Wert dieses Felds eindeutig sein muss.
- Datenquellenstandard verwenden: Aktivieren Sie diese Option, wenn der Wert des Felds von der Datenquelle bereitgestellt wird, z. B. mithilfe von Autoinkrement oder einem Ereigniszeitstempel.
- Datentypoptionen: Felder bestimmter Datentypen können mit Datentypoptionen wie Minimal- oder Maximalwerten konfiguriert werden.

5. Um ein Feld zu löschen, wählen Sie das Papierkorbsymbol des Felds, das Sie löschen möchten.

Datenaktionen erstellen, bearbeiten oder löschen

Datenaktionen werden in Anwendungen verwendet, um Aktionen für die Daten einer Entität auszuführen, z. B. das Abrufen aller Datensätze oder das Abrufen eines Datensatzes anhand der ID. Datenaktionen können verwendet werden, um Daten zu finden und zurückzugeben, die bestimmten Bedingungen entsprechen, sodass sie in Komponenten wie Tabellen oder Detailansichten angezeigt werden können.

Inhalt

- Datenaktionen erstellen
- Datenaktionen bearbeiten oder konfigurieren
- Bedingungsoperatoren für Datenaktionen und Beispiele
  - Unterstützung für Bedingungsoperatoren durch die Datenbank
  - Beispiele für Bedingungen für Datenaktionen
- Datenaktionen löschen

Datenaktionen erstellen

> **Tip**
>
> Sie können STRG+Z drücken, um die letzte Änderung an Ihrer Entität rückgängig zu machen.

1. Navigieren Sie bei Bedarf zu der Entität, für die Sie Datenaktionen erstellen möchten.
2. Wählen Sie die Registerkarte Datenaktionen aus.
3. Es gibt zwei Methoden zum Erstellen von Datenaktionen:
   - (Empfohlen) Wenn Sie KI verwenden möchten, um Datenaktionen auf der Grundlage Ihres Entitätsnamens, Ihrer Felder und der verbundenen Datenquelle für Sie zu generieren, wählen Sie Datenaktionen generieren aus. Die folgenden Aktionen werden generiert:
     1. `getAll`: Ruft alle Datensätze von einer Entität ab. Diese Aktion ist nützlich, wenn Sie eine Liste von Datensätzen anzeigen oder Operationen an mehreren Datensätzen gleichzeitig ausführen müssen.
     2. `getByID`: Ruft einen einzelnen Datensatz von einer Entität auf der Grundlage seiner eindeutigen Kennung (ID oder Primärschlüssel) ab. Diese Aktion ist nützlich, wenn Sie einen bestimmten Datensatz anzeigen oder Operationen an einem bestimmten Datensatz ausführen müssen.
   - Um eine einzelne Datenaktion hinzuzufügen, wählen Sie + Datenaktion hinzufügen.
4. Informationen zum Anzeigen oder Konfigurieren der neuen Datenaktion finden Sie im folgenden Abschnitt,Datenaktionen bearbeiten oder konfigurieren.

Datenaktionen bearbeiten oder konfigurieren

1. Navigieren Sie bei Bedarf zu der Entität, für die Sie Datenaktionen erstellen möchten.
2. Wählen Sie die Registerkarte Datenaktionen aus.
3. Konfigurieren Sie unter Felder die Felder, die von der Abfrage zurückgegeben werden sollen. Standardmäßig sind alle konfigurierten Felder in der Entität ausgewählt.

   Sie können der Datenaktion auch Joins hinzufügen, indem Sie die folgenden Schritte ausführen:

   1. Wählen Sie + Join hinzufügen, um ein Dialogfeld zu öffnen.
   2. Wählen Sie unter Verbundene Entität die Entität aus, die Sie mit der aktuellen Entität verbinden möchten.
   3. Geben Sie im Feld Alias optional einen temporären Aliasnamen für die zugehörige Entität ein.
   4. Wählen Sie unter Verbindungstyp den gewünschten Verbindungstyp aus.
   5. Definieren Sie die Join-Klausel, indem Sie die Felder aus jeder Entität auswählen.
   6. Wählen Sie Hinzufügen, um die Verknüpfung zu erstellen.

   Nach der Erstellung wird der Join im Bereich Joins angezeigt, sodass zusätzliche Felder in der Drop-down-Liste „Zurückzusetzende Felder“ verfügbar sind. Sie können mehrere Verknüpfungen hinzufügen, einschließlich verketteter Verknüpfungen zwischen Entitäten. Sie können auch nach Feldern verbundener Entitäten filtern und sortieren.

   Um einen Join zu löschen, klicken Sie auf das Papierkorbsymbol neben dem Join. Dadurch werden alle Felder aus dieser Verknüpfung entfernt und alle abhängigen Verknüpfungen oder Einschränkungen, die diese Felder verwenden, werden aufgehoben.
4. Fügen Sie unter Bedingungen Regeln hinzu, bearbeiten oder entfernen Sie sie, die die Ausgabe der Abfrage filtern. Sie können Regeln in Gruppen organisieren und mehrere Regeln mit `OR` Oder-Anweisungen `AND` verketten. Weitere Informationen zu den Operatoren, die Sie verwenden können, finden Sie unterBedingungsoperatoren für Datenaktionen und Beispiele.
5. Konfigurieren Sie unter Sortierung, wie die Abfrageergebnisse sortiert werden, indem Sie ein Attribut auswählen und eine auf- oder absteigende Reihenfolge wählen. Sie können die Sortierkonfiguration entfernen, indem Sie auf das Papierkorbsymbol neben der Sortierregel klicken.

6. Unter Transformationsergebnisse können Sie benutzerdefinierte Werte eingeben, JavaScript um Ergebnisse zu ändern oder zu formatieren, bevor sie angezeigt oder an Automatisierungen gesendet werden.
7. Sehen Sie sich in der Ausgabevorschau eine Vorschautabelle der Abfrageausgabe an, die auf den konfigurierten Feldern, Filtern, Sortierungen und JavaScript basiert.

Bedingungsoperatoren für Datenaktionen und Beispiele

Sie können Bedingungsoperatoren verwenden, um einen konfigurierten Ausdruckswert mit einer Entitätsspalte zu vergleichen, um eine Teilmenge von Datenbankobjekten zurückzugeben. Die Operatoren, die Sie verwenden können, hängen vom Datentyp der Spalte und vom Datenbanktyp ab, mit dem die Entität verbunden ist, z. B. Amazon Redshift, Amazon Aurora oder Amazon DynamoDB.

Die folgenden Bedingungsoperatoren können mit allen Datenbankdiensten verwendet werden:

- =und`!=`: Verfügbar für alle Datentypen (ausgenommen Primärschlüsselspalten).
- <=, >=<, und>=: Nur für numerische Spalten verfügbar.
- `IS NULL`und`IS NOT NULL`: Wird verwendet, um Spalten abzugleichen, die Null- oder Leerwerte enthalten. Nullwerte werden häufig in jeder Datenbank unterschiedlich interpretiert. In App Studio sucht der `NULL` Operator jedoch nach Datensätzen, die Nullwerte in der verbundenen Datenbanktabelle enthalten, und gibt sie zurück.

Die folgenden Bedingungsoperatoren können nur in Entitäten verwendet werden, die mit Datenbankdiensten verbunden sind, die sie unterstützen:

- `LIKE`und `NOT LIKE` (Redshift, Aurora): Wird für die Durchführung musterbasierter Abfragen in der verbundenen Datenbank verwendet. Der `LIKE` Operator bietet Flexibilität bei der Suchfunktion, da er Datensätze findet und zurückgibt, die den angegebenen Mustern entsprechen. Sie definieren die Muster mithilfe von Platzhalterzeichen, die einem beliebigen Zeichen oder einer beliebigen Zeichenfolge innerhalb des Musters entsprechen. Jedes Datenbankmanagementsystem hat einen eigenen Satz von Platzhalterzeichen, aber die beiden beliebtesten sind `%` die Darstellung einer beliebigen Anzahl von Zeichen (einschließlich 0) und `_` die Darstellung eines einzelnen Zeichens.
- `Contains`und `Not Contains` (DynamoDB): Wird verwendet, um eine Suche mit Berücksichtigung von Groß- und Kleinschreibung durchzuführen, um festzustellen, ob der angegebene Text in den Spaltenwerten gefunden wird.

- `Starts With`und `Not Starts With` (DynamoDB): Wird für eine Suche mit Berücksichtigung von Groß- und Kleinschreibung verwendet, um festzustellen, ob der angegebene Text am Anfang der Spaltenwerte gefunden wird.

Unterstützung für Bedingungsoperatoren durch die Datenbank

Die folgende Tabelle zeigt, welche Bedingungsoperatoren für Datenaktionen von jeder Datenbank unterstützt werden, die eine Verbindung zu App Studio herstellen kann.

	=, !=, <, >, <=, >=	WIE, NICHT WIE	Enthält, enthält nicht	Beginnt mit, beginnt nicht mit	IST NULL, IST NICHT NULL
DynamoDB	Ja	Nein	Ja	Ja	Ja
Aurora	Ja	Ja	Nein	Nein	Ja
Redshift	Ja	Ja	Nein	Nein	Ja

Beispiele für Bedingungen für Datenaktionen

Betrachten Sie die folgende Datenbanktabelle, die mehrere Elemente mit `hireDate` Feldern `namecity`, und enthält.

Name	city	Einstellungsdatum
Adam	Seattle	2025-03-01
Adrienne	Boston	05. März 2025
Bob	Albuquerque	06.03.2025
Carlos	Chicago	2025-03-10
Caroline	NULL	12.03.2025
Rita	Miami	15.03.2025

Erwägen Sie nun, Datenaktionen in App Studio zu erstellen, die das `name` Feld für Elemente zurückgeben, die bestimmten Bedingungen entsprechen. Die folgende Liste enthält Beispiele für Bedingungen und die Werte, die die Tabelle für jede Bedingung zurückgibt.

> **Note**
>
> Die Beispiele sind als SQL-Beispiele formatiert. Sie werden möglicherweise nicht wie in App Studio angezeigt, dienen aber dazu, das Verhalten der Operatoren zu veranschaulichen.

- `WHERE name LIKE 'Adam'`: Gibt zurück`Adam`.
- `WHERE name LIKE 'A%'`: Kehrt zurück `Adam` und`Adrienne`.
- `WHERE name NOT LIKE 'B_B'`: Gibt`Adam`, `Adrienne` `CarlosCaroline`, und zurück`Rita`.
- `WHERE contains(name, 'ita')`: Kehrt zurück`Rita`.
- `WHERE begins_with(name, 'Car')`: Kehrt zurück `Carlos` und`Caroline`.
- `WHERE city IS NULL`: Kehrt zurück`Caroline`.
- `WHERE hireDate < "2025-03-06"`: Kehrt zurück `Adam` und`Adrienne`.
- `WHERE hireDate >= DateTime.now().toISODate()`: Hinweis, der das aktuelle Datum `DateTime.now().toISODate()` zurückgibt. In einem Szenario, in dem das aktuelle Datum 2025-03-10 ist, gibt der Ausdruck, und zurück`Carlos`. `Caroline` `Rita`

> **Tip**
>
> Weitere Informationen zum Vergleichen von Datums- und Uhrzeitangaben in Ausdrücken finden Sie unter. Datum und Uhrzeit

Datenaktionen löschen

Gehen Sie wie folgt vor, um Datenaktionen aus einer App Studio-Entität zu löschen.

1. Navigieren Sie bei Bedarf zu der Entität, für die Sie Datenaktionen löschen möchten.
2. Wählen Sie die Registerkarte Datenaktionen aus.
3. Wählen Sie für jede Datenaktion, die Sie löschen möchten, das Dropdownmenü neben Bearbeiten und dann Löschen aus.
4. Wählen Sie im Dialogfeld die Option Bestätigen aus.

Beispieldaten hinzufügen oder löschen

Sie können Beispieldaten zu Entitäten in einer App Studio-Anwendung hinzufügen. Da Anwendungen erst mit externen Diensten kommunizieren, wenn sie veröffentlicht wurden, können Beispieldaten verwendet werden, um Ihre Anwendung und Entität in Vorschauumgebungen zu testen.

1. Navigieren Sie bei Bedarf zu der Entität, die Sie bearbeiten möchten.
2. Wählen Sie die Registerkarte Beispieldaten.
3. Um Beispieldaten zu generieren, wählen Sie Weitere Beispieldaten generieren aus.
4. Um Beispieldaten zu löschen, aktivieren Sie die Kontrollkästchen der Daten, die Sie löschen möchten, und drücken Sie die Lösch- oder Rücktaste. Wählen Sie Speichern, um die Änderungen zu speichern.

Fügen Sie verbundene Datenquellen hinzu oder bearbeiten Sie sie und ordnen Sie Felder zu

> **Tip**
>
> Sie können STRG+Z drücken, um die letzte Änderung an Ihrer Entität rückgängig zu machen.

1. Navigieren Sie bei Bedarf zu der Entität, die Sie bearbeiten möchten.
2. Wählen Sie die Registerkarte Verbindung, um die Verbindung zwischen der Entität und einer Datenquellentabelle anzuzeigen oder zu verwalten, in der Daten gespeichert werden, wenn Ihre Anwendung veröffentlicht wird. Sobald eine Datenquellentabelle verbunden ist, können Sie die Entitätsfelder den Spalten der Tabelle zuordnen.
3. Wählen Sie unter Connector den Konnektor aus, der eine Verbindung zur gewünschten Datenquellentabelle enthält. Weitere Informationen zu Konnektoren finden Sie unterConnect App Studio über Konnektoren mit anderen Diensten.
4. Wählen Sie unter Tabelle die Tabelle aus, die Sie als Datenquelle für die Entität verwenden möchten.
5. Die Tabelle zeigt die Felder der Entität und die Datenquellenspalte, der sie zugeordnet sind. Wählen Sie Automatische Zuordnung, um Ihre Entitätsfelder automatisch Ihren Datenquellenspalten zuzuordnen. Sie können Felder in der Tabelle auch manuell zuordnen, indem Sie die Datenquellenspalte in der Dropdownliste für jedes Entitätsfeld auswählen.

Eine Entität löschen

Gehen Sie wie folgt vor, um eine Entität aus einer App Studio-Anwendung zu löschen.

> **Note**
>
> Durch das Löschen einer Entität aus einer App Studio-App wird die verbundene Datenquellentabelle, einschließlich der entsprechenden DynamoDB-Tabelle mit verwalteten Entitäten, nicht gelöscht. Die Datenquellentabellen verbleiben im zugehörigen AWS Konto und müssen auf Wunsch aus dem entsprechenden Dienst gelöscht werden.

Um eine Entität zu löschen

1. Navigieren Sie bei Bedarf zu Ihrer Anwendung.
2. Wählen Sie den Tab Daten aus.
3. Wählen Sie im Menü Entitäten auf der linken Seite das Ellipsenmenü neben der Entität, die Sie löschen möchten, und wählen Sie Löschen.
4. Überprüfen Sie die Informationen im Dialogfeld, geben Sie die Entität ein **`confirm`** und wählen Sie Löschen, um sie zu löschen.

Verwaltete Datenentitäten in AWS App Studio

In der Regel konfigurieren Sie eine Entität in App Studio mit einer Verbindung zu einer externen Datenbanktabelle. Sie müssen jedes Entitätsfeld erstellen und einer Spalte in der verbundenen Datenbanktabelle zuordnen. Wenn Sie eine Änderung am Datenmodell vornehmen, müssen sowohl die externe Datenbanktabelle als auch die Entität aktualisiert werden, und die geänderten Felder müssen neu zugeordnet werden. Diese Methode ist zwar flexibel und ermöglicht die Verwendung verschiedener Arten von Datenquellen, erfordert jedoch mehr Vorausplanung und kontinuierliche Wartung.

Eine verwaltete Entität ist eine Art von Entität, für die App Studio den gesamten Datenspeicher- und Konfigurationsprozess für Sie verwaltet. Wenn Sie eine verwaltete Entität erstellen, wird eine entsprechende DynamoDB-Tabelle im zugehörigen AWS Konto erstellt. Dies gewährleistet eine sichere und transparente Datenverwaltung innerhalb. AWS Bei einer verwalteten Entität konfigurieren Sie das Schema der Entität in App Studio, und die entsprechende DynamoDB-Tabelle wird ebenfalls automatisch aktualisiert.

Verwaltete Entitäten in mehreren Anwendungen verwenden

Sobald Sie eine verwaltete Entität in einer App Studio-App erstellt haben, kann diese Entität in anderen App Studio-Apps verwendet werden. Dies ist hilfreich für die Konfiguration des Datenspeichers für Apps mit identischen Datenmodellen und Schemas, indem eine einzige zugrunde liegende Ressource zur Verwaltung bereitgestellt wird.

Wenn Sie eine verwaltete Entität in mehreren Anwendungen verwenden, müssen alle Schemaaktualisierungen der entsprechenden DynamoDB-Tabelle mit der ursprünglichen Anwendung vorgenommen werden, in der die verwaltete Entität erstellt wurde. Durch Schemaänderungen an der Entität in anderen Anwendungen wird die entsprechende DynamoDB-Tabelle nicht aktualisiert.

Einschränkungen verwalteter Entitäten

Einschränkungen bei der Aktualisierung von Primärschlüsseln: Sie können den Namen oder Typ des Primärschlüssels der Entität nicht ändern, nachdem sie erstellt wurde, da dies eine destruktive Änderung in DynamoDB darstellt und zum Verlust vorhandener Daten führen würde.

Umbenennen von Spalten: Wenn Sie eine Spalte in DynamoDB umbenennen, erstellen Sie tatsächlich eine neue Spalte, während die ursprüngliche Spalte die Originaldaten enthält. Die Originaldaten werden nicht automatisch in die neue Spalte kopiert oder aus der ursprünglichen Spalte gelöscht. Sie können verwaltete Entitätsfelder, den so genannten Systemnamen, umbenennen, aber Sie verlieren dadurch den Zugriff auf die ursprüngliche Spalte und ihre Daten. Bei der Umbenennung des Anzeigenamens gibt es keine Einschränkungen.

Ändern des Datentyps: DynamoDB bietet zwar die Flexibilität, Spaltendatentypen nach der Tabellenerstellung zu ändern, aber solche Änderungen können sich erheblich auf bestehende Daten sowie auf die Abfragelogik und Genauigkeit auswirken. Bei Änderungen des Datentyps müssen alle vorhandenen Daten so transformiert werden, dass sie dem neuen Format entsprechen, was bei großen, aktiven Tabellen komplex ist. Darüber hinaus können Datenaktionen zu unerwarteten Ergebnissen führen, bis die Datenmigration abgeschlossen ist. Sie können zwischen den Datentypen von Feldern wechseln, aber die vorhandenen Daten werden nicht auf den neuen Datentyp migriert.

Spalte sortieren: DynamoDB ermöglicht den Abruf sortierter Daten über Sortierschlüssel. Sortierschlüssel müssen zusammen mit dem Partitionsschlüssel als Teil zusammengesetzter Primärschlüssel definiert werden. Zu den Einschränkungen gehören ein obligatorischer Sortierschlüssel, eine begrenzte Sortierung innerhalb einer Partition und keine globale Sortierung über Partitionen hinweg. Eine sorgfältige Datenmodellierung von Sortierschlüsseln ist erforderlich, um heiße Partitionen zu vermeiden. Wir werden den Meilenstein Sorting for Preview nicht unterstützen.

Verknüpfungen: Verknüpfungen werden in DynamoDB nicht unterstützt. Tabellen sind konstruktionsbedingt denormalisiert, um teure Join-Operationen zu vermeiden. Um one-to-many Beziehungen zu modellieren, enthält die untergeordnete Tabelle ein Attribut, das auf den Primärschlüssel der übergeordneten Tabelle verweist. Bei Datenabfragen mit mehreren Tabellen werden Elemente aus der übergeordneten Tabelle nachgeschlagen, um Details abzurufen. Wir werden im Rahmen des Vorschau-Meilensteins keine systemeigenen Verknüpfungen für verwaltete Entitäten unterstützen. Um dieses Problem zu umgehen, werden wir einen Automatisierungsschritt einführen, der eine Datenzusammenführung von zwei Entitäten durchführen kann. Dies wird einer Suche auf einer Ebene sehr ähnlich sein. Wir werden den Meilenstein Sorting for Preview nicht unterstützen.

Env Stage: Wir werden die Veröffentlichung zu Testzwecken zulassen, aber in beiden Umgebungen denselben Managed Store verwenden

Seiten- und Automatisierungsparameter

Parameter sind eine leistungsstarke Funktion in AWS App Studio, mit der dynamische Werte zwischen verschiedenen Komponenten, Seiten und Automatisierungen innerhalb Ihrer Anwendung übergeben werden. Mithilfe von Parametern können Sie flexible und kontextsensitive Erlebnisse schaffen, wodurch Ihre Anwendungen reaktionsschneller und personalisierter werden. Dieser Artikel behandelt zwei Arten von Parametern: Seitenparameter und Automatisierungsparameter.

Themen

- Seitenparameter
- Automatisierungsparameter

Seitenparameter

Seitenparameter sind eine Möglichkeit, Informationen zwischen Seiten zu senden. Sie werden häufig verwendet, wenn innerhalb einer App Studio-App von einer Seite zur anderen navigiert wird, um den Kontext beizubehalten oder Daten zu übergeben. Seitenparameter bestehen normalerweise aus einem Namen und einem Wert.

Anwendungsfälle für Seitenparameter

Seitenparameter werden für die Übertragung von Daten zwischen verschiedenen Seiten und Komponenten in Ihren App Studio-Anwendungen verwendet. Sie sind besonders hilfreich für die folgenden Anwendungsfälle:

1. Suchen und Filtern: Wenn Nutzer auf der Startseite Ihrer App suchen, können die Suchbegriffe als Parameter an die Ergebnisseite übergeben werden, sodass nur die relevanten gefilterten Elemente angezeigt werden. Wenn ein Benutzer beispielsweise nach sucht*`noise-cancelling headphones`*, *`noise-cancelling headphones`* kann der Parameter mit dem Wert an die Seite mit der Produktliste übergeben werden.
2. Artikeldetails anzeigen: Wenn ein Benutzer auf ein Angebot klickt, z. B. auf ein Produkt, kann die eindeutige Kennung dieses Artikels als Parameter an die Detailseite übergeben werden. Auf diese Weise können auf der Detailseite alle Informationen zu dem bestimmten Artikel angezeigt werden. Wenn ein Benutzer beispielsweise auf ein Kopfhörerprodukt klickt, wird die eindeutige ID des Produkts als Parameter an die Produktdetailseite übergeben.
3. Weitergabe des Benutzerkontextes bei der Seitennavigation: Wenn Benutzer zwischen den Seiten navigieren, können Parameter wichtige Kontexte wie den Standort des Benutzers, bevorzugte Produktkategorien, Warenkorbinhalte und andere Einstellungen weitergeben. Wenn ein Nutzer beispielsweise verschiedene Produktkategorien in Ihrer App durchsucht, werden sein Standort und seine bevorzugten Kategorien als Parameter gespeichert, was für ein personalisiertes und einheitliches Erlebnis sorgt.
4. Deep-Links: Verwenden Sie Seitenparameter, um einen Link zu einer bestimmten Seite innerhalb der App zu teilen oder mit einem Lesezeichen zu versehen.
5. Datenaktionen: Sie können Datenaktionen erstellen, die Parameterwerte akzeptieren, um Ihre Datenquellen auf der Grundlage der übergebenen Parameter zu filtern und abzufragen. Auf der Produktlistenseite können Sie beispielsweise eine Datenaktion erstellen, die `category` Parameter akzeptiert, um die entsprechenden Produkte abzurufen.

Überlegungen zur Sicherheit von Seitenparametern

Seitenparameter bieten zwar eine leistungsstarke Möglichkeit, Daten zwischen Seiten zu übertragen, sollten jedoch mit Vorsicht verwendet werden, da sie möglicherweise vertrauliche Informationen preisgeben können, wenn sie nicht ordnungsgemäß verwendet werden. Hier sind einige wichtige Sicherheitsüberlegungen, die Sie beachten sollten:

1. Vermeiden Sie die Offenlegung sensibler Daten in URLs
   a. Risiko: URLs, einschließlich der Parameter für Datenaktionen, sind häufig in Serverprotokollen, im Browserverlauf und an anderen Stellen sichtbar. Daher ist es wichtig, zu vermeiden, dass sensible Daten wie Benutzeranmeldedaten, personenbezogene Daten (PII) oder andere vertrauliche Daten in Seitenparameterwerten offengelegt werden.

b. Schadensbegrenzung: Erwägen Sie die Verwendung von Kennungen, die den sensiblen Daten sicher zugeordnet werden können. Anstatt beispielsweise den Namen oder die E-Mail-Adresse eines Benutzers als Parameter zu übergeben, könnten Sie eine zufällige eindeutige Kennung übergeben, mit der der Name oder die E-Mail-Adresse des Benutzers abgerufen werden kann.

Automatisierungsparameter

Automatisierungsparameter sind eine leistungsstarke Funktion in App Studio, mit der flexible und wiederverwendbare Automatisierungen erstellt werden können, indem dynamische Werte aus verschiedenen Quellen übergeben werden, z. B. aus der Benutzeroberfläche, anderen Automatisierungen oder Datenaktionen. Sie dienen als Platzhalter, die bei der Ausführung der Automatisierung durch tatsächliche Werte ersetzt werden, sodass Sie dieselbe Automatisierung jedes Mal mit unterschiedlichen Eingaben verwenden können.

Innerhalb einer Automatisierung haben Parameter eindeutige Namen, und Sie können auf den Wert eines Parameters verweisen, indem Sie die Variable params gefolgt vom Namen des Parameters verwenden, z. B. `{{params.customerId}}`

Dieser Artikel bietet ein tiefes Verständnis der Automatisierungsparameter, einschließlich ihrer grundlegenden Konzepte, Verwendung und bewährten Methoden.

Vorteile von Automatisierungsparametern

Automatisierungsparameter bieten mehrere Vorteile, darunter die folgende Liste:

1. Wiederverwendbarkeit: Mithilfe von Parametern können Sie wiederverwendbare Automatisierungen erstellen, die mit unterschiedlichen Eingabewerten angepasst werden können, sodass Sie dieselbe Automatisierungslogik mit unterschiedlichen Eingaben wiederverwenden können.
2. Flexibilität: Anstatt Werte fest in einer Automatisierung zu codieren, können Sie Parameter definieren und bei Bedarf unterschiedliche Werte angeben, wodurch Ihre Automatisierungen dynamischer und anpassungsfähiger werden.
3. Trennung von Belangen: Parameter helfen dabei, die Automatisierungslogik von den verwendeten spezifischen Werten zu trennen, was die Organisation und Wartbarkeit des Codes fördert.
4. Validierung: Jeder Parameter hat einen Datentyp, z. B. eine Zeichenfolge, eine Zahl oder einen booleschen Wert, der zur Laufzeit validiert wird. Dadurch wird sichergestellt, dass Anfragen mit

falschen Datentypen abgelehnt werden, ohne dass ein benutzerdefinierter Validierungscode erforderlich ist.

5. Optionale und erforderliche Parameter: Sie können Automatisierungsparameter als optional oder erforderlich festlegen. Erforderliche Parameter müssen bei der Ausführung der Automatisierung angegeben werden, während optionale Parameter Standardwerte haben oder weggelassen werden können. Diese Flexibilität ermöglicht es Ihnen, vielseitigere Automatisierungen zu erstellen, die auf der Grundlage der bereitgestellten Parameter unterschiedliche Szenarien bewältigen können.

Szenarien und Anwendungsfälle

Szenario: Produktdetails werden abgerufen

Stellen Sie sich vor, Sie haben eine Automatisierung, die Produktdetails auf der Grundlage einer Produkt-ID aus einer Datenbank abruft. Diese Automatisierung könnte einen Parameter haben, der aufgerufen wird`productId`.

Der `productId` Parameter fungiert als Platzhalter, den Sie beim Ausführen der Automatisierung mit dem tatsächlichen Produkt-ID-Wert ausfüllen können. Anstatt eine bestimmte Produkt-ID fest in der Automatisierung zu codieren, können Sie den `productId` Parameter definieren und bei jeder Ausführung der Automatisierung unterschiedliche Produkt-ID-Werte übergeben.

Sie könnten diese Automatisierung von der Datenquelle einer Komponente aus aufrufen und die ID des ausgewählten Produkts als `productId` Parameter mit der Syntax in doppelten geschweiften Klammern übergeben:. `{{ui.productsTable.selectedRow.id}}` Auf diese Weise ruft die Automatisierung, wenn ein Benutzer ein Produkt aus einer Tabelle (`ui.productsTable`) auswählt, die Details für das ausgewählte Produkt ab, indem sie die ID der ausgewählten Zeile als `productId` Parameter übergibt.

Alternativ können Sie diese Automatisierung von einer anderen Automatisierung aus aufrufen, die eine Produktliste durchläuft und die Details für jedes Produkt abruft, indem die Produkt-ID als Parameter übergeben wird. `productId` In diesem Szenario würde der `productId` Parameterwert in jeder Iteration der `{{product.id}}` Schleife dynamisch aus dem Ausdruck bereitgestellt.

Durch die Verwendung des `productId` Parameters und der Syntax mit doppelten geschweiften Klammern können Sie diese Automatisierung flexibler und wiederverwendbarer machen. Anstatt separate Automatisierungen für jedes Produkt zu erstellen, können Sie eine einzige Automatisierung

verwenden, mit der Details für jedes Produkt abgerufen werden können, indem Sie einfach die entsprechende Produkt-ID als Parameterwert aus verschiedenen Quellen angeben, z. B. aus UI-Komponenten oder anderen Automatisierungen.

Szenario: Umgang mit optionalen Parametern mit Fallback-Werten

Stellen wir uns ein Szenario vor, in dem Sie eine Entität „Aufgabe“ mit einer erforderlichen Spalte „Besitzer“ haben, aber Sie möchten, dass dieses Feld in der Automatisierung optional ist und Sie einen Fallback-Wert angeben, falls der Besitzer nicht ausgewählt ist.

1. Erstellen Sie eine Automatisierung mit einem Parameter namens`0wner`, der dem `0wner` Feld der `Task` Entität zugeordnet ist.
2. Da das `0wner` Feld in der Entität erforderlich ist, wird der `0wner` Parameter mit der erforderlichen Einstellung synchronisiert.
3. Um den `0wner` Parameter in der Automatisierung optional zu machen, schalten Sie die `required` Einstellung für diesen Parameter aus.
4. In Ihrer Automatisierungslogik können Sie einen Ausdruck wie `{{params.0wner || currentUser.userId}}` verwenden. Dieser Ausdruck prüft, ob der `0wner` Parameter angegeben ist. Wenn er nicht angegeben wird, wird auf die ID des aktuellen Benutzers als Eigentümer zurückgegriffen.
5. Auf diese Weise weist die Automatisierung automatisch den aktuellen Benutzer als Eigentümer für die Aufgabe zu, wenn der Benutzer keinen Besitzer in einem Formular oder einer Komponente auswählt.

Indem Sie die `required` Einstellung für den `0wner` Parameter umschalten und einen Fallback-Ausdruck verwenden, können Sie ihn von der Entitätsfeldanforderung entkoppeln, ihn in der Automatisierung optional machen und einen Standardwert angeben, wenn der Parameter nicht bereitgestellt wird.

Definieren von Automatisierungsparametertypen

Indem Sie Parametertypen verwenden, um Datentypen zu spezifizieren und Anforderungen festzulegen, können Sie die Eingaben für Ihre Automatisierungen steuern. Auf diese Weise können Sie sicherstellen, dass Ihre Automatisierungen zuverlässig mit den erwarteten Eingaben ausgeführt werden.

Synchronisieren von Typen aus einer Entität

Die dynamische Synchronisation von Parametertypen und Anforderungen aus Entitätsfelddefinitionen optimiert die Gebäudeautomatisierung, die mit Entitätsdaten interagiert, und stellt sicher, dass der Parameter immer den neuesten Feldtyp und die neuesten Anforderungen der Entitätsfelder widerspiegelt.

Das folgende Verfahren beschreibt die allgemeinen Schritte zum Synchronisieren von Parametertypen aus einer Entität:

1. Erstellen Sie eine Entität mit typisierten Feldern (z. B. Boolean, Number usw.) und markieren Sie Felder nach Bedarf.
2. Erstellen Sie eine neue Automatisierung.
3. Fügen Sie der Automatisierung Parameter hinzu und wählen Sie bei der Auswahl des Typs das Entitätsfeld aus, mit dem Sie synchronisieren möchten. Der Datentyp und die erforderliche Einstellung werden automatisch aus dem zugewiesenen Entitätsfeld synchronisiert
4. Bei Bedarf können Sie die Einstellung „erforderlich“ überschreiben, indem Sie sie on/off für jeden Parameter umschalten. Das bedeutet, dass der erforderliche Status nicht mit dem Entitätsfeld synchronisiert wird, ansonsten aber synchronisiert bleibt.

Typen manuell definieren

Sie können Parametertypen auch manuell definieren, ohne von einer Entität aus zu synchronisieren

Durch die Definition benutzerdefinierter Parametertypen können Sie Automatisierungen erstellen, die bestimmte Eingabetypen akzeptieren und optionale oder erforderliche Parameter nach Bedarf verarbeiten, ohne sich auf Entitätsfeldzuordnungen verlassen zu müssen.

1. Erstellen Sie eine Entität mit typisierten Feldern (z. B. Boolean, Number usw.) und markieren Sie Felder nach Bedarf.
2. Erstellen Sie eine neue Automatisierung.
3. Fügen Sie der Automatisierung Parameter hinzu und wählen Sie bei der Auswahl des Typs den gewünschten Typ aus.

Konfiguration dynamischer Werte, die an Automatisierungsparameter übergeben werden sollen

Sobald Sie Parameter für eine Automatisierung definiert haben, können Sie ihnen beim Aufrufen der Automatisierung Werte übergeben. Sie können Parameterwerte auf zwei Arten übergeben:

1. Komponententrigger: Wenn Sie die Automatisierung über einen Komponententrigger aufrufen, z. B. durch einen Klick auf eine Schaltfläche, können Sie JavaScript Ausdrücke verwenden, um Werte aus dem Komponentenkontext zu übergeben. Wenn Sie beispielsweise ein Texteingabefeld mit dem Namen haben`emailInput`, können Sie seinen Wert mit dem folgenden Ausdruck an den E-Mail-Parameter übergeben:`ui.emailInput.value`.
2. Andere Automatisierungen: Wenn Sie die Automatisierung von einer anderen Automatisierung aus aufrufen, können Sie JavaScript Ausdrücke verwenden, um Werte aus dem Automatisierungskontext zu übergeben. Sie können beispielsweise den Wert eines anderen Parameters oder das Ergebnis eines vorherigen Aktionsschritts übergeben.

Geben Sie Sicherheit ein

Durch die Definition von Parametern mit bestimmten Datentypen wie String, Number oder Boolean können Sie sicherstellen, dass die an Ihre Automatisierung übergebenen Werte dem erwarteten Typ entsprechen.

> **Note**
>
> In App Studio sind Datum (e) ISO-Zeichenkettendaten, und diese werden ebenfalls validiert.

Diese Typsicherheit trägt dazu bei, Typkonflikte zu vermeiden, die zu Fehlern oder unerwartetem Verhalten in Ihrer Automatisierungslogik führen können. Wenn Sie beispielsweise einen Parameter als a definieren`Number`, können Sie sicher sein, dass jeder an diesen Parameter übergebene Wert eine Zahl ist, und Sie müssen in Ihrer Automatisierung keine zusätzlichen Typprüfungen oder Konvertierungen durchführen.

Validierung

Sie können Ihren Parametern Validierungsregeln hinzufügen, um sicherzustellen, dass die an Ihre Automatisierung übergebenen Werte bestimmten Kriterien entsprechen.

App Studio bietet zwar keine integrierten Validierungseinstellungen für Parameter, Sie können jedoch benutzerdefinierte Validierungen implementieren, indem Sie Ihrer Automatisierung eine JavaScript Aktion hinzufügen, die einen Fehler auslöst, wenn bestimmte Einschränkungen verletzt werden.

Für Entitätsfelder wird eine Teilmenge von Validierungsregeln, wie z. B. minimum/maximum Werte, unterstützt. Diese werden jedoch nicht auf Automatisierungsebene validiert, sondern nur auf der Datenebene, wenn Create/Update/Delete Datensatzaktionen ausgeführt werden.

Bewährte Methoden für Automatisierungsparameter

Um sicherzustellen, dass Ihre Automatisierungsparameter gut konzipiert, wartungsfreundlich und benutzerfreundlich sind, befolgen Sie diese bewährten Methoden:

1. Verwenden Sie aussagekräftige Parameternamen: Wählen Sie Parameternamen, die den Zweck oder den Kontext des Parameters klar beschreiben.
2. Geben Sie Parameterbeschreibungen an: Nutzen Sie das Feld Beschreibung, wenn Sie Parameter definieren, um deren Zweck, Einschränkungen und Erwartungen zu erläutern. Diese Beschreibungen werden in den JSDoc Kommentaren angezeigt, wenn auf den Parameter verwiesen wird, sowie in allen Benutzeroberflächen, in denen Benutzer beim Aufrufen der Automatisierung Werte für die Parameter angeben müssen.
3. Verwenden Sie geeignete Datentypen: Überlegen Sie sich sorgfältig, welchen Datentyp die einzelnen Parameter auf der Grundlage der erwarteten Eingabewerte haben, z. B.: Zeichenfolge, Zahl, Boolean, Objekt.
4. Überprüfen Sie die Parameterwerte: Implementieren Sie geeignete Validierungsprüfungen innerhalb Ihrer Automatisierung, um sicherzustellen, dass die Parameterwerte bestimmten Anforderungen entsprechen, bevor Sie mit weiteren Aktionen fortfahren.
5. Verwenden Sie Ausweichwerte oder Standardwerte: App Studio unterstützt derzeit zwar nicht die Festlegung von Standardwerten für Parameter, Sie können jedoch Fallback- oder Standardwerte implementieren, wenn Sie die Parameter in Ihrer Automatisierungslogik verwenden. Sie können beispielsweise einen Ausdruck wie verwenden, `{{ params.param1 || "default value" }}` um einen Standardwert bereitzustellen, wenn der `param1` Parameter nicht bereitgestellt wird oder einen falschen Wert hat.
6. Beibehaltung der Parameterkonsistenz: Wenn Sie mehrere Automatisierungen haben, die ähnliche Parameter erfordern, versuchen Sie, die Konsistenz der Parameternamen und Datentypen für alle Automatisierungen aufrechtzuerhalten.

7. Verwendung von Dokumentparametern: Sorgen Sie für eine übersichtliche Dokumentation Ihrer Automatisierungen, einschließlich Beschreibungen der einzelnen Parameter, ihres Zwecks, der erwarteten Werte und aller relevanten Beispiele oder Grenzfälle.
8. Regelmäßige Überprüfung und Refaktorierung: Überprüfen Sie Ihre Automatisierungen und ihre Parameter regelmäßig und überarbeiten oder konsolidieren Sie die Parameter nach Bedarf, um die Übersichtlichkeit, Wartbarkeit und Wiederverwendbarkeit zu verbessern.
9. Beschränken Sie die Anzahl der Parameter: Parameter bieten zwar Flexibilität, aber zu viele Parameter können eine Automatisierung komplex und schwierig zu handhaben machen. Versuchen Sie, ein Gleichgewicht zwischen Flexibilität und Einfachheit zu finden, indem Sie die Anzahl der Parameter auf das Notwendige beschränken.
10. Erwägen Sie die Gruppierung von Parametern: Wenn Sie mehrere verwandte Parameter definieren müssen, sollten Sie in Erwägung ziehen, sie zu einem einzigen `Object` Parameter zu gruppieren.
11. Separate Bedenken: Vermeiden Sie es, einen einzelnen Parameter für mehrere Zwecke zu verwenden oder Werte, die nichts miteinander zu tun haben, zu einem einzigen Parameter zu kombinieren. Jeder Parameter sollte für ein bestimmtes Anliegen oder eine bestimmte Information stehen.
12. Verwenden Sie Parameteraliase: Wenn Sie Parameter mit langen oder komplexen Namen haben, sollten Sie die Verwendung von Aliasen oder Kurzversionen innerhalb der Automatisierungslogik erwägen, um die Lesbarkeit und Wartbarkeit zu verbessern.

Wenn Sie diese bewährten Methoden befolgen, können Sie sicherstellen, dass Ihre Automatisierungsparameter gut konzipiert, wartungsfreundlich und benutzerfreundlich sind, was letztendlich die Gesamtqualität und Effizienz Ihrer Automatisierungen verbessert.

Verwenden JavaScript , um Ausdrücke in App Studio zu schreiben

In AWS App Studio können Sie JavaScript Ausdrücke verwenden, um das Verhalten und das Erscheinungsbild Ihrer Anwendungen dynamisch zu steuern. Einzeilige JavaScript Ausdrücke werden in doppelten geschweiften Klammern geschrieben und können in verschiedenen Kontexten wie Automatisierungen, UI-Komponenten und Datenabfragen verwendet werden. `{{ }}` Diese Ausdrücke werden zur Laufzeit ausgewertet und können verwendet werden, um Berechnungen durchzuführen, Daten zu manipulieren und die Anwendungslogik zu steuern.

App Studio bietet native Unterstützung für drei JavaScript Open-Source-Bibliotheken: Luxon, UUID, Lodash sowie SDK-Integrationen zur Erkennung von JavaScript Syntax- und Typprüfungsfehlern in den Konfigurationen Ihrer App.

> **Important**
>
> App Studio unterstützt nicht die Verwendung von Bibliotheken von Drittanbietern oder benutzerdefinierten Bibliotheken. JavaScript

Basissyntax

JavaScript Ausdrücke können Variablen, Literale, Operatoren und Funktionsaufrufen enthalten. Ausdrücke werden häufig verwendet, um Berechnungen durchzuführen oder Bedingungen auszuwerten.

Im Folgenden sind einige Beispiele aufgeführt:

- `{{ 2 + 3 }}`wird mit 5 bewertet.
- `{{ "Hello, " + "World!" }}`wird mit „Hello, World!“ bewertet.
- `{{ Math.max(5, 10) }}`wird mit 10 bewertet.
- `{{ Math.random() * 10 }}`gibt eine Zufallszahl (mit Dezimalzahlen) zwischen [0-10) zurück.

Interpolation

Sie können sie auch verwenden JavaScript , um dynamische Werte innerhalb von statischem Text zu interpolieren. Dies wird erreicht, indem der JavaScript Ausdruck in doppelte geschweifte Klammern eingeschlossen wird, wie im folgenden Beispiel:

```
Hello {{ currentUser.firstName }}, welcome to App Studio!
```

In diesem Beispiel `currentUser.firstName` handelt es sich um einen JavaScript Ausdruck, der den Vornamen des aktuellen Benutzers abruft, der dann dynamisch in die Begrüßungsnachricht eingefügt wird.

Verkettung

Sie können Zeichenketten und Variablen mit dem + Operator in verketten JavaScript, wie im folgenden Beispiel gezeigt.

```
{{ currentRow.FirstName + " " + currentRow.LastName }}
```

Dieser Ausdruck kombiniert die Werte von `currentRow.FirstName` und `currentRow.LastName` mit einem Leerzeichen dazwischen, was den vollständigen Namen der aktuellen Zeile ergibt. Wenn beispielsweise is und `currentRow.FirstName` `currentRow.LastName` is `JohnDoe`, dann würde der Ausdruck wie folgt aufgelöst werden`John Doe`.

Datum und Uhrzeit

JavaScript stellt verschiedene Funktionen und Objekte für die Arbeit mit Datums- und Uhrzeitangaben bereit. Zum Beispiel:

- `{{ new Date().toLocaleDateString() }}`: Gibt das aktuelle Datum in einem lokalisierten Format zurück.
- `{{ DateTime.now().toISODate() }}`: Gibt das aktuelle Datum im YYYY-MM-DD Format zurück, das in der Date-Komponente verwendet werden kann.

Vergleich von Datum und Uhrzeit

Verwenden Sie Operatoren wie=,,>, oder <>=, <= um Datums- oder Uhrzeitwerte zu vergleichen. Zum Beispiel:

- `{{ui.timeInput.value > "10:00 AM"}}`: Prüft, ob die Uhrzeit nach 10:00 Uhr liegt.
- `{{ui.timeInput.value <= "5:00 PM"}}`: Prüft, ob die Uhrzeit um oder vor 17:00 Uhr ist.
- `{{ui.timeInput.value > DateTime.now().toISOTime()}}`: Prüft, ob die Uhrzeit nach der aktuellen Uhrzeit liegt.
- `{{ui.dateInput.value > DateTime.now().toISODate()}}`: Prüft, ob das Datum vor dem aktuellen Datum liegt.
- `{{ DateTime.fromISO(ui.dateInput.value).diff(DateTime.now(), "days").days >= 5 }}`: Prüft, ob das Datum mindestens 5 Tage vom aktuellen Datum entfernt ist.

Codeblöcke

Neben Ausdrücken können Sie auch mehrzeilige JavaScript Codeblöcke schreiben. Im Gegensatz zu Ausdrücken benötigen Codeblöcke keine geschweiften Klammern. Stattdessen können Sie Ihren JavaScript Code direkt im Codeblock-Editor schreiben.

> **Note**
>
> Während Ausdrücke ausgewertet und ihre Werte angezeigt werden, werden Codeblöcke ausgeführt und ihre Ausgabe (falls vorhanden) angezeigt.

Globale Variablen und Funktionen

App Studio bietet Zugriff auf bestimmte globale Variablen und Funktionen, die in Ihren JavaScript Ausdrücken und Codeblöcken verwendet werden können. `currentUser`Ist beispielsweise eine globale Variable, die den aktuell angemeldeten Benutzer darstellt, und Sie können auf Eigenschaften zugreifen, `currentUser.role` um beispielsweise die Rolle des Benutzers abzurufen.

Referenzieren oder Aktualisieren von Werten von UI-Komponenten

Sie können Ausdrücke in Komponenten und Automatisierungsaktionen verwenden, um Werte von UI-Komponenten sowohl zu referenzieren als auch zu aktualisieren. Durch programmgesteuertes Referenzieren und Aktualisieren von Komponentenwerten können Sie dynamische und interaktive Benutzeroberflächen erstellen, die auf Benutzereingaben und Datenänderungen reagieren.

Referenzieren von UI-Komponentenwerten

Sie können interaktive und datengesteuerte Anwendungen erstellen, indem Sie dynamisches Verhalten implementieren, indem Sie auf Werte aus UI-Komponenten zugreifen.

Sie können auf derselben Seite auf Werte und Eigenschaften von UI-Komponenten zugreifen, indem Sie den `ui` Namespace in Ausdrücken verwenden. Indem Sie auf den Namen einer Komponente verweisen, können Sie ihren Wert abrufen oder Operationen basierend auf ihrem Status ausführen.

> **Note**
>
> Im `ui` Namespace werden nur Komponenten auf der aktuellen Seite angezeigt, da Komponenten auf ihre jeweiligen Seiten beschränkt sind.

Die grundlegende Syntax für den Verweis auf Komponenten in einer App Studio-App lautet:. `{{ui.componentName}}`

Die folgende Liste enthält Beispiele für die Verwendung des `ui` Namespace für den Zugriff auf Werte von UI-Komponenten:

- `{{ui.textInputName.value}}`: Stellt den Wert einer Texteingabekomponente mit dem Namen *textInputName* dar.
- `{{ui.formName.isValid}}`: Überprüfen Sie anhand der von Ihnen angegebenen Validierungskriterien, ob alle Felder im genannten Formular gültig *formName* sind.
- `{{ui.tableName.currentRow.columnName}}`: Stellt den Wert einer bestimmten Spalte in der aktuellen Zeile einer benannten Tabellenkomponente dar*tableName*.
- `{{ui.tableName.selectedRowData.fieldName}}`: Stellt den Wert des angegebenen Felds aus der ausgewählten Zeile in einer Tabellenkomponente mit dem Namen dar*tableName*. Sie können dann einen Feldnamen wie `ID` (`{{ui.tableName.selectedRowData.ID}}`) anhängen, um auf den Wert dieses Felds aus der ausgewählten Zeile zu verweisen.

Die folgende Liste enthält spezifischere Beispiele für die Referenzierung von Komponentenwerten:

- `{{ui.inputText1.value.trim().length > 0}}`: Prüfen Sie, ob der Wert der *inputText1* Komponente nach dem Abschneiden aller führenden oder nachfolgenden Leerzeichen eine nicht leere Zeichenfolge enthält. Dies kann nützlich sein, um Benutzereingaben oder enabling/disabling andere Komponenten auf der Grundlage des Werts des Eingabe-Textfeldes zu validieren.
- `{{ui.multiSelect1.value.join(", ")}}`: Bei einer Komponente mit Mehrfachauswahl mit dem Namen *multiSelect1* konvertiert dieser Ausdruck das Array der ausgewählten Optionswerte in eine durch Kommas getrennte Zeichenfolge. Dies kann hilfreich sein, um die ausgewählten Optionen in einem benutzerfreundlichen Format anzuzeigen oder die Auswahl an eine andere Komponente oder Automatisierung zu übergeben.
- `{{ui.multiSelect1.value.includes("option1")}}`: Dieser Ausdruck prüft, ob der *option1* Wert in der Reihe der ausgewählten Optionen für die *multiSelect1* Komponente enthalten ist. Er gibt true zurück, wenn ausgewählt *option1* ist, andernfalls false. Dies kann nützlich sein, um Komponenten bedingt zu rendern oder Aktionen auf der Grundlage bestimmter Optionsauswahlen auszuführen.
- `{{ui.s3Upload1.files.length > 0}}`: Für eine Amazon S3 S3-Datei-Upload-Komponente mit dem Namen überprüft dieser Ausdruck*s3Upload1*, ob Dateien hochgeladen wurden, indem er

die Länge des Dateiarrays überprüft. Dies kann für enabling/disabling andere Komponenten oder Aktionen nützlich sein, je nachdem, ob Dateien hochgeladen wurden.

- `{{ui.s3Upload1.files.filter(file => file.type === "image/png").length}}`: Dieser Ausdruck filtert die Liste der hochgeladenen Dateien in der `s3Upload1` Komponente so, dass sie nur PNG-Bilddateien enthält, und gibt die Anzahl dieser Dateien zurück. Dies kann hilfreich sein, um Informationen über die Typen der hochgeladenen Dateien zu überprüfen oder anzuzeigen.

Aktualisierung der Werte von UI-Komponenten

Um den Wert einer Komponente zu aktualisieren oder zu bearbeiten, verwenden Sie den `RunComponentAction` innerhalb einer Automatisierung. Hier ist ein Beispiel für die Syntax, die Sie verwenden können, um den Wert einer Texteingabekomponente zu aktualisieren, die `myInput` mit der `RunComponentAction` Aktion benannt wurde:

```
RunComponentAction(ui.myInput, "setValue", "New Value")
```

In diesem Beispiel ruft der `RunComponentAction` Schritt die `setValue` Aktion für die `myInput` Komponente auf und übergibt den neuen Wert`New Value`.

Mit Tabellendaten arbeiten

Sie können auf Tabellendaten und Werte zugreifen, um Operationen auszuführen. Sie können die folgenden Ausdrücke verwenden, um auf Tabellendaten zuzugreifen:

- `currentRow`: Wird verwendet, um auf Tabellendaten aus der aktuellen Zeile in der Tabelle zuzugreifen. Beispielsweise das Festlegen des Namens einer Tabellenaktion, das Senden eines Werts aus der Zeile an eine Automatisierung, die von einer Aktion aus gestartet wird, oder das Verwenden von Werten aus vorhandenen Spalten in einer Tabelle, um eine neue Spalte zu erstellen.
- `ui.tableName.selectedRow`und `ui.tableName.selectedRowData` werden beide für den Zugriff auf Tabellendaten von anderen Komponenten auf der Seite verwendet. Beispielsweise das Festlegen des Namens einer Schaltfläche außerhalb der Tabelle auf der Grundlage der ausgewählten Zeile. Die zurückgegebenen Werte sind dieselben, aber die Unterschiede zwischen `selectedRow` und `selectedRowData` sind wie folgt:
  - `selectedRow`: Dieser Namespace enthält den Namen, der in der Spaltenüberschrift für jedes Feld angezeigt wird. Sie sollten ihn verwenden`selectedRow`, wenn Sie auf einen Wert aus

einer sichtbaren Spalte in der Tabelle verweisen. Zum Beispiel, wenn Sie eine benutzerdefinierte oder berechnete Spalte in Ihrer Tabelle haben, die nicht als Feld in der Entität existiert.

- `selectedRowData`: Dieser Namespace umfasst die Felder in der Entität, die als Quelle für die Tabelle verwendet werden. Sie sollten ihn verwenden`selectedRowData`, um auf einen Wert aus der Entität zu verweisen, der in der Tabelle nicht sichtbar ist, aber für andere Komponenten oder Automatisierungen in Ihrer App nützlich ist.

Die folgende Liste enthält Beispiele für den Zugriff auf Tabellendaten in Ausdrücken:

- `{{ui.`*`tableName`*`.selectedRow.`*`columnNameWithNoSpace`*`}}`: Gibt den Wert der *`columnNameWithNoSpace`* Spalte aus der ausgewählten Zeile in der Tabelle zurück.
- `{{ui.`*`tableName`*`.selectedRow.['`*`Column Name With Space`*`']}}`: Gibt den Wert der *`Column Name With Space`* Spalte aus der ausgewählten Zeile in der Tabelle zurück.
- `{{ui.`*`tableName`*`.selectedRowData.`*`fieldName`*`}}`: Gibt den Wert des *`fieldName`* Entitätsfeldes aus der ausgewählten Zeile in der Tabelle zurück.
- `{{ui.`*`tableName`*`.selectedRows[0].`*`columnMappingName`*`}}`: Referenziert den Spaltennamen der ausgewählten Zeile aus anderen Komponenten oder Ausdrücken auf derselben Seite.
- `{{currentRow.`*`firstName`*` + ' ' + currentRow.`*`lastNamecolumnMapping`*`}}`: Verketten Sie Werte aus mehreren Spalten, um eine neue Spalte in einer Tabelle zu erstellen.
- `{{ { "Blocked": "#", "Delayed": "#", "On track": "#" } [currentRow.`*`statuscolumnMapping`*`] + " " + currentRow.`*`statuscolumnMapping`*`}}`: Passen Sie den Anzeigewert eines Felds in einer Tabelle basierend auf dem gespeicherten Statuswert an.
- `{{currentRow.`*`colName`*`}}`, `{{currentRow["`*`First Name`*`"]}}{{currentRow}}`, oder`{{ui.`*`tableName`*`.selectedRows[0]}}`: Übergibt den Kontext der referenzierten Zeile innerhalb einer Zeilenaktion.

Zugreifen auf Automatisierungen

Sie können Automatisierungen verwenden, um serverseitige Logik und Operationen in App Studio auszuführen. In Automatisierungsaktionen können Sie Ausdrücke verwenden, um Daten zu verarbeiten, dynamische Werte zu generieren und Ergebnisse aus früheren Aktionen zu integrieren.

Zugreifen auf Automatisierungsparameter

Sie können dynamische Werte aus UI-Komponenten und anderen Automatisierungen an Automatisierungen übergeben, sodass sie wiederverwendbar und flexibel sind. Dies erfolgt mithilfe von Automatisierungsparametern mit dem `params` Namespace wie folgt:

`{{params.parameterName}}`: Verweisen Sie auf einen Wert, der von einer UI-Komponente oder einer anderen Quelle an die Automatisierung übergeben wurde. `{{params.ID}}`Würde beispielsweise auf einen Parameter mit dem Namen verweisen`ID`.

Manipulation von Automatisierungsparametern

Sie können es verwenden JavaScript , um Automatisierungsparameter zu manipulieren. Im Folgenden sind einige Beispiele aufgeführt:

- `{{params.firstName}} {{params.lastName}}`: Verkettet Werte, die als Parameter übergeben wurden.
- `{{params.numberParam1 + params.numberParam2}}`: Fügt zwei Zahlenparameter hinzu.
- `{{params.valueProvided?.length > 0 ? params.valueProvided : 'Default'}}`: Prüft, ob ein Parameter nicht Null oder undefiniert ist und eine Länge ungleich Null hat. Falls wahr, verwenden Sie den angegebenen Wert; andernfalls legen Sie einen Standardwert fest.
- `{{params.rootCause || "No root cause provided"}}`: Wenn der `params.rootCause` Parameter falsch ist (Null, undefiniert oder eine leere Zeichenfolge), verwenden Sie den angegebenen Standardwert.
- `{{Math.min(params.numberOfProducts, 100)}}`: Beschränkt den Wert eines Parameters auf einen Maximalwert (in diesem Fall`100`).
- `{{ DateTime.fromISO(params.startDate).plus({ days: 7 }).toISO() }}`: Wenn der `params.startDate` Parameter „ist“`"2023-06-15T10:30:00.000Z"`, wird für diesen Ausdruck das Datum ausgewertet`"2023-06-22T10:30:00.000Z"`, das eine Woche nach dem Startdatum liegt.

Der Zugriff auf die Automatisierung resultiert aus einer früheren Aktion

Automatisierungen ermöglichen es der Anwendung, serverseitige Logik und Operationen auszuführen, z. B. Datenbanken abzufragen APIs, mit Daten zu interagieren oder Datentransformationen durchzuführen. Der `results` Namespace bietet Zugriff auf die Ausgaben

und Daten, die von früheren Aktionen innerhalb derselben Automatisierung zurückgegeben wurden. Beachten Sie beim Zugriff auf Automatisierungsergebnisse die folgenden Punkte:

1. Sie können nur auf Ergebnisse früherer Automatisierungsschritte innerhalb derselben Automatisierung zugreifen.
2. Wenn Sie Aktionen benannt *`action1`* und *`action2`* in dieser Reihenfolge haben, *`action1`* können Sie nicht auf Ergebnisse verweisen und *`action2`* können nur darauf zugreifen`results.action1`.
3. Dies funktioniert auch bei clientseitigen Aktionen. Zum Beispiel, wenn Sie eine Schaltfläche haben, die mithilfe der Aktion eine Automatisierung auslöst. `InvokeAutomation` Sie können dann einen Navigationsschritt mit einer `Run If` Bedingung ausführen, z. B. `results.`*`myInvokeAutomation1`*`.fileType === "pdf"` mit einem PDF-Viewer zu einer Seite zu navigieren, wenn die Automatisierung angibt, dass es sich bei der Datei um ein PDF handelt.

Die folgende Liste enthält die Syntax für den Zugriff auf Automatisierungsergebnisse einer früheren Aktion unter Verwendung des `results` Namespace.

- `{{results.`*`stepName`*`.data}}`: Ruft das Datenarray aus einem Automatisierungsschritt mit dem Namen *`stepName`* ab.
- `{{results.`*`stepName`*`.output}}`: Ruft die Ausgabe eines Automatisierungsschritts mit dem Namen ab*`stepName`*.

Wie Sie auf die Ergebnisse eines Automatisierungsschritts zugreifen, hängt von der Art der Aktion und den zurückgegebenen Daten ab. Verschiedene Aktionen können unterschiedliche Eigenschaften oder Datenstrukturen zurückgeben. Hier sind einige gängige Beispiele:

- Für eine Datenaktion können Sie mit dem Befehl auf das zurückgegebene Datenarray zugreifen`results.`*`stepName`*`.data`.
- Bei einer API-Aufrufaktion können Sie auf den Antworttext zugreifen, indem Sie`results.`*`stepName`*`.body`.
- Für eine Amazon S3 S3-Aktion können Sie auf den Dateiinhalt zugreifen mit`results.`*`stepName`*`.Body.transformToWebStream()`.

In der Dokumentation zu den spezifischen Aktionstypen, die Sie verwenden, finden Sie Informationen zur Form der Daten, die sie zurückgeben, und darüber, wie Sie innerhalb des `results` Namespace darauf zugreifen können. Die folgende Liste enthält einige Beispiele

- `{{results.getDataStep.data.filter(row => row.status === "pending").length}}`: Unter der Annahme, dass es sich um eine `Invoke Data Action` Automatisierungsaktion *`getDataStep`* handelt, die ein Array von Datenzeilen zurückgibt, filtert dieser Ausdruck das Datenarray so, dass es nur Zeilen enthält, bei denen das Statusfeld gleich ist`pending`, und gibt die Länge (Anzahl) des gefilterten Arrays zurück. Dies kann nützlich sein, wenn Daten unter bestimmten Bedingungen abgefragt oder verarbeitet werden sollen.
- `{{params.email.split("@")[0]}}`: Wenn der `params.email` Parameter eine E-Mail-Adresse enthält, teilt dieser Ausdruck die Zeichenfolge am @-Symbol auf und gibt den Teil vor dem @-Symbol zurück, wodurch der Nutzername aus der E-Mail-Adresse extrahiert wird.
- `{{new Date(params.timestamp * 1000)}}`: Dieser Ausdruck verwendet einen Unix-Zeitstempelparameter (`params.timestamp`) und konvertiert ihn in ein JavaScript Date-Objekt. Er geht davon aus, dass der Zeitstempel in Sekunden angegeben ist, also multipliziert er ihn mit 1000, um ihn in Millisekunden umzurechnen, was dem vom Konstruktor erwarteten Format entspricht. `Date` Dies kann für die Arbeit mit Datums- und Uhrzeitwerten in Automatisierungen nützlich sein.
- `{{results.stepName.Body}}`: Bei einer `Amazon S3 GetObject` Automatisierungsaktion mit dem Namen *`stepName`* ruft dieser Ausdruck den Dateiinhalt ab, der von Benutzeroberflächenkomponenten wie Image oder PDF Viewer zur Anzeige der abgerufenen Datei verwendet werden kann. Beachten Sie, dass dieser Ausdruck in der Automatisierungsausgabe der Automatisierung konfiguriert werden müsste, um ihn in Komponenten verwenden zu können.

Datenabhängigkeiten und zeitliche Überlegungen

Bei der Erstellung komplexer Anwendungen in App Studio ist es wichtig, die Datenabhängigkeiten zwischen verschiedenen Datenkomponenten wie Formularen, Detailansichten und automatisierungsgestützten Komponenten zu verstehen und zu verwalten. Datenkomponenten und Automatisierungen schließen ihren Datenabruf oder ihre Ausführung möglicherweise nicht gleichzeitig ab, was zu Zeitproblemen, Fehlern und unerwartetem Verhalten führen kann. Wenn Sie sich potenzieller Timing-Probleme bewusst sind und Best Practices befolgen, können Sie zuverlässigere und konsistentere Benutzererlebnisse in Ihren App Studio-Anwendungen schaffen.

Einige mögliche Probleme sind wie folgt:

1. Konflikte beim Rendern: Datenkomponenten werden möglicherweise in einer Reihenfolge gerendert, die nicht ihren Datenabhängigkeiten entspricht, was möglicherweise zu visuellen Inkonsistenzen oder Fehlern führen kann.
2. Timing der Automatisierungsausführung: Automatisierungsaufgaben können abgeschlossen werden, bevor die Komponenten vollständig geladen sind, was zu Fehlern bei der Ausführung der Laufzeit führt.
3. Komponentenabstürze: Komponenten, die auf Automatisierungen basieren, können bei ungültigen Antworten oder wenn die Automatisierung noch nicht abgeschlossen ist, abstürzen.

Beispiel: Bestelldetails und Kundeninformationen

Dieses Beispiel zeigt, wie Abhängigkeiten zwischen Datenkomponenten zu Zeitproblemen und potenziellen Fehlern bei der Datenanzeige führen können.

Stellen Sie sich eine Anwendung mit den folgenden zwei Datenkomponenten auf derselben Seite vor:

- Eine Detailkomponente (`orderDetails`), die Auftragsdaten abruft.
- Eine Detailkomponente (`customerDetails`), die Kundendetails zur Bestellung anzeigt.

In dieser Anwendung gibt es zwei Felder in der `orderDetails` Detailkomponente, die mit den folgenden Werten konfiguriert sind:

```
// 2 text fields within the orderDetails detail component

// Info from orderDetails Component
{{ui.orderDetails.data[0].name}}

// Info from customerDetails component
{{ui.customerDetails.data[0].name}} // Problematic reference
```

In diesem Beispiel versucht die `orderDetails` Komponente, den Kundennamen anzuzeigen, indem sie auf Daten aus der `customerDetails` Komponente verweist. Dies ist problematisch, da die `orderDetails` Komponente möglicherweise rendert, bevor die `customerDetails` Komponente ihre Daten abgerufen hat. Wenn der Abruf der `customerDetails` Komponentendaten verzögert wird oder fehlschlägt, zeigt die `orderDetails` Komponente unvollständige oder falsche Informationen an.

Bewährte Methoden für Datenabhängigkeit und Timing

Verwenden Sie die folgenden bewährten Methoden, um Probleme mit Datenabhängigkeit und Timing in Ihrer App Studio-App zu minimieren:

1. Bedingtes Rendern verwenden: Rendern Sie Komponenten oder zeigen Sie Daten nur an, wenn Sie bestätigt haben, dass sie verfügbar sind. Verwenden Sie bedingte Anweisungen, um zu überprüfen, ob Daten vorhanden sind, bevor Sie sie anzeigen. Der folgende Ausschnitt zeigt ein Beispiel für eine bedingte Anweisung:

   ```
   {{ui.someComponent.data ? ui.someComponent.data.fieldName : "Loading..."}}
   ```

2. Sichtbarkeit untergeordneter Komponenten verwalten: Für Komponenten wie Stepflow, Form oder Detail, die untergeordnete Komponenten rendern, bevor ihre Daten geladen werden, legen Sie die Sichtbarkeit der untergeordneten Komponenten manuell fest. Der folgende Ausschnitt zeigt ein Beispiel für die Einstellung der Sichtbarkeit auf der Grundlage der Datenverfügbarkeit der übergeordneten Komponente:

   ```
   {{ui.parentComponent.data ? true : false}}
   ```

3. Verwenden Sie Verbindungsabfragen: Verwenden Sie nach Möglichkeit Verbindungsabfragen, um verwandte Daten in einer einzigen Abfrage abzurufen. Dadurch wird die Anzahl der separaten Datenabrufe reduziert und Zeitprobleme zwischen Datenkomponenten minimiert.
4. Implementieren Sie die Fehlerbehandlung in Automatisierungen: Implementieren Sie eine robuste Fehlerbehandlung in Ihren Automatisierungen, um Szenarien, in denen erwartete Daten nicht verfügbar sind oder ungültige Antworten eingehen, elegant zu verwalten.
5. Optionale Verkettung verwenden: Verwenden Sie beim Zugriff auf verschachtelte Eigenschaften die optionale Verkettung, um Fehler zu vermeiden, wenn eine übergeordnete Eigenschaft nicht definiert ist. Der folgende Ausschnitt zeigt ein Beispiel für optionale Verkettung:

   ```
   {{ui.component.data?.[0]?.fieldSystemName}}
   ```

Eine App mit mehreren Benutzern erstellen

Mehrere Benutzer können an einer einzigen App Studio-App arbeiten, aber nur ein Benutzer kann eine App gleichzeitig bearbeiten. In den folgenden Abschnitten finden Sie Informationen zur

Einladung anderer Benutzer zur Bearbeitung einer App und zum Verhalten, wenn mehrere Benutzer versuchen, eine App gleichzeitig zu bearbeiten.

Laden Sie Entwickler ein, eine App zu bearbeiten

Verwenden Sie die folgenden Anweisungen, um andere Builder zur Bearbeitung einer App Studio-App einzuladen.

Um andere Builder zur Bearbeitung einer App einzuladen

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung.
2. Wählen Sie Freigeben.
3. Verwenden Sie auf der Registerkarte Entwicklung das Textfeld, um nach Gruppen oder einzelnen Benutzern zu suchen und diese auszuwählen, die Sie zur Bearbeitung der App einladen möchten.
4. Wählen Sie für jeden Benutzer oder jede Gruppe das Drop-down-Menü aus und wählen Sie die Berechtigungen aus, die Sie diesem Benutzer oder dieser Gruppe gewähren möchten.
    - Mitinhaber: Miteigentümer haben dieselben Berechtigungen wie App-Besitzer.
    - Nur bearbeiten: Benutzer mit der Rolle „Nur bearbeiten“ haben dieselben Berechtigungen wie Eigentümer und Mitinhaber, mit Ausnahme der folgenden:
        - Sie können andere Benutzer nicht einladen, die App zu bearbeiten.
        - Sie können die App nicht in der Test- oder Produktionsumgebung veröffentlichen.
        - Sie können der App keine Datenquellen hinzufügen.
        - Sie können die App nicht löschen oder duplizieren.

Es wird versucht, eine App zu bearbeiten, die gerade von einem anderen Benutzer bearbeitet wird

Eine einzelne App Studio-App kann jeweils nur von einem Benutzer bearbeitet werden. Sehen Sie sich das folgende Beispiel an, um zu verstehen, was passiert, wenn mehrere Benutzer versuchen, eine App gleichzeitig zu bearbeiten.

In diesem Beispiel bearbeitet `User A` gerade eine App und hat sie mit geteiltUser B. `User B`versucht dann, die App zu bearbeiten, die gerade bearbeitet wird`User A`.

Wenn `User B` versucht wird, die App zu bearbeiten, erscheint ein Dialogfeld mit der Information, dass die App gerade bearbeitet `User A` wird. Wenn Sie fortfahren, wird das Anwendungsstudio `User A` verlassen und alle Änderungen werden gespeichert. `User B`Sie können wählen, ob Sie den Vorgang abbrechen und `User A` fortfahren lassen möchten, oder ob Sie fortfahren und das Anwendungsstudio aufrufen möchten, um die App zu bearbeiten. In diesem Beispiel entscheiden sie sich dafür, die App zu bearbeiten.

Wenn `User B` sich entscheidet, die App zu bearbeiten, `User A` erhält sie eine Benachrichtigung, dass mit der Bearbeitung der App begonnen `User B` wurde und ihre Sitzung beendet wurde. Beachten Sie, dass sie die Benachrichtigung möglicherweise nicht erhalten, wenn `User A` die App in einem inaktiven Browser-Tab geöffnet wäre. Wenn sie in diesem Fall versuchen, zur App zurückzukehren und eine Änderung vorzunehmen, erhalten sie eine Fehlermeldung und werden aufgefordert, die Seite zu aktualisieren, wodurch sie wieder zur Liste der Anwendungen zurückkehren.

Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren

Jede Anwendung in App Studio verfügt über Inhaltssicherheitseinstellungen, mit denen Sie verhindern können, dass externe Medien oder Ressourcen wie Bilder, iFrames geladen werden oder dass sie nur PDFs von bestimmten Domains oder URLs (einschließlich Amazon S3 S3-Buckets) geladen werden dürfen. Sie können auch die Domains angeben, in die Ihre App Objekte auf Amazon S3 hochladen kann.

Die Standardeinstellungen für die Inhaltssicherheit für alle Apps bestehen darin, das Laden aller Medien aus externen Quellen, einschließlich Amazon S3-Buckets, und das Hochladen von Objekten auf Amazon S3 zu blockieren. Um Bilder, iFrames oder ähnliche Medien zu laden PDFs, müssen Sie daher die Einstellungen bearbeiten, um die Medienquellen zuzulassen. Um das Hochladen von Objekten auf Amazon S3 zu ermöglichen, müssen Sie außerdem die Einstellungen bearbeiten, um die Domains zuzulassen, in die hochgeladen werden können.

> **Note**
>
> Die Inhaltssicherheitseinstellungen werden verwendet, um Content Security Policy (CSP) -Header in Ihrer Anwendung zu konfigurieren. CSP ist ein Sicherheitsstandard, der dazu beiträgt, Ihre App vor Cross-Site Scripting (XSS), Clickjacking und anderen Code-Injection-

Angriffen zu schützen. Weitere Informationen zu CSP finden Sie unter Content Security Policy (CSP) in den MDN Web Docs.

Um die Inhaltssicherheitseinstellungen Ihrer App zu aktualisieren

1. Navigieren Sie bei Bedarf zum Anwendungsstudio Ihrer Anwendung, indem Sie in der Anwendungsliste auswählen, ob Sie es bearbeiten möchten.
2. Wählen Sie App-Einstellungen.
3. Wählen Sie den Tab „Content Security Settings“, um die folgenden Einstellungen zu sehen:

   - Frame-Quelle: Wird verwendet, um die Domains zu verwalten, aus denen Ihre App Frames und Iframes (z. B. interaktive Inhalte oder PDFs) laden kann. Diese Einstellung wirkt sich auf die folgenden Komponenten oder App-Ressourcen aus:
     - Komponente zum Einbetten von iFrames
     - Komponente für den PDF-Viewer
   - Bildquelle: Wird verwendet, um die Domains zu verwalten, aus denen Ihre App Bilder laden kann. Diese Einstellung wirkt sich auf die folgenden Komponenten oder App-Ressourcen aus:
     - Logo und Banner der App
     - Komponente „Image Viewer“
   - Verbindungsquelle: Wird verwendet, um die Domains zu verwalten, in die Ihre App Amazon S3 S3-Objekte hochladen kann.
4. Wählen Sie für jede Einstellung die gewünschte Einstellung aus dem Drop-down-Menü aus:

   - Alles blockieren frames/images/connections: Erlauben Sie nicht, dass Medien (Bilder, Frames PDFs) geladen oder Objekte auf Amazon S3 hochgeladen werden.
   - Alle zulassen frames/images/connections: Erlaubt das Laden aller Medien (Bilder, Frames PDFs) aus allen Domains oder erlaubt das Hochladen von Objekten auf Amazon S3 für alle Domains.
   - Bestimmte Domänen zulassen: Erlaubt das Laden von Medien von oder das Hochladen von Medien in bestimmte Domänen. Domänen oder URLs werden als durch Leerzeichen getrennte Liste von Ausdrücken angegeben, wobei Platzhalter (`*`) für Subdomänen, Hostadressen oder Portnummern verwendet werden können, um anzuzeigen, dass alle zulässigen Werte der einzelnen Ausdrücke gültig sind. Die Angabe entspricht `http` auch. `https` Die folgende Liste enthält Beispiele für gültige Einträge:

- `blob:`: Entspricht allen Blobs, einschließlich Dateidaten, die durch Automatisierungsaktionen zurückgegeben wurden, wie z. B. die `GetObject` Rückgabe von Artikeln aus Amazon S3 S3-Buckets oder von Amazon Bedrock generierte Bilder.

  > **Important**
  >
  > Sie müssen `blob:` zu Ihrem angegebenen Ausdruck hinzufügen, damit Dateidaten von Aktionen zurückgegeben werden können. Auch wenn Ihr Ausdruck dies ist`*`, sollten Sie ihn wie folgt aktualisieren `* blob:`

- `http://*.example.com`: Entspricht allen Ladeversuchen aus einer beliebigen Subdomain von`example.com`. Stimmt auch mit `https` Ressourcen überein.
- `https://source1.example.com https//source2.example.com`: Entspricht allen Ladeversuchen `https://source1.example.com` sowohl als auch `https://source2.example.com`
- `https://example.com/subdirectory/`: Entspricht allen Versuchen, Dateien in ein Unterverzeichnisverzeichnis zu laden. Beispiel, `https://example.com/subdirectory/path/to/file.jpeg`. Es stimmt nicht überein`https://example.com/path/to/file.jpeg`.

5. Wählen Sie Speichern, um Ihre Änderungen zu speichern.

Problembehandlung und Debuggen von App Studio

Themen

- Problembehebung bei der Einrichtung, den Berechtigungen und dem Onboarding von App Studio
- Problembehandlung und Debuggen von Apps
- Problembehebung beim Veröffentlichen und Teilen von Anwendungen

Problembehebung bei der Einrichtung, den Berechtigungen und dem Onboarding von App Studio

Dieses Thema enthält Informationen zur Behebung häufiger Probleme bei der Einrichtung oder beim Onboarding von App Studio und zur Verwaltung von Berechtigungen.

Das App Studio-Setup ist fehlgeschlagen, als die Option Eine Kontoinstanz für mich erstellen ausgewählt wurde

Problem: Die Einrichtung von App Studio mit der Instanz Ein Konto für mich erstellen schlägt fehl, wenn Sie in einer beliebigen AWS Region eine IAM Identity Center-Instanz auf Kontoebene haben, da IAM Identity Center nur eine Instanz unterstützt.

Lösung: Rufen Sie die IAM Identity Center-Konsole unter auf, https://console.aws.amazon.com/singlesignon/um zu überprüfen, ob Sie über eine IAM Identity Center-Instanz verfügen. Überprüfen Sie alle unterstützten AWS Regionen, bis Sie die Instanz gefunden haben. Sie können diese Instanz entweder bei der Einrichtung von App Studio verwenden oder die IAM Identity Center-Instanz löschen und es mit der Option Kontoinstanz für mich erstellen erneut versuchen.

> **Warning**
>
> Das Löschen der IAM Identity Center-Instanz wirkt sich auf alle vorhandenen Anwendungsfälle aus. Stellen Sie vor dem Löschen sicher, dass die Instanz nicht verwendet wird, oder verwenden Sie die Instanz, um App Studio einzurichten.

Nach der Einrichtung kann nicht auf App Studio zugegriffen werden

Problem: Bei der Einrichtung von App Studio haben Sie möglicherweise IAM Identity Center-Gruppen angegeben, in denen Sie kein Mitglied sind. Sie müssen Mitglied mindestens einer Gruppe sein, um auf App Studio zugreifen zu können.

Lösung: Navigieren Sie zur IAM Identity Center-Konsole unter https://console.aws.amazon.com/singlesignon/, um sich einer Gruppe hinzuzufügen, die bei der Einrichtung zu App Studio hinzugefügt wurde.

Sie sind sich nicht sicher, welchen Benutzernamen oder welches Passwort Sie verwenden sollen, wenn Sie sich bei App Studio anmelden

Problem: Möglicherweise sind Sie sich nicht sicher, wie Sie sich bei App Studio anmelden sollen, entweder weil Sie Ihre IAM Identity Center-Anmeldeinformationen nicht eingerichtet haben oder weil Sie Ihren IAM Identity Center-Benutzernamen oder Ihr Passwort vergessen haben.

Lösung: Bei der Einrichtung von App Studio ohne eine IAM Identity Center-Instanz wurden für jeden Benutzer eine E-Mail-Adresse und ein Benutzername angegeben, die zur Erstellung von IAM Identity Center-Benutzern verwendet wurden. Jede der angegebenen E-Mail-Adressen erhielt eine E-Mail mit einer Einladung, IAM Identity Center beizutreten. Jeder Benutzer muss die Einladung annehmen und ein Passwort für seine IAM Identity Center-Benutzeranmeldedaten erstellen. Jeder Benutzer kann sich dann mit dem Benutzernamen und dem Passwort für das IAM Identity Center bei App Studio anmelden.

Wenn Sie bereits Anmeldeinformationen eingerichtet und Ihren Benutzernamen oder Ihr Passwort vergessen haben, müssen Sie Ihren Administrator bitten, in der IAM Identity Center-Konsole Ihren Benutzernamen einzusehen und einzugeben oder Ihr Passwort zurückzusetzen.

Beim Einrichten von App Studio erhalte ich einen Systemfehler

Problem: Beim Einrichten von App Studio wird die folgende Fehlermeldung angezeigt:

```
System error. We encountered a problem. Report the issue and the App Studio service
 team will get back to you.
```

Dieser Fehler tritt auf, wenn der Dienst auf einen unbekannten Fehler gestoßen ist.

Lösung: Kontaktiere das Support-Team, indem du der Community Slack beitrittst. Wähle dazu in der linken Navigationsleiste im Abschnitt Lernen die Option Join us on Slack oder im oberen Banner, während du eine App bearbeitest.

Ich kann die URL meiner App Studio-Instanz nicht finden

Wenn Sie die URL für den Zugriff auf Ihre App Studio-Instanz nicht finden können, wenden Sie sich an den Administrator, der App Studio eingerichtet hat. Der Administrator kann die URL in der App Studio-Konsole im einsehen AWS Management Console.

Ich kann Gruppen oder Rollen in App Studio nicht ändern

Problem: Sie können den Link Rollen in der linken Navigationsleiste nicht sehen. Das liegt daran, dass nur Benutzer mit der Administratorrolle Gruppen und Rollen in App Studio ändern können.

Lösung: Wenden Sie sich an einen Benutzer mit der Administratorrolle, um Gruppen oder Rollen zu ändern, oder wenden Sie sich an Ihren Administrator, um zu einer Admin-Gruppe hinzugefügt zu werden.

Wie kann ich App Studio verlassen

Derzeit können Sie App Studio nicht verlassen. Es wird empfohlen, alle Ressourcen wie Apps und Connectors zu entfernen und die Rolle der Gruppen in „App-Benutzer“ zu ändern, um den Zugriff oder die Nutzung zu verhindern. Sie sollten auch Ressourcen von Drittanbietern löschen, die ausschließlich für App Studio verwendet werden, z. B. IAM-Rollen oder Datenbanktabellen.

Problembehandlung und Debuggen von Apps

Die folgenden Themen enthalten Informationen zur Fehlerbehebung und zum Debuggen von App Studio-Apps.

Themen

- Fehlerbehebung beim AI Builder-Assistenten und Chat
- Fehlerbehebung im Anwendungsstudio
- Problembehandlung bei der Vorschau von Apps
- Fehlerbehebung in der Testumgebung
- Debuggen mit Protokollen von veröffentlichten Apps in Amazon CloudWatch Logs
- Fehlerbehebung bei Konnektoren

Fehlerbehebung beim AI Builder-Assistenten und Chat

Dieses Thema enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme bei der Verwendung des AI Builder-Assistenten.

Fehler beim Erstellen einer App mit KI

Wenn Sie die KI-Eingabeaufforderung verwenden, um eine App zu erstellen, kann der folgende Fehler auftreten:

```
We apologize, but we cannot proceed with your request. The request may contain content
 that violates our policies and guidelines. Please revise your prompt before trying
 again.
```

Problem: Die Anfrage wurde aufgrund potenziell schädlicher Inhalte blockiert.

Lösung: Formulieren Sie die Aufforderung neu und versuchen Sie es erneut.

Bei der mit KI generierten App handelt es sich um eine leere App oder es fehlen Komponenten.

Problem: Dies kann durch einen unerwarteten Servicefehler verursacht werden.

Lösung: Versuchen Sie erneut, die App mithilfe von KI zu erstellen, oder erstellen Sie die Komponenten manuell in der generierten App.

Fehlerbehebung im Anwendungsstudio

Dieses Thema enthält Anleitungen zur Fehlerbehebung und zum Debuggen von Problemen beim Erstellen von Anwendungen.

Verwenden des Debug-Bedienfelds

Um Sie beim Live-Debuggen während der Erstellung Ihrer Apps zu unterstützen, bietet App Studio ein zusammenklappbares Builder-Debug-Bedienfeld, das sich über die Seiten, Automatisierungen und Datenregister des Anwendungsstudios erstreckt. In diesem Bereich werden sowohl Fehler als auch Warnungen angezeigt. Warnungen dienen zwar als umsetzbare Vorschläge, z. B. für Ressourcen, die nicht konfiguriert wurden, aber Fehler müssen behoben werden, um Ihre App erfolgreich in der Vorschau anzeigen oder veröffentlichen zu können. Jeder Fehler oder jede Warnung enthält einen Link zum Anzeigen, über den Sie zum Ort des Problems navigieren können.

Das Debug-Bedienfeld wird automatisch mit neuen Fehlern oder Warnungen aktualisiert, sobald sie auftreten, und die Fehler oder Warnungen verschwinden automatisch, sobald sie behoben sind. Der Status dieser Warn- und Fehlermeldungen wird beibehalten, wenn Sie den Builder verlassen.

JavaScript Ausdruckssyntax und Behandlung von Datentypen

App Studio bietet eine JavaScript Fehlererkennung und hebt Fehler hervor, indem Ihr Code mit roten Linien unterstrichen wird. Diese Kompilierungsfehler, die verhindern, dass die App erfolgreich erstellt werden kann, deuten auf Probleme wie Tippfehler, ungültige Verweise, ungültige Operationen und falsche Ausgaben für erforderliche Datentypen hin. In der folgenden Liste finden Sie häufig auftretende Probleme:

1. Fehler beim Umbenennen von Ressourcen: Wenn JavaScript Ausdrücke auf Ressourcennamen in App Studio verweisen, führt das Ändern dieser Namen dazu, dass die Ausdrücke falsch sind und zu Fehlern führen. Sie können sich diese Fehler im Debug-Bereich ansehen.
2. Probleme mit Datentypen: Nicht übereinstimmende Datentypen führen zu Fehlern in Ihrer App. Wenn eine Automatisierung beispielsweise so konfiguriert ist, dass sie einen Typparameter akzeptiert`String`, eine Komponente jedoch so konfiguriert ist, dass sie einen Wert vom Typ sendet`Integer`, tritt ein Fehler auf. Überprüfen Sie, ob die Datentypen der entsprechenden Ressourcen, einschließlich Komponenten, Automatisierungen sowie Datenentitäten und Aktionen, übereinstimmen. Möglicherweise müssen Sie den Typ des Werts in einem JavaScript Ausdruck ändern.

Problembehandlung bei der Vorschau von Apps

Dieses Thema enthält Informationen zur Behebung von Problemen beim Versuch, eine Vorschau von Apps anzuzeigen.

Die Vorschau kann mit dem folgenden Fehler nicht geladen werden: **`Your app failed to build and cannot be previewed`**

Problem: Ihre App muss erfolgreich erstellt werden, damit sie in der Vorschau angezeigt werden kann. Dieser Fehler tritt auf, wenn ein Kompilierungsfehler auftritt, der verhindert, dass Ihre App erfolgreich erstellt wird.

Lösung: Überprüfen und beheben Sie die Fehler mithilfe des Debug-Bedienfelds im Anwendungsstudio.

Das Laden der Vorschau dauert sehr lange

Problem: Bestimmte Arten von App-Updates benötigen viel Zeit zum Kompilieren und Erstellen.

Lösung: Lassen Sie den Tab geöffnet und warten Sie, bis die Updates erstellt sind. In der oberen rechten Ecke des Anwendungsstudios der App sollte „Gespeichert“ angezeigt werden, und die Vorschau wird erneut geladen.

Die Vorschau spiegelt nicht die neuesten Änderungen wider

Problem: Dies kann passieren, wenn Ihre App-Bearbeitungssitzung von einem anderen Benutzer übernommen wurde, Sie aber nicht benachrichtigt wurden. Dies kann dazu führen, dass die bearbeitete App nicht mit der Vorschauumgebung übereinstimmt.

Lösung: Aktualisieren Sie den Browser-Tab von Application Studio und übernehmen Sie bei Bedarf die Bearbeitungssitzung.

Fehlerbehebung in der Testumgebung

Dieses Thema enthält Informationen zur Fehlerbehebung bei Apps, die in der Testumgebung veröffentlicht wurden.

> **Note**
>
> Eine HTTP 500-Antwort von einer Automatisierungs- oder Datenaktion kann durch einen Laufzeitabsturz in Ihren Ausdrücken, einen Konnektorfehler oder eine Drosselung durch eine Datenquelle verursacht werden, die mit Ihrer Anwendung verbunden ist. Folgen Sie den Anweisungen unterVerwenden Sie Ihre Browserkonsole zum Debuggen, um die Debug-Protokolle mit den zugrunde liegenden Fehlerdetails anzuzeigen.

Verwenden des Debug-Bedienfelds

Ähnlich wie das Build-Debug-Bedienfeld, das beim Erstellen Ihrer Apps verwendet wird, bietet App Studio in der Testumgebung ein zusammenklappbares Debug-Bedienfeld. In diesem Bereich werden Informationsmeldungen wie die Ladezeit der Seite, die Benutzernavigation und App-Ereignisse angezeigt. Es enthält auch Fehler und Warnungen. Das Debug-Bedienfeld wird automatisch mit neuen Meldungen aktualisiert, wenn Ereignisse eintreten.

Verwenden Sie Ihre Browserkonsole zum Debuggen

Da während der Vorschau Ihrer App keine Aktionen aufgerufen werden, muss Ihre App in der Testumgebung veröffentlicht werden, um ihre Anruf- und Antwortverarbeitung zu testen. Wenn während der Ausführung Ihrer Automatisierung ein Fehler auftritt oder wenn Sie verstehen möchten, warum sich die Anwendung auf eine bestimmte Weise verhält, können Sie die Konsole Ihres Browsers für das Debuggen in Echtzeit verwenden.

Um Ihre Browserkonsole zum Debuggen von Apps in der Testumgebung zu verwenden

1. Hängen Sie die URL `?debug=true` an das Ende an und drücken Sie die Eingabetaste. Beachten Sie, dass, wenn die URL bereits eine Abfragezeichenfolge enthält (sie enthält?), diese stattdessen `&debug=true` an das Ende der URL angehängt werden sollte.
2. Öffnen Sie Ihre Browserkonsole, um mit dem Debuggen zu beginnen, indem Sie Ihre Aktionen oder API-Eingaben und -Ausgaben untersuchen.

    - In Chrome: Klicken Sie mit der rechten Maustaste in Ihren Browser und wählen Sie Inspect. Weitere Informationen zum Debuggen mit Chrome DevTools finden Sie in der DevTools Chrome-Dokumentation.
    - In Firefox: Halten Sie ein Webseitenelement gedrückt oder klicken Sie mit der rechten Maustaste darauf und wählen Sie dann „Element untersuchen“. Weitere Informationen zum Debuggen mit Firefox DevTools finden Sie in den DevTools Firefox-Benutzerdokumenten.

Die folgende Liste enthält einige häufig auftretende Probleme, die zu Fehlern führen:

- Laufzeitfehler
    - Problem: Wenn eine Automatisierung oder ein Ausdruck falsch konfiguriert ist, kann dies zu einem Fehler führen, wenn die Automatisierung ausgeführt wird. Häufige Fehler sind das Umbenennen von Assets, was zu falschen Ausdrücken führt, andere JavaScript Kompilierungsfehler oder Versuche, Daten oder Assets zu verwenden, die es sind. `undefined`
    - Lösung: Überprüfen Sie jede Verwendung von benutzerdefiniertem Codeeingabe (Ausdrücke JavaScript, und JSON) und stellen Sie sicher, dass im Code-Editor oder im Debug-Bereich keine Kompilierungsfehler vorliegen.
- Probleme mit dem Connector
    - Problem: Da App Studio-Apps erst mit externen Diensten über Konnektoren kommunizieren, wenn sie veröffentlicht wurden, können in der Testumgebung Fehler auftreten, die in der Vorschauversion nicht aufgetreten sind. Wenn eine Aktion in einer Automatisierung, die einen

Connector verwendet, fehlschlägt, kann dies an einer Fehlkonfiguration in der Aktion liegen, die die Anforderung an den Connector sendet, oder an der Connectorkonfiguration selbst.

- Lösung: Sie sollten Mocked Output verwenden, um Automatisierungen zu einem frühen Zeitpunkt in der Vorschauumgebung zu testen, um diese Fehler zu vermeiden. Stellen Sie sicher, dass Ihr Connector richtig konfiguriert ist. Weitere Informationen finden Sie unter. Fehlerbehebung bei Konnektoren Schließlich können Sie sie verwenden, CloudWatch um die Protokolle zu überprüfen. Weitere Informationen finden Sie unter Debuggen mit Protokollen von veröffentlichten Apps in Amazon CloudWatch Logs. In den `ConnectorService` Namespace-Protokollen sollten Fehlermeldungen oder Metadaten enthalten sein, die vom Connector stammen.

Debuggen mit Protokollen von veröffentlichten Apps in Amazon CloudWatch Logs

Amazon CloudWatch Logs überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können CloudWatch Logs verwenden, um Metriken zu sammeln und nachzuverfolgen. Dabei handelt es sich um Variablen, die Sie für Ihre Ressourcen und Anwendungen messen können.

Für das Debuggen von App Studio-Apps ist CloudWatch Logs nützlich, um Fehler zu verfolgen, die während der Ausführung einer App auftreten, Informationen zu prüfen und Kontext zu Benutzeraktionen und proprietären Interaktionen bereitzustellen. Die Protokolle bieten historische Daten, anhand derer Sie die Nutzung und Zugriffsmuster von Anwendungen überprüfen und Fehler überprüfen können, auf die Benutzer gestoßen sind.

> **Note**
>
> CloudWatch Logs bietet keine Echtzeit-Tracking von Parameterwerten, die von der Benutzeroberfläche einer Anwendung übergeben wurden.

Gehen Sie wie folgt vor, um in CloudWatch Logs auf Logs aus Ihren App Studio-Apps zuzugreifen.

1. Suchen Sie im App Studio-Anwendungsstudio für Ihre App nach Ihrer App-ID und notieren Sie sich diese, indem Sie in der URL nachsehen. Die App-ID könnte etwa so aussehen:`802a3bd6-ed4d-424c-9f6b-405aa42a62c5`.
2. Öffnen Sie die CloudWatch Konsole unter https://console.aws.amazon.com/cloudwatch/.

3. Wählen Sie im Navigationsbereich Protokollgruppen aus.
4. Hier finden Sie fünf Protokollgruppen pro Anwendung. Wählen Sie je nach Art der Informationen, an denen Sie interessiert sind, eine Gruppe aus und schreiben Sie eine Abfrage für die Daten, die Sie ermitteln möchten.

   Die folgende Liste enthält die Protokollgruppen und Informationen darüber, wann die einzelnen Gruppen verwendet werden sollten:

   1. `/aws/appstudio/teamId/appId/TEST/app`: Wird verwendet, um Automatisierungsantworten, Komponentenfehler oder JavaScript Code im Zusammenhang mit der Version Ihrer App zu debuggen, die derzeit in der Testumgebung veröffentlicht ist.
   2. `/aws/appstudio/teamId/appId/TEST/audit`: Wird verwendet, um JavaScript Codefehler zu debuggen, z. B. bedingte Sichtbarkeit oder Transformation, Abfragefehler und Fehler bei der Anmeldung oder bei Zugriffsberechtigungen im Zusammenhang mit der Version Ihrer App, die derzeit in der Testumgebung veröffentlicht ist.
   3. `/aws/appstudio/teamId/setup`: Wird verwendet, um Builder- oder Admin-Aktionen zu überwachen.
   4. `/aws/appstudio/teamId/appId/PRODUCTION/app`: Wird verwendet, um Automatisierungsantworten, Abfragefehler, Komponentenfehler oder JavaScript Code im Zusammenhang mit der Version Ihrer App zu debuggen, die derzeit in der Produktionsumgebung veröffentlicht ist.
   5. `/aws/appstudio/teamId/appId/PRODUCTION/audit`: Wird verwendet, um JavaScript Codefehler wie eingeschränkte Sichtbarkeit oder Transformation sowie Fehler bei Anmelde- oder Zugriffsberechtigungen im Zusammenhang mit der Version Ihrer App zu debuggen, die derzeit in der Produktionsumgebung veröffentlicht ist.

   > **Note**
   >
   > Die meisten Protokolle, die für das Debuggen verwendet werden, sind dem `DebugLogClient` Namespace zugeordnet.

5. Sobald Sie sich in einer Protokollgruppe befinden, können Sie entweder die neuesten Protokolldatenströme auswählen oder einen, bei dem die Uhrzeit des letzten Ereignisses dem gewünschten Zeitpunkt am nächsten kommt, oder Sie können alle Protokolldatenströme durchsuchen, um alle Ereignisse in dieser Protokollgruppe zu durchsuchen. Weitere

Informationen zum Anzeigen von Protokolldaten in CloudWatch Logs finden Sie unter An CloudWatch Logs gesendete Protokolldaten anzeigen.

Verwenden von CloudWatch Logs Insights-Abfragen zum Filtern und Sortieren von Protokollen

Sie können CloudWatch Logs Insights verwenden, um mehrere Protokollgruppen gleichzeitig abzufragen. Sobald Sie eine Liste von Protokollgruppen identifiziert haben, die Sitzungsinformationen enthalten, navigieren Sie zu CloudWatch Logs Insights und wählen Sie die Protokollgruppen aus. Grenzen Sie dann die Zielprotokolleinträge weiter ein, indem Sie die Abfrage anpassen. Hier sind einige Beispielabfragen:

Liste der Logs, die das Schlüsselwort enthalten: ***error***

```
fields @timestamp, @message
| filter @message like 'error'
| sort @timestamp desc
```

Debug-Protokolle aus der Testumgebung:

```
fields @timestamp, @message
| filter namespace = "DebugLogClient"
| sort @timestamp desc
```

Gesamtzahl der Fehler 504/404/500 in Intervallen von 5 Minuten:

```
filter @message like '/api/automation' and (@message like ': 404' or @message like ':
 500' or @message like ': 504')
| fields @timestamp, method, path, statusCode
| stats count(*) as errorCount by bin(5m)
```

Weitere Informationen zu CloudWatch Logs Insights finden Sie unter Analysieren von Protokolldaten mit CloudWatch Logs Insights im Amazon CloudWatch Logs-Benutzerhandbuch.

Fehlerbehebung bei Konnektoren

Dieses Thema enthält Anleitungen zur Fehlerbehebung bei häufig auftretenden Verbindungsproblemen. Sie müssen Mitglied einer Administratorgruppe sein, um Connectoren anzeigen oder bearbeiten zu können.

Vergewissern Sie sich, dass Ihre IAM-Rolle über die richtige benutzerdefinierte Vertrauensrichtlinie und das richtige Tag verfügt

Stellen Sie beim Einrichten der IAM-Rolle für Ihren Connector sicher, dass die benutzerdefinierte Vertrauensrichtlinie ordnungsgemäß konfiguriert ist, um Zugriff auf App Studio zu gewähren. Diese benutzerdefinierte Vertrauensrichtlinie ist weiterhin erforderlich, wenn sich die AWS Ressourcen in demselben AWS Konto befinden, mit dem App Studio eingerichtet wurde.

- Stellen Sie sicher, dass die AWS Kontonummer im `Principal` Abschnitt der AWS Konto-ID des Kontos entspricht, das zur Einrichtung von App Studio verwendet wurde. Diese Kontonummer ist nicht immer das Konto, in dem sich die Ressource befindet.
- Stellen Sie sicher`"aws:PrincipalTag/IsAppStudioAccessRole": "true"`, dass der `sts:AssumeRole` Abschnitt ordnungsgemäß hinzugefügt wurde.

```
{
  "Version": "2012-10-17",
  "Statement": [
  {
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:PrincipalTag/IsAppStudioAccessRole": "true"
      }
    }
  }
  ]
}
```

Stellen Sie außerdem sicher, dass der IAM-Rolle ein Tag mit dem folgenden Schlüssel und Wert hinzugefügt wurde. Weitere Informationen zum Hinzufügen von Tags finden Sie unter Tag-IAM-Rollen:

> **Note**
>
> Beachten Sie, dass sich `IsAppStudioDataAccessRole` der Wert des Tags geringfügig von dem Wert in der benutzerdefinierten Vertrauensrichtlinie () `IsAppStudioAccessRole` unterscheidet.

- Schlüssel: `IsAppStudioDataAccessRole`
- Value (Wert): `true`

Überprüfen Sie die Konfiguration der Ressourcen in dem Produkt oder Dienst, mit dem Ihr Connector eine Verbindung herstellt. Einige Ressourcen, wie z. B. Amazon Redshift Redshift-Tabellen, erfordern eine zusätzliche Konfiguration, um sie mit App Studio verwenden zu können.

Überprüfen Sie Ihre Connector-Konfiguration. Gehen Sie für AWS Dienste zum Connector in App Studio und stellen Sie sicher, dass der richtige Amazon-Ressourcenname (ARN) angegeben ist und dass die angegebene AWS Region die ist, die Ihre Ressourcen enthält.

Überprüfen Sie, ob Ihre IAM-Rolle über die richtigen Berechtigungen verfügt

Um App Studio Zugriff auf AWS Ressourcen zu gewähren, müssen Sie der von Ihrem Connector verwendeten IAM-Rolle entsprechende Berechtigungen zuweisen. Die erforderlichen Berechtigungen gelten nur für den Dienst, die Ressource und die auszuführenden Aktionen. Beispielsweise erfordert das Lesen von Daten aus einer Amazon Redshift Redshift-Tabelle andere Berechtigungen als das Hochladen eines Objekts in einen Amazon S3 S3-Bucket. Weitere Informationen finden Sie im entsprechenden Thema Connect zu AWS Diensten her unter.

Fehlerbehebung Amazon Redshift Redshift-Konnektoren

Dieser Abschnitt enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme mit Amazon Redshift Redshift-Konnektoren. Informationen zur Konfiguration von Amazon Redshift Redshift-Konnektoren und -Ressourcen finden Sie unterConnect zu Amazon Redshift her.

1. Stellen Sie sicher, dass der `Isolated Session` Schalter im Amazon Redshift Redshift-Editor aktiviert ist. `OFF` Diese Einstellung ist erforderlich, um die Sichtbarkeit von Datenänderungen zu ermöglichen, die von anderen Benutzern vorgenommen wurden, z. B. in einer App Studio-App.
2. Stellen Sie sicher, dass die entsprechenden Berechtigungen für die Amazon Redshift Redshift-Tabelle erteilt wurden.

3. Stellen Sie in der Konnektorkonfiguration sicher, dass der entsprechende Berechnungstyp (`Provisioned`oder`Serverless`) ausgewählt ist, der dem Amazon Redshift Redshift-Tabellentyp entspricht.

Fehlerbehebung bei Aurora-Anschlüssen

Dieser Abschnitt enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme mit Aurora-Anschlüssen. Informationen zur Konfiguration von Aurora-Konnektoren und Ressourcen finden Sie unterConnect zu Amazon Aurora her.

1. Stellen Sie sicher, dass bei der Erstellung der Tabelle die entsprechende und unterstützte Aurora-Version ausgewählt wurde.
2. Stellen Sie sicher, dass die Amazon RDS-Daten-API aktiviert ist, da dies eine Voraussetzung ist, damit App Studio Operationen an den Aurora-Tabellen ausführen kann. Weitere Informationen finden Sie unter Amazon RDS Data API aktivieren.
3. Stellen Sie sicher, dass die AWS Secrets Manager Berechtigungen erteilt wurden.

Problembehandlung bei DynamoDB-Konnektoren

Dieser Abschnitt enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme mit DynamoDB-Konnektoren. Informationen zur Konfiguration von DynamoDB-Konnektoren und Ressourcen finden Sie unter. Connect zu Amazon DynamoDB her

Wenn Ihre DynamoDB-Tabellenschemas bei der Erstellung des Connectors nicht angezeigt werden, kann dies daran liegen, dass Ihre DynamoDB-Tabelle mit einem vom Kunden verwalteten Schlüssel (CMK) verschlüsselt ist und ohne Berechtigungen zum Beschreiben des Schlüssels und Entschlüsseln der Tabelle nicht auf die Tabellendaten zugegriffen werden kann. Um einen DynamoDB-Connector mit einer mit einem CMK verschlüsselten Tabelle zu erstellen, müssen Sie Ihrer IAM-Rolle die `kms:describeKey` Berechtigungen `kms:decrypt` und hinzufügen.

Fehlerbehebung Amazon S3 S3-Konnektoren

Dieser Abschnitt enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme mit Amazon S3 S3-Connectoren. Informationen zur Konfiguration von Amazon S3 S3-Konnektoren und Ressourcen finden Sie unterConnect zu Amazon Simple Storage Service (Amazon S3) her.

Zu den allgemeinen Anleitungen zur Fehlerbehebung gehört die Überprüfung der folgenden Punkte:

1. Stellen Sie sicher, dass der Amazon S3 S3-Connector mit der AWS Region konfiguriert ist, in der sich die Amazon S3 S3-Ressourcen befinden.
2. Stellen Sie sicher, dass die IAM-Rolle korrekt konfiguriert ist.
3. Stellen Sie im Amazon S3 S3-Bucket sicher, dass die CORS-Konfiguration die entsprechenden Berechtigungen gewährt. Weitere Informationen finden Sie unter Schritt 1: Amazon S3 S3-Ressourcen erstellen und konfigurieren.

Fehler beim Hochladen der Amazon S3 S3-Datei: Die vorsignierte URL konnte nicht berechnet werden

Beim Versuch, eine Datei mithilfe der Komponente S3 Upload in einen Amazon S3-Bucket hochzuladen, tritt möglicherweise der folgende Fehler auf:

```
Error while uploading file to S3: Failed to calculate presigned URL.
```

Dieser Fehler wird in der Regel entweder durch eine falsche IAM-Rollenkonfiguration oder durch eine falsche CORS-Konfiguration im Amazon S3 S3-Bucket verursacht und kann behoben werden, indem diese Konfigurationen mit den Informationen in repariert werden. Connect zu Amazon Simple Storage Service (Amazon S3) her

Problembehebung beim Veröffentlichen und Teilen von Anwendungen

Dieses Thema enthält Anleitungen zur Fehlerbehebung für häufig auftretende Probleme beim Veröffentlichen oder Teilen von App Studio-Anwendungen.

Ich sehe im Dialogfeld „Teilen“ keine neu erstellten App-Rollen

Neu erstellte Rollen auf App-Ebene werden erst im Dialogfeld „Teilen“ angezeigt, nachdem die App erneut veröffentlicht wurde. Veröffentlichen Sie die App, nachdem die neuen Rollen erstellt wurden, um sie zu verwenden.

Ich habe keine E-Mail erhalten, als die Veröffentlichung meiner App abgeschlossen war

Nur der App-Besitzer erhält eine E-Mail, wenn eine App veröffentlicht wird.

Die Endbenutzer meiner App können nicht auf die veröffentlichte App zugreifen

Wenn Ihre Endbenutzer nicht auf Ihre veröffentlichte App zugreifen können und beim Versuch, darauf zuzugreifen, eine `Forbidden` Nachricht erhalten, ist es wahrscheinlich, dass die veröffentlichte App nicht mit den Benutzern geteilt wird, die versuchen, darauf zuzugreifen. Veröffentlichte Apps müssen für Gruppen freigegeben werden, um den Benutzern in den Gruppen Zugriff zu gewähren.

Weitere Informationen zum Freigeben von Anwendungen finden Sie unter Gemeinsame Nutzung veröffentlichter Anwendungen.

Sicherheit in AWS App Studio

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das Modell der geteilten Verantwortung beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Die Wirksamkeit unserer Sicherheitsfunktionen wird regelmäßig von externen Prüfern im Rahmen des AWS -Compliance-Programms getestet und überprüft. Weitere Informationen zu den Compliance-Programmen, die für App Studio gelten, finden Sie unter AWS Services im Umfang nach Compliance-Programmen.
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. In Ihre Verantwortung fallen außerdem weitere Faktoren, wie z. B. die Vertraulichkeit der Daten, die Anforderungen Ihrer Organisation sowie geltende Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von App Studio anwenden können. In den folgenden Themen erfahren Sie, wie Sie App Studio konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Außerdem erfahren Sie, wie Sie andere AWS Dienste nutzen können, die Ihnen bei der Überwachung und Sicherung Ihrer App Studio-Ressourcen helfen können.

Themen

- Sicherheitsüberlegungen und Abhilfemaßnahmen
- Datenschutz in AWS App Studio
- AWS App Studio und AWS Identity and Access Management (IAM)
- Konformitätsprüfung für App Studio AWS
- Resilienz in AWS App Studio
- Sicherheit der Infrastruktur in AWS App Studio
- Konfiguration und Schwachstellenanalyse in AWS App Studio
- Serviceübergreifende Confused-Deputy-Prävention
- Regionsübergreifender Datentransfer in AWS App Studio

Sicherheitsüberlegungen und Abhilfemaßnahmen

Sicherheitsüberlegungen

Beim Umgang mit Datenkonnektoren, Datenmodellen und veröffentlichten Anwendungen ergeben sich verschiedene Sicherheitsbedenken im Zusammenhang mit Datenexposition, Zugriffskontrolle und potenziellen Sicherheitslücken. Die folgende Liste enthält die wichtigsten Sicherheitsbedenken.

Unsachgemäße Konfiguration der IAM-Rollen

Eine falsche Konfiguration der IAM-Rollen für Datenkonnektoren kann zu unberechtigtem Zugriff und Datenlecks führen. Wenn Sie der IAM-Rolle eines Datenkonnektors zu viel Zulässigkeit gewähren, können unbefugte Benutzer möglicherweise auf vertrauliche Daten zugreifen und diese ändern.

Verwenden von IAM-Rollen zur Durchführung von Datenoperationen

Da Endbenutzer einer App Studio-App die in der Connector-Konfiguration angegebene IAM-Rolle übernehmen, um Aktionen auszuführen, erhalten diese Endbenutzer möglicherweise Zugriff auf Daten, auf die sie normalerweise keinen Zugriff haben.

Löschen von Datenkonnektoren veröffentlichter Anwendungen

Wenn ein Datenconnector gelöscht wird, werden die zugehörigen geheimen Anmeldeinformationen nicht automatisch aus veröffentlichten Anwendungen entfernt, die diesen Connector bereits verwenden. Wenn in diesem Szenario eine Anwendung mit bestimmten Connectoren veröffentlicht wurde und einer dieser Connectoren aus App Studio gelöscht wurde, funktioniert die veröffentlichte Anwendung weiterhin mit den zuvor gespeicherten Connector-Anmeldeinformationen. Es ist wichtig zu beachten, dass die veröffentlichte App trotz des Löschens des Connectors davon unberührt bleibt und betriebsbereit bleibt.

Bearbeitung von Datenkonnektoren in veröffentlichten Anwendungen

Wenn ein Datenconnector bearbeitet wird, werden die Änderungen nicht automatisch in veröffentlichten Anwendungen widergespiegelt, die diesen Connector verwenden. Wenn eine Anwendung mit bestimmten Connectoren veröffentlicht wurde und einer dieser Connectoren in App Studio geändert wurde, verwendet die veröffentlichte Anwendung weiterhin die zuvor gespeicherte Connectorkonfiguration und die Anmeldeinformationen. Um die aktualisierten Connector-Änderungen zu übernehmen, muss die Anwendung erneut veröffentlicht werden. Bis zur erneuten Veröffentlichung bleibt die App fehlerhaft und funktionsunfähig oder unbeeinflusst und betriebsbereit, spiegelt aber nicht die neuesten Connector-Änderungen wider.

Empfehlungen zur Minderung von Sicherheitsrisiken

In diesem Abschnitt werden Empfehlungen zur Risikominderung zur Vermeidung von Sicherheitsrisiken aufgeführt, die im vorherigen Abschnitt zu Sicherheitsüberlegungen beschrieben wurden.

1. Richtige IAM-Rollenkonfiguration: Stellen Sie sicher, dass die IAM-Rollen für Data Connectors korrekt nach dem Prinzip der geringsten Rechte konfiguriert sind, um unbefugten Zugriff und Datenlecks zu verhindern.
2. Eingeschränkter App-Zugriff: Teilen Sie Ihre Apps nur mit Benutzern, die berechtigt sind, die Anwendungsdaten einzusehen oder Aktionen mit ihnen durchzuführen.
3. Veröffentlichung von Apps: Stellen Sie sicher, dass Apps immer dann erneut veröffentlicht werden, wenn ein Connector aktualisiert oder gelöscht wird.

Datenschutz in AWS App Studio

Das Modell der AWS gemeinsamen Verantwortung und geteilter Verantwortung gilt für den Datenschutz in AWS App Studio. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der alle Systeme laufen AWS Cloud. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter Häufig gestellte Fragen zum Datenschutz. Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.

- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS App Studio oder anderen Geräten arbeiten und die Konsole, die API oder AWS-Services verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Datenverschlüsselung

App Studio speichert und überträgt Daten sicher, indem es Daten im Ruhezustand und bei der Übertragung verschlüsselt.

Verschlüsselung im Ruhezustand

„Verschlüsselung im Ruhezustand“ bezieht sich auf den Schutz Ihrer Daten vor unbefugtem Zugriff durch deren Verschlüsselung während der Speicherung. App Studio bietet standardmäßig Verschlüsselung im Ruhezustand mithilfe von AWS KMS Schlüsseln, und Sie müssen keine zusätzliche Konfiguration für die Datenverschlüsselung im Ruhezustand vornehmen.

App Studio speichert die folgenden Daten für Ihre Anwendungen sicher: Quellcode, Build-Artefakte, Metadaten und Berechtigungsinformationen.

Bei der Verwendung von Datenquellen, die mit einem vom AWS KMS Kunden verwalteten Schlüssel (CMK) verschlüsselt sind, werden App Studio-Ressourcen weiterhin mit einem AWS verwalteten Schlüssel verschlüsselt, wohingegen die Daten in den verschlüsselten Datenquellen vom CMK verschlüsselt werden. Weitere Informationen zur Verwendung verschlüsselter Datenquellen in App Studio-Apps finden Sie unter. Verwenden Sie verschlüsselte Datenquellen mit CMKs

App Studio verwendet Amazon CloudFront , um Ihre App für Ihre Nutzer bereitzustellen. CloudFront verwendet SSDs , die für Edge-Standorte (POPs) verschlüsselt sind, und verschlüsselte EBS-Volumes für regionale Edge-Caches (RECs). Der Funktionscode und die Konfiguration in CloudFront Functions werden immer in einem verschlüsselten Format auf dem am Edge verschlüsselten Standort und SSDs an anderen Speicherorten gespeichert POPs, die von verwendet werden. CloudFront

Verschlüsselung während der Übertragung

Der Begriff „Verschlüsselung während der Übertragung“ bedeutet, dass Ihre Daten davor geschützt werden, abgefangen zu werden, während sie zwischen Kommunikationsendpunkten verschoben werden. App Studio bietet standardmäßig Verschlüsselung für Daten während der Übertragung. Die gesamte Kommunikation zwischen Kunden und App Studio sowie zwischen App Studio und seinen nachgelagerten Abhängigkeiten wird durch TLS-Verbindungen geschützt, die mit dem Signature Version 4-Signaturprozess signiert wurden. Alle App Studio-Endpunkte verwenden SHA-256-Zertifikate, die von einer AWS Certificate Manager privaten Zertifizierungsstelle verwaltet werden.

Schlüsselverwaltung

App Studio unterstützt die Verwaltung von Verschlüsselungsschlüsseln nicht.

Datenschutz für den Datenverkehr zwischen Netzwerken

Wenn Sie eine Instanz in App Studio erstellen, wählen Sie die AWS Region aus, in der die Daten und Ressourcen für diese Instanz gespeichert werden. Artefakte und Metadaten beim Erstellen von Anwendungen verlassen diese AWS Region nie.

Beachten Sie jedoch die folgenden Informationen:

- Da App Studio Amazon verwendet, CloudFront um Ihre Anwendung bereitzustellen, und Lambda @Edge verwendet, um die Authentifizierung für Ihre Anwendung zu verwalten, würde auf einen begrenzten Satz von Authentifizierungsdaten, Autorisierungsdaten und Anwendungsmetadaten von CloudFront Edge-Standorten aus zugegriffen werden, die sich in einer anderen Region befinden könnten.
- AWS App Studio überträgt Daten zwischen AWS Regionen, um bestimmte generative KI-Funktionen im Service zu aktivieren. Weitere Informationen zu den Funktionen, die durch regionsübergreifende Datenübertragungen aktiviert werden, zur Art der Daten, die zwischen Regionen übertragen werden, und zur Deaktivierung finden Sie unterRegionsübergreifender Datentransfer in AWS App Studio.

AWS App Studio und AWS Identity and Access Management (IAM)

In AWS App Studio verwalten Sie den Zugriff und die Berechtigungen im Service, indem Sie Gruppen in IAM Identity Center der entsprechenden Rolle in App Studio zuweisen. Die Berechtigungen der Gruppenmitglieder werden durch die zugewiesene Rolle bestimmt und nicht durch die Konfiguration von Benutzern, Rollen oder Berechtigungen direkt in AWS Identity and Access Management (IAM). Weitere Informationen zur Verwaltung von Zugriff und Berechtigungen in App Studio finden Sie unterZugriff und Rollen in App Studio verwalten.

App Studio lässt sich in IAM integrieren, wenn eine Instanz zu Abrechnungszwecken verifiziert wird und wenn eine Verbindung zu einem AWS Konto besteht, um Ressourcen in diesem AWS Konto zu erstellen und zu verwenden. Informationen zur Verbindung von App Studio mit anderen AWS Diensten zur Verwendung in Ihren Anwendungen finden Sie unter. Connect zu AWS Diensten her

Wenn Sie eine Instanz in App Studio erstellen, müssen Sie ein AWS Konto als Abrechnungs- und Verwaltungskonto für Ihre Instanz verbinden. Um wichtige Funktionen zu aktivieren, erstellt App Studio auch IAM-Dienstrollen, um dem Dienst die erforderlichen Berechtigungen zur Ausführung von Aufgaben in Ihrem Namen zu gewähren.

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um App Studio-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

Themen

- Identitätsbasierte Richtlinien für App Studio
- Ressourcenbasierte Richtlinien in App Studio
- Richtlinienaktionen für App Studio
- Richtlinienressourcen für App Studio
- Bedingungsschlüssel für Richtlinien für App Studio
- ACLs in App Studio
- ABAC mit App Studio
- Temporäre Anmeldeinformationen mit App Studio verwenden
- Serviceübergreifende Prinzipalberechtigungen für App Studio
- Servicerollen für App Studio
- Dienstbezogene Rollen für App Studio

- AWS verwaltete Richtlinien für AWS App Studio
- Mit Diensten verknüpfte Rollen für App Studio
- Beispiele für identitätsbasierte Richtlinien für App Studio AWS

Bevor Sie IAM verwenden, um den Zugriff auf App Studio zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für die Verwendung mit App Studio verfügbar sind.

IAM-Funktionen, die Sie mit App Studio verwenden können AWS

IAM-Feature	App Studio-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Bedingungsschlüssel für die Richtlinie	Nein
ACLs	Nein
ABAC (Tags in Richtlinien)	Nein
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Ja
Service-verknüpfte Rollen	Ja

Einen allgemeinen Überblick darüber, wie App Studio und andere AWS Dienste mit den meisten IAM-Funktionen funktionieren, finden Sie im IAM-Benutzerhandbuch unter AWS Dienste, die mit IAM funktionieren.

Identitätsbasierte Richtlinien für App Studio

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der IAM-Referenz für JSON-Richtlinienelemente im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für App Studio

Beispiele für identitätsbasierte Richtlinien von App Studio finden Sie unter. Beispiele für identitätsbasierte Richtlinien für App Studio AWS

Ressourcenbasierte Richtlinien in App Studio

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen.

Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

Richtlinienaktionen für App Studio

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer Zugriff auf was hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der App Studio-Aktionen finden Sie unter Von AWS App Studio definierte Aktionen in der Serviceautorisierungsreferenz.

Richtlinienaktionen in App Studio verwenden vor der Aktion das folgende Präfix:

```
appstudio
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "appstudio:action1",
      "appstudio:action2"
         ]
```

Die folgende Anweisung listet alle Aktionen in App Studio auf:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWS App Studio permissions",
            "Effect": "Allow",
            "Action": [
                "appstudio:GetAccountStatus", // Required to get the current account's
 App Studio instance status
                "appstudio:GetEnablementJobStatus", // Required to get the status of an
 enablement job of an App Studio instance
                "appstudio:StartEnablementJob", // Required to start the enablement of
 an App Studio instance
                "appstudio:StartRollbackEnablementJob", // Required to disable an
 enabled App Studio instance
                "appstudio:StartTeamDeployment" // Required to start deployment in
 order to update the App Studio instance infrastructure
            ],
            "Resource": "*"
        }
    ]
}
```

Richtlinienressourcen für App Studio

Unterstützt Richtlinienressourcen: Ja

App Studio-Berechtigungen unterstützen nur einen Platzhalter (*) im `Resource` Element einer Richtlinie.

Bedingungsschlüssel für Richtlinien für App Studio

Unterstützt dienstspezifische Richtlinien-Bedingungsschlüssel: Nein

App Studio unterstützt keine Richtlinien-Bedingungsschlüssel.

ACLs in App Studio

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit App Studio

Unterstützt ABAC (Tags in Richtlinien): Nein

App Studio unterstützt keine attributebasierte Zugriffskontrolle (ABAC).

Temporäre Anmeldeinformationen mit App Studio verwenden

Unterstützt temporäre Anmeldeinformationen: Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services funktionieren AWS-Services , finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM.

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter Wechseln von einer Benutzerrolle zu einer IAM-Rolle (Konsole) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter Temporäre Sicherheitsanmeldeinformationen in IAM.

Serviceübergreifende Prinzipalberechtigungen für App Studio

Unterstützt Forward Access Sessions (FAS): Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem

Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter Zugriffssitzungen weiterleiten.

Servicerollen für App Studio

Unterstützt Servicerollen: Ja

Eine Servicerolle ist eine IAM-Rolle, die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service im IAM-Benutzerhandbuch.

AWS App Studio verwendet für einige Funktionen IAM-Dienstrollen, um App Studio die Erlaubnis zu erteilen, Aufgaben in Ihrem Namen auszuführen. Die Konsole erstellt automatisch Servicerollen für unterstützte Funktionen, wenn Sie App Studio einrichten.

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität von App Studio beeinträchtigen. Bearbeiten Sie Servicerollen nur, wenn App Studio eine Anleitung dazu bereitstellt.

Dienstbezogene Rollen für App Studio

Unterstützt dienstverknüpfte Rollen: Ja

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter AWS -Services, die mit IAM funktionieren. Suchen Sie in der Tabelle nach einem Service mit einem `Yes` in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

AWS verwaltete Richtlinien für AWS App Studio

Um Benutzern, Gruppen und Rollen Berechtigungen hinzuzufügen, ist es einfacher, AWS verwaltete Richtlinien zu verwenden, als Richtlinien selbst zu schreiben. Es erfordert Zeit und Fachwissen, um von Kunden verwaltete IAM-Richtlinien zu erstellen, die Ihrem Team nur die benötigten Berechtigungen bieten. Um schnell loszulegen, können Sie unsere AWS verwalteten Richtlinien verwenden. Diese Richtlinien decken allgemeine Anwendungsfälle ab und sind in Ihrem AWS-Konto verfügbar. Weitere Informationen zu AWS verwalteten Richtlinien finden Sie im IAM-Benutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS Dienste verwalten und aktualisieren AWS verwaltete Richtlinien. Sie können die Berechtigungen in AWS verwalteten Richtlinien nicht ändern. Services fügen einer von AWS verwalteten Richtlinien gelegentlich zusätzliche Berechtigungen hinzu, um neue Features zu unterstützen. Diese Art von Update betrifft alle Identitäten (Benutzer, Gruppen und Rollen), an welche die Richtlinie angehängt ist. Services aktualisieren eine von AWS verwaltete Richtlinie am ehesten, ein neues Feature gestartet wird oder neue Vorgänge verfügbar werden. Dienste entfernen keine Berechtigungen aus einer AWS verwalteten Richtlinie, sodass durch Richtlinienaktualisierungen Ihre bestehenden Berechtigungen nicht beeinträchtigt werden.

AWS Unterstützt außerdem verwaltete Richtlinien für Jobfunktionen, die sich über mehrere Dienste erstrecken. Die ReadOnlyAccess AWS verwaltete Richtlinie bietet beispielsweise schreibgeschützten Zugriff auf alle AWS Dienste und Ressourcen. Wenn ein Dienst eine neue Funktion startet, werden nur Leseberechtigungen für neue Operationen und Ressourcen AWS hinzugefügt. Eine Liste und Beschreibungen der Richtlinien für Auftragsfunktionen finden Sie in Verwaltete AWS -Richtlinien für Auftragsfunktionen im IAM-Leitfaden.

AWS verwaltete Richtlinie: AppStudioServiceRolePolicy

Sie können `AppStudioServiceRolePolicy` nicht an Ihre IAM-Entitäten anhängen. Diese Richtlinie ist mit einer dienstbezogenen Rolle verknüpft, die es App Studio ermöglicht, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen finden Sie unter Mit Diensten verknüpfte Rollen für App Studio.

Diese Richtlinie gewährt Berechtigungen, die es der dienstbezogenen Rolle ermöglichen, Ressourcen zu verwalten AWS .

Details zu Berechtigungen

Diese Richtlinie umfasst die folgenden Berechtigungen:

- `logs`- Erstellen Sie CloudWatch Protokollgruppen und Protokollstreams. Erteilt auch die Erlaubnis, Protokollereignisse in diesen Protokollgruppen und Streams zu erstellen.
- `secretsmanager`- Verwaltete Geheimnisse, die von App Studio verwaltet werden, erstellen, lesen, aktualisieren und löschen.
- `sso`- Rufen Sie Anwendungsinstanzen ab.
- `sso-directory`- Ruft Informationen über Benutzer ab und ruft die Liste der Mitglieder in Gruppen ab.

```
{
 "Version": "2012-10-17",
 "Statement": [
     {
         "Sid": "AppStudioResourcePermissionsForCloudWatch",
         "Effect": "Allow",
         "Action": [
             "logs:CreateLogGroup",
             "logs:CreateLogStream",
             "logs:PutLogEvents"
         ],
         "Resource": [
             "arn:aws:logs:*:*:log-group:/aws/appstudio/*"
         ],
         "Condition": {
             "StringEquals": {
                 "aws:ResourceAccount": "${aws:PrincipalAccount}"
             }
         }
     },
     {
         "Sid": "AppStudioResourcePermissionsForSecretsManager",
         "Effect": "Allow",
         "Action": [
             "secretsmanager:CreateSecret",
             "secretsmanager:DeleteSecret",
             "secretsmanager:DescribeSecret",
             "secretsmanager:GetSecretValue",
             "secretsmanager:PutSecretValue",
```

```
            "secretsmanager:UpdateSecret",
            "secretsmanager:TagResource"
        ],
        "Resource": "arn:aws:secretsmanager:*:*:secret:appstudio-*",
        "Condition": {
            "ForAllValues:StringEquals": {
                "aws:TagKeys": [
                    "IsAppStudioSecret"
                ]
            },
            "StringEquals": {
                "aws:ResourceAccount": "${aws:PrincipalAccount}",
                "aws:ResourceTag/IsAppStudioSecret": "true"
            }
        }
    },
    {
        "Sid": "AppStudioResourcePermissionsForManagedSecrets",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:DeleteSecret",
            "secretsmanager:DescribeSecret",
            "secretsmanager:GetSecretValue",
            "secretsmanager:PutSecretValue",
            "secretsmanager:UpdateSecret"
        ],
        "Resource": "arn:aws:secretsmanager:*:*:secret:appstudio!*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceAccount": "${aws:PrincipalAccount}",
                "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
"appstudio"
            }
        }
    },
    {
        "Sid": "AppStudioResourceWritePermissionsForManagedSecrets",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:CreateSecret"
        ],
        "Resource": "arn:aws:secretsmanager:*:*:secret:appstudio!*",
        "Condition": {
            "StringEquals": {
```

```
                "aws:ResourceAccount": "${aws:PrincipalAccount}"
            }
        }
    },
    {
        "Sid": "AppStudioResourcePermissionsForSSO",
        "Effect": "Allow",
        "Action": [
            "sso:GetManagedApplicationInstance",
            "sso-directory:DescribeUsers",
            "sso-directory:ListMembersInGroup"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:ResourceAccount": "${aws:PrincipalAccount}"
            }
        }
    }
 ]
}
```

App Studio aktualisiert AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für App Studio an, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen.

Änderung	Beschreibung	Datum
AppStudioServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	App Studio hat neue Berechtigungen hinzugefügt, um die Verwaltung von verwalteten Geheimnissen in App Studio zu ermöglichen AWS Secrets Manager.	14. März 2025
App Studio hat begonnen, Änderungen zu verfolgen	App Studio hat damit begonnen, Änderungen für seine AWS verwalteten Richtlinien nachzuverfolgen.	28. Juni 2024

Mit Diensten verknüpfte Rollen für App Studio

App Studio verwendet AWS Identity and Access Management dienstverknüpfte Rollen (IAM). Eine dienstverknüpfte Rolle ist eine einzigartige Art von IAM-Rolle, die direkt mit App Studio verknüpft ist. Dienstbezogene Rollen sind von App Studio vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstverknüpfte Rolle erleichtert die Einrichtung von App Studio, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. App Studio definiert die Berechtigungen seiner dienstbezogenen Rollen, und sofern nicht anders definiert, kann nur App Studio seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre App Studio-Ressourcen geschützt, da Sie die Zugriffsberechtigung für die Ressourcen nicht versehentlich entfernen können.

Inhalt

- Mit dem Dienst verknüpfte Rollenberechtigungen für App Studio
- Eine dienstverknüpfte Rolle für App Studio erstellen
- Eine dienstverknüpfte Rolle für App Studio bearbeiten
- Löschen einer dienstverknüpften Rolle für App Studio

Mit dem Dienst verknüpfte Rollenberechtigungen für App Studio

App Studio verwendet die angegebene dienstverknüpfte Rolle. `AWSServiceRoleForAppStudio` Es handelt sich um eine dienstbezogene Rolle, die App Studio benötigt, um AWS Dienste dauerhaft zu verwalten und die Anwendungserstellung aufrechtzuerhalten.

Die `AWSServiceRoleForAppStudio` dienstverknüpfte Rolle verwendet die folgende Vertrauensrichtlinie, die nur dem Dienst vertraut: `appstudio-service.amazonaws.com`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
```

```
                "Service": "appstudio-service.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

In Bezug auf Berechtigungen bietet die `AWSServiceRoleForAppStudio` dienstverknüpfte Rolle Berechtigungen für die folgenden Dienste:

- Amazon CloudWatch: Zum Senden von Protokollen und Metriken für die Nutzung von App Studio.
- AWS Secrets Manager: Zur Verwaltung von Anmeldeinformationen für Konnektoren in App Studio, die verwendet werden, um Apps mit anderen Diensten zu verbinden.
- IAM Identity Center: Für schreibgeschützten Zugriff zur Verwaltung des Benutzerzugriffs.

Insbesondere `AWSServiceRoleForAppStudio` werden die mit gewährten Berechtigungen durch die beigefügte `AppStudioServiceRolePolicy` verwaltete Richtlinie definiert. Weitere Informationen zur verwalteten Richtlinie, einschließlich der darin enthaltenen Berechtigungen, finden Sie unterAWS verwaltete Richtlinie: AppStudioServiceRolePolicy.

Eine dienstverknüpfte Rolle für App Studio erstellen

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine App Studio-Instanz erstellen, erstellt App Studio die dienstverknüpfte Rolle für Sie.

Wenn Sie diese dienstverknüpfte Rolle löschen, wird empfohlen, eine App Studio-Instanz zu erstellen, damit automatisch eine weitere für Sie erstellt wird.

Dies ist zwar nicht erforderlich, Sie können aber auch die IAM-Konsole verwenden oder dienstverknüpfte Rollen erstellen AWS CLI , indem Sie eine dienstverknüpfte Rolle mit dem `appstudio-service.amazonaws.com` Dienstnamen erstellen, wie im oben gezeigten Vertrauensrichtlinien-Snippet dargestellt. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Eine dienstverknüpfte Rolle für App Studio bearbeiten

App Studio erlaubt es Ihnen nicht, die `AWSServiceRoleForAppStudio` dienstverknüpfte Rolle zu bearbeiten. Nachdem Sie eine serviceverknüpfte Rolle erstellt haben, können Sie den Namen der Rolle nicht mehr ändern, da verschiedene Entitäten auf die Rolle verweisen könnten. Sie können

jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer dienstverknüpften Rolle für App Studio

Sie müssen die `AWSServiceRoleForAppStudio` Rolle nicht löschen. Wenn Sie die App Studio-Instanz löschen, bereinigt App Studio die Ressourcen und löscht die mit dem Dienst verknüpfte Rolle automatisch.

Obwohl dies nicht empfohlen wird, können Sie die IAM-Konsole oder die verwenden, AWS CLI um die dienstverknüpfte Rolle zu löschen. Dazu müssen Sie zuerst die Ressourcen für Ihre serviceverknüpfte Rolle bereinigen und dann können Sie sie löschen.

> **Note**
>
> Wenn App Studio die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie die -servicegebundene Rolle mit IAM

1. Löschen Sie die Anwendungen und Connectors aus Ihrer App Studio-Instanz.
2. Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um die `AWSServiceRoleForAppStudio`-serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Beispiele für identitätsbasierte Richtlinien für App Studio AWS

Standardmäßig sind Benutzer und Rollen nicht berechtigt, App Studio-Ressourcen zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS Management Console, AWS Command Line Interface (AWS CLI) oder AWS API ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von IAM-Richtlinien (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den von App Studio definierten Aktionen und Ressourcentypen, einschließlich des Formats ARNs für die einzelnen Ressourcentypen, finden Sie unter Aktionen, Ressourcen und Bedingungsschlüssel für AWS App Studio in der Referenz zur Serviceautorisierung.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der App Studio-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer
- Beispiel 1: Erlauben Sie Benutzern, eine App Studio-Instanz einzurichten
- Beispiel 2: Benutzern die Einrichtung einer App Studio-Instanz verweigern

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand App Studio-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um Ihren Benutzern und Workloads zunächst Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS -verwaltete Richtlinien oder AWS -verwaltete Richtlinien für Auftrags-Funktionen im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben,

um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung mit IAM Access Analyzer im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Sicherer API-Zugriff mit MFA im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der App Studio-Konsole

Um auf die AWS App Studio-Konsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den App Studio-Ressourcen in Ihrem aufzulisten und anzuzeigen AWS-Konto. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die App Studio-Konsole weiterhin verwenden können, fügen Sie den Entitäten auch die App Studio `ConsoleAccess` - oder `ReadOnly` AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der API AWS CLI oder AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Beispiel 1: Erlauben Sie Benutzern, eine App Studio-Instanz einzurichten

Das folgende Beispiel zeigt eine identitätsbasierte Richtlinie, die es einer Rolle ermöglicht, eine App Studio-Instanz einzurichten.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "appstudio:GetAccountStatus",
            "appstudio:GetEnablementJobStatus",
            "appstudio:StartEnablementJob",
            "appstudio:StartRollbackEnablementJob",
            "appstudio:StartTeamDeployment"
        ],
        "Resource": "*"
    }]
}
```

Beispiel 2: Benutzern die Einrichtung einer App Studio-Instanz verweigern

Das folgende Beispiel zeigt eine identitätsbasierte Richtlinie, mit der einer Rolle die Einrichtung einer App Studio-Instanz verweigert wird.

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Deny",
        "Action": [
            "appstudio:*"
        ],
        "Resource": "*"
    }]
}
```

Konformitätsprüfung für App Studio AWS

Informationen darüber, ob AWS-Service ein AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services

unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Compliance und Governance im Bereich Sicherheit – In diesen Anleitungen für die Lösungsimplementierung werden Überlegungen zur Architektur behandelt. Außerdem werden Schritte für die Bereitstellung von Sicherheits- und Compliance-Features beschrieben.
- Referenz für berechtigte HIPAA-Services – Listet berechtigte HIPAA-Services auf. Nicht alle AWS-Services sind HIPAA-fähig.
- AWS Compliance-Ressourcen — Diese Sammlung von Arbeitsmappen und Leitfäden gilt möglicherweise für Ihre Branche und Ihren Standort.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS-Services und die Leitlinien den Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zugeordnet.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Auf diese AWS-Service Weise erhalten Sie einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Die Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerelementreferenz.
- Amazon GuardDuty — Dies AWS-Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen wie PCI DSS zu erfüllen, indem es die in bestimmten Compliance-Frameworks vorgeschriebenen Anforderungen zur Erkennung von Eindringlingen erfüllt.

- AWS Audit Manager— Auf diese AWS-Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Resilienz in AWS App Studio

Die AWS globale Infrastruktur basiert auf Availability AWS-Regionen Zones. AWS-Regionen bieten mehrere physisch getrennte und isolierte Availability Zones, die über Netzwerke mit niedriger Latenz, hohem Durchsatz und hoher Redundanz miteinander verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu Availability Zones AWS-Regionen und Availability Zones finden Sie unter AWS Globale Infrastruktur.

Zusätzlich zur AWS globalen Infrastruktur bietet AWS App Studio mehrere Funktionen zur Unterstützung Ihrer Datenausfallsicherheit und Backup-Anforderungen.

Sicherheit der Infrastruktur in AWS App Studio

Als verwalteter Service ist AWS App Studio durch die AWS globalen Netzwerksicherheitsverfahren geschützt, die im Whitepaper Amazon Web Services: Sicherheitsprozesse im Überblick beschrieben sind.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf App Studio zuzugreifen. Clients müssen mindestens Transport Layer Security (TLS) 1.2 unterstützen, TLS 1.3 wird jedoch empfohlen. Clients müssen außerdem Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) unterstützen. Die meisten modernen Systemen wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Konfiguration und Schwachstellenanalyse in AWS App Studio

Konfiguration und IT-Kontrollen liegen in der gemeinsamen Verantwortung AWS von Ihnen, unserem Kunden. Weitere Informationen finden Sie im Modell der AWS gemeinsamen Verantwortung.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann ein dienstübergreifendes Identitätswechsels zu einem Problem mit dem verwirrten Stellvertreter führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der aufrufende Service kann manipuliert werden, um seine Berechtigungen zu verwenden, um Aktionen auf die Ressourcen eines anderen Kunden auszuführen, für die er sonst keine Zugriffsberechtigung haben sollte. Um dies zu verhindern, bietet AWS Tools, mit denen Sie Ihre Daten für alle Services mit Serviceprinzipalen schützen können, die Zugriff auf Ressourcen in Ihrem Konto erhalten haben.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn`und die `aws:SourceAccount`globalen Bedingungsschlüssel in Ressourcenrichtlinien zu verwenden, um die Berechtigungen einzuschränken, die der Ressource einen anderen Dienst gewähren. Verwenden Sie `aws:SourceArn`, wenn Sie nur eine Ressource mit dem betriebsübergreifenden Zugriff verknüpfen möchten. Verwenden Sie `aws:SourceAccount`, wenn Sie zulassen möchten, dass Ressourcen in diesem Konto mit der betriebsübergreifenden Verwendung verknüpft werden.

Der effektivste Weg, um sich vor dem Confused-Deputy-Problem zu schützen, ist die Verwendung des globalen Bedingungskontext-Schlüssels `aws:SourceArn` mit dem vollständigen ARN der Ressource. Wenn Sie den vollständigen ARN der Ressource nicht kennen oder wenn Sie mehrere Ressourcen angeben, verwenden Sie den globalen Kontextbedingungsschlüssel `aws:SourceArn` mit Platzhalterzeichen (*) für die unbekannten Teile des ARN. Beispiel, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Wenn der `aws:SourceArn`-Wert die Konto-ID nicht enthält, z. B. einen Amazon-S3-Bucket-ARN, müssen Sie beide globale Bedingungskontextschlüssel verwenden, um Berechtigungen einzuschränken.

Der `aws:SourceArn`-Wert muss ResourceDescription lauten.

Das folgende Beispiel zeigt, wie Sie die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globale Bedingung verwenden können, um das Problem des verwirrten Stellvertreters zu vermeiden.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "servicename.amazonaws.com"
    },
    "Action": "servicename:ActionName",
    "Resource": [
      "arn:aws:servicename:::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:servicename:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Regionsübergreifender Datentransfer in AWS App Studio

AWS App Studio überträgt Daten zwischen AWS Regionen, um bestimmte generative KI-Funktionen im Service zu aktivieren. Dieses Thema enthält Informationen zu den Funktionen, die durch regionsübergreifende Datenübertragungen aktiviert werden, zur Art der Daten, die zwischen Regionen übertragen werden, und zur Deaktivierung.

Die folgenden Funktionen werden durch die regionsübergreifende Datenübertragung aktiviert und sind in Ihrer Instanz nicht verfügbar, wenn Sie sich abmelden:

1. Das Erstellen einer App mit KI wird verwendet, um die Erstellung von Apps anzukurbeln, indem Ihre App in natürlicher Sprache beschrieben und Ressourcen für Sie erstellt werden.
2. Der KI-Chat im Anwendungsstudio, in dem Fragen zum Erstellen, Veröffentlichen und Teilen von Apps gestellt wurden.

Die folgenden Daten werden zwischen Regionen übertragen:

1. Die Eingabeaufforderungen oder Benutzereingaben aus den zuvor beschriebenen Funktionen.

Um die regionsübergreifende Datenübertragung und die dadurch aktivierten Funktionen zu deaktivieren, gehen Sie wie folgt vor, um das Abmeldeformular von der Konsole aus auszufüllen:

1. Öffnen Sie die App Studio-Konsole unter https://console.aws.amazon.com/appstudio/.
2. Wählen Sie „Datenübertragung deaktivieren“.
3. Geben Sie Ihre AWS Konto-ID und Ihre E-Mail-Adresse ein.
4. Wählen Sie Absenden aus.
5. Nach dem Absenden wird Ihre Anfrage zur Deaktivierung der regionsübergreifenden Datenübertragung bearbeitet, was bis zu 60 Tage dauern kann.

Unterstützte Browser für AWS App Studio

Dieses Thema enthält Informationen zu den unterstützten und empfohlenen Browsern für AWS App Studio, einschließlich Browserunterstützung sowohl für Endbenutzer, die auf veröffentlichte Anwendungen zugreifen, als auch für Anwendungsgeneratoren.

Unterstützte und empfohlene Browser für die Erstellung von Anwendungen

Für ein optimales Erlebnis bei der Anwendungserstellung unterstützt App Studio Google Chrome und empfiehlt ausdrücklich die Verwendung von Google Chrome.

> **Note**
>
> Obwohl dies nicht empfohlen wird, können Sie Anwendungen auch mit anderen gängigen Webbrowsern wie Mozilla Firefox, Microsoft Edge oder Apple Safari für macOS erstellen. Beachten Sie jedoch, dass diese Browser nicht offiziell unterstützt oder validiert werden und Sie möglicherweise Einstellungen aktualisieren müssen, um auf einige Builder-Funktionen zugreifen zu können. Weitere Informationen finden Sie unter Aktualisieren Sie die Browsereinstellungen, um Apps in App Studio zu erstellen.

App Studio unterstützt nicht das Erstellen von Anwendungen auf mobilen Plattformen.

Unterstützte und empfohlene Browser für Endbenutzer von Anwendungen

Für Endbenutzer, die auf veröffentlichte Anwendungen zugreifen, empfiehlt App Studio dringend die Verwendung von Google Chrome oder Mozilla Firefox. Dies sind zwar die empfohlenen Browser, Endbenutzer können jedoch auch mit anderen gängigen Webbrowsern wie Microsoft Edge oder Apple Safari für macOS auf veröffentlichte Apps zugreifen.

Endbenutzer können auch von mobilen Plattformen aus auf veröffentlichte Anwendungen zugreifen.

Aktualisieren Sie die Browsereinstellungen, um Apps in App Studio zu erstellen

App Studio unterstützt und empfiehlt offiziell die Verwendung von Google Chrome zum Erstellen von Anwendungen. Wenn Sie jedoch andere Browser zum Erstellen von Anwendungen verwenden möchten, müssen Sie möglicherweise bestimmte Einstellungen oder Cookies im Zusammenhang mit Cross-Site-Tracking aktualisieren, um auf bestimmte Seiten in App Studio zugreifen zu können.

Für Mozilla Firefox: Um eine Vorschau von Anwendungen anzuzeigen, aktualisieren Sie die folgende Einstellung: `Firefox Settings > Privacy & Security > Enhanced Tracking Protection` auf`Custom > Cookies > Cross-site tracking cookies`.

Für Apple Safari für macOS: Um Anwendungen zu erstellen oder eine Vorschau anzuzeigen, deaktivieren Sie die folgende Einstellung:`Settings > Privacy > Prevent cross-site tracking`.

Kontingente für AWS App Studio

In der folgenden Tabelle werden Kontingente und Limits für AWS App Studio beschrieben.

Maximale Anzahl von Apps in einer App Studio-Instanz	20
Maximale Anzahl von Anwendungen, die in einer App Studio-Instanz in der Test- oder Produktionsumgebung veröffentlicht wurden. Eine einzelne Anwendung, die sowohl in der Test- als auch in der Produktionsumgebung veröffentlicht wurde, zählt als zwei veröffent lichte Anwendungen.	6
Maximale Anzahl verwalteter Entitäten pro App	20
Maximale Anzahl von Zeilen, die pro Abfrage zurückgegeben werden	3000
Maximale Anzahl von Zeilen mit Beispieldaten pro Entität	500
Maximale Laufzeit einer Automatisierung	2 Minuten. Automatisierungen, die länger als 2 Minuten laufen, schlagen fehl.
Maximale Größe der Automatisierungseingabe und -ausgabe	5 GB pro Eingang oder Ausgang.
Maximale Datengröße, die von einer Automatis ierung oder Datenaktion verwendet wird	450 MB pro Automatisierungs- oder Datenakti onslauf.
Seitennamen und Komponentennamen	Muss nicht leer und eindeutig sein. Darf nur Buchstaben, Zahlen, Unterstriche (_) und Dollarzeichen ($) enthalten. Darf keine Leerzeichen enthalten.

Dokumentenverlauf für das AWS App Studio-Benutzerhandbuch

In der folgenden Tabelle werden die Dokumentationsversionen für AWS App Studio beschrieben.

Änderung	Beschreibung	Datum
Neues Thema: Rollback der Version einer App für die Produktionsumgebung	Es wurden Informationen zum Rollback der veröffentlichten Version Ihrer App auf eine zuvor veröffentlichte Version hinzugefügt, falls Probleme festgestellt werden. Weitere Informationen finden Sie unter Rollback zu einer zuvor veröffentlichten Version.	10. April 2025
Neue Themen: Doppelte Apps, Seiten und Komponenten	Es wurden neue Themen mit Informationen zum Duplizieren von Apps, Seiten und Komponenten in App Studio hinzugefügt. Weitere Informationen finden Sie unter Duplizieren von Anwendungen, Duplizierenvon Seiten und Duplizieren von Komponenten.	7. April 2025
Neue Themen: Anwendungen zwischen App Studio-Instanzen importieren und exportieren	Es wurden neue Themen mit Informationen zum Importieren und Exportieren von Anwendungen zwischen App Studio-Instanzen hinzugefügt, einschließlich einer Liste von importierbaren Anwendungen, die von App Studio bereitgestellt werden und mit denen	30. März 2025

	Konzepte zur App-Erste llung erlernt werden können. Weitere Informationen finden Sie unter Anwendungen importieren und Anwendungen exportieren.	
Aktualisiertes Thema: AWS verwaltete Richtlinie: AppStudioServiceRolePolicy	Die `AppStudioServiceRo lePolicy` Berechtigungen wurden aktualisiert und es wurden Informationen zur Richtlinienbeschreibung für jeden Dienst hinzugefügt. Weitere Informationen finden Sie unter AWS Verwaltet e Richtlinie: AppStudio ServiceRolePolicy.	14. März 2025
Aktualisiertes Thema: Datenaktionen bearbeiten oder konfigurieren	Es wurden Informationen zu vorhandenen und neuen Operatoren hinzugefügt, die in Datenaktionsbeding ungen verwendet werden. Diese werden verwendet , um eine Teilmenge von Daten aus Ihrer Datenbank tabelle abzurufen, die den Bedingungen entspricht. Weitere Informationen finden Sie unter Datenaktionen bearbeiten oder konfigurieren.	27. Februar 2025

Aktualisiertes Thema: Verwenden JavaScript zum Schreiben von Ausdrücken	Informationen zum Referenzieren oder Aktualisieren von UI-Komponenten- und Tabellendaten mithilfe von Ausdrücken in Anwendungen wurden neu organisiert und hinzugefügt. Weitere Informationen finden Sie unter Verwenden JavaScript zum Schreiben von Ausdrücken in App Studio.	18. Februar 2025
Aktualisiertes Thema: Sicherheitseinstellungen für App-Inhalte	Es wurden Informationen zur Sicherheitseinstellung für Inhalte der App „Inhaltsquelle“ hinzugefügt. Sie können diese Einstellung verwenden, um die Domänen einzuschränken, in denen Ihre App Objekte auf Amazon S3 hochladen kann. Weitere Informationen finden Sie unter Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren.	14. Februar 2025
Neues Thema: Lambda-Funktionen in einer App Studio-App aufrufen	Es wurde ein kurzes Tutorial hinzugefügt, in dem detailliert beschrieben wird, wie Lambda-Funktionen in einer App Studio-App aufgerufen werden. Weitere Informationen finden Sie unter Aufrufen von Lambda-Funktionen.	24. Januar 2025

Neues Thema: Connect zu Amazon SES herstellen	Es wurden Anweisungen zum Erstellen eines Amazon SES SES-Connectors hinzugefügt, um den Service in App Studio-Apps zu verwenden. Weitere Informationen finden Sie unter Connect zu Amazon Simple Email Service herstellen.	16. Januar 2025
Aktualisiertes Thema: Erstmaliges Erstellen und Einrichten einer App Studio-Instanz	Es wurden Anweisungen zur Verwendung der Methode easy create hinzugefügt, um eine App Studio-Instanz zu erstellen, um schneller loslegen zu können. Weitere Informationen finden Sie unter Erstmaliges Erstellen und Einrichten einer App Studio-Instanz.	13. Dezember 2024
Neues Thema: Bewährte Methoden für den Umgang mit Datenabhängigkeiten und Zeitproblemen	Es wurde eine Dokumentation zur ordnungsgemäßen Verwaltung von Datenabhängigkeiten und Zeitproblemen in App Studio-Apps hinzugefügt. Weitere Informationen finden Sie unter Datenabhängigkeiten und Überlegungen zum Timing.	20. November 2024

Aktualisierte Themen: Bearbeiten Ihrer App mit KI	Es wurde eine Dokumentation mit Informationen zur Bearbeitung deiner App mithilfe des KI-Chat im Anwendungsstudio hinzugefügt. Weitere Informationen finden Sie unter Erstellen Ihrer App Studio-App mit generativer KI.	18. November 2024
Aktualisiertes Thema: Verwenden Sie KI, um JavaScript für Sie zu generieren	Die Referenz zu JavaScript Automatisierungsaktionen wurde aktualisiert und enthält nun Informationen zur Verwendung von KI zur Generierung JavaScript für Sie. Weitere Informationen finden Sie unter JavaScript Automatisierungsaktion.	18. November 2024
Aktualisiertes Thema: Erstellen Sie eine KI-Textzusammenfassungs-App mit Amazon Bedrock	Das Amazon Bedrock Prompt-Tutorial wurde aktualisiert, um die neu veröffentlichte GenAI Prompt-Aktion zu verwenden. Weitere Informationen finden Sie unter Erstellen einer KI-Textzusammenfassungs-App mit Amazon Bedrock.	18. November 2024
Neues Thema: Ändern Sie die Farben Ihrer App mit App-Themen	Es wurde ein Thema mit Informationen zum Ändern der Farben in Ihrer App mithilfe von App-Themen hinzugefügt. Weitere Informationen finden Sie unter Ändern der Farben in Ihrer App mit App-Themen.	18. November 2024

Neues Thema: Bewährte Methoden für Datenmodelle	Es wurde ein Thema mit bewährten Methoden für die Erstellung sicherer, robuster und skalierbarer Datenmodelle für die Verwendung in App Studio-Apps hinzugefügt. Weitere Informationen finden Sie unter Bewährte Methoden beim Entwerfen von Datenmodellen.	15. November 2024
Aktualisierte Themen: Verbindung zu AWS Diensten herstellen	Die Vertrauensrichtlinien wurden dahingehend aktualisiertsts:ExternalId , dass sie nun enthalten, was für IAM-Rollen erforderlich ist, die zur Erstellung von Konnektoren zu AWS Diensten verwendet werden. Weitere Informationen finden Sie unter Connect zu AWS Diensten herstellen.	13. November 2024
Neues Thema: Rollback oder Rückkehr zu einer zuvor veröffentlichten App-Version	Es wurde ein Thema mit Informationen zum Rollback oder zum Zurücksetzen einer Anwendung auf eine zuvor veröffentlichte Version hinzugefügt. Weitere Informationen finden Sie unter Rollback zu einer zuvor veröffentlichten Version.	13. November 2024

Neues Thema: Eine App Studio-Instanz löschen	Es wurde ein Thema hinzugefügt, das Informationen zum Löschen einer App Studio-Instanz enthält, einschließlich Anweisungen zum Löschen einer Instanz. Weitere Informationen finden Sie unter Löschen einer App Studio-Instanz.	12. November 2024
Neues Thema: Sicherheitseinstellungen für App-Inhalte aktualisieren	Es wurde ein Thema hinzugefügt, das Informationen zu den Sicherheitseinstellungen für App-Inhalte in App Studio enthält, einschließlich deren Aktualisierung. Weitere Informationen finden Sie unter Inhaltssicherheitseinstellungen Ihrer App anzeigen oder aktualisieren.	8. November 2024
Aktualisierte Themen: Sicherheit in AWS App Studio	Die Sicherheitsdokumentation wurde um Informationen zum Datenschutz und zur Interaktion von App Studio mit IAM erweitert. Weitere Informationen finden Sie unter Sicherheit in AWS App Studio.	6. November 2024

Aktualisiertes Thema: Kontingente in AWS App Studio	Die Dokumentation zu den Quotas und Limits des App Studio-Dienstes wurde aktualisiert, um falsche Werte zu korrigieren und einige Kontingente zu entfernen. Weitere Informationen finden Sie unter Kontingente in AWS App Studio.	21. Oktober 2024
Aktualisierte Themen: App Studio mit anderen AWS Diensten verbinden	Die Dokumentation für das Herstellen einer Verbindung zu AWS Diensten wurde aktualisiert, um den bewährten Sicherheitspraktiken besser gerecht zu werden. Es wurden Anweisungen und Beispiele bereitgestellt, wie App Studio die minimal erforderlichen Berechtigungen für die Dienste oder Ressourcen erteilt. Weitere Informationen finden Sie unter Connect zu AWS Diensten herstellen.	18. Oktober 2024
Aktualisiertes Thema: Versionsunterstützung zur Aurora-Connector-Dokumentation hinzugefügt	Der Aurora-Connector-Dokumentation wurde eine Liste unterstützter Versionen hinzugefügt. Weitere Informationen finden Sie unter Connect zu Amazon Aurora herstellen.	16. Oktober 2024

Neues Thema: Unterstützte Browser für App Studio	Es wurde ein Thema hinzugefügt, das Browserunterstützung und Empfehlungen für die Verwendung von App Studio enthält. Weitere Informationen finden Sie unter Unterstützte Browser.	10. Oktober 2024
Neues Thema: So funktioniert AWS App Studio	Es wurde ein Thema hinzugefügt, das die wichtigsten Konzepte der App-Entwicklung in App Studio erläutert, einschließlich Diagrammen und Screenshots. Weitere Informationen finden Sie unter So funktioniert App Studio.	10. Oktober 2024
Neues Thema: Seiten ordnen und organisieren	Es wurde ein Thema hinzugefügt, das Informationen zum Neuanordnen und Ausblenden oder Einblenden von Seiten in der Navigation einer Vorschau oder einer veröffentlichten App enthält. Weitere Informationen finden Sie unter Seiten anordnen und organisieren.	24. September 2024
Neues Thema: Kontingente in AWS App Studio	Es wurde ein Thema hinzugefügt, das die Dienstkontingente und -beschränkungen im Zusammenhang mit App Studio enthält. Weitere Informationen finden Sie unter Kontingente in AWS App Studio.	11. September 2024

Aktualisiertes Thema: Connect zu verschlüsselten DynamoDB-Tabellen herstellen	Es wurden Informationen hinzugefügt, z. B. erforderliche Berechtigungen, um DynamoDB-Tabellen, die mit AWS KMS vom Kunden verwalteten Schlüsseln (CMKs) verschlüsselt sind, mit App Studio zu verwenden. Weitere Informationen finden Sie unter Connect to DynamoDB.	6. September 2024
Aktualisierte Themen: Connect zu DynamoDB, Amazon Redshift und Aurora herstellen	Es wurden die erforderlichen Mindestberechtigungen für das Hinzufügen zu einer IAM-Rolle hinzugefügt, um DynamoDB-, Amazon Redshift- und Aurora-Ressourcen mit App Studio-Apps zu verwenden. Weitere Informationen finden Sie unter Connect zu AWS Diensten herstellen.	5. September 2024
Aktualisiertes Thema: Connect zu Amazon Aurora herstellen	Die Dokumentation für die Erstellung und Konfiguration von Amazon Aurora Aurora-Datenbanken und -Tabellen für die Verwendung mit App Studio-Apps wurde aktualisiert. Weitere Informationen finden Sie unter Connect zu Amazon Aurora herstellen.	5. September 2024

Neue und aktualisierte Themen: Fehlerbehebung und Debugging	Die Dokumentation zur Fehlerbehebung und zum Debuggen wurde erweitert, um allgemeine Probleme mit App Studio zu lösen, einschließlich Debugging-Informationen für die Erstellung von Anwendungen. Weitere Informationen finden Sie unter Problembehandlung und Debuggen von App Studio.	26. August 2024
Neues Thema: Tutorial: Erstellen Sie eine KI-Textzusammenfassungs-App mit Amazon Bedrock	Folgen Sie den Schritten im Tutorial, um eine App zu erstellen, die eine Eingabeaufforderung von Endbenutzern entgegennimmt, sie an Amazon Bedrock sendet und eine zusammengefasste Version zurücksendet und anzeigt. Weitere Informationen finden Sie unter Erstellen einer KI-Textzusammenfassungs-App mit Amazon Bedrock.	20. August 2024

Aktualisierte Themen: App Studio-Apps in der Vorschau anzeigen, veröffentlichen und teilen	Die Dokumentation zum Vorschauen, Veröffentlichen und Teilen wurde erweitert, um mehr Klarheit zu schaffen, der Benutzererfahrung im Service zu entsprechen und zusätzliche Informationen zu den Veröffentlichungsumgebungen und der Anzeige von Apps in ihnen bereitzustellen. Weitere Informationen finden Sie unter Anwendungen in der Vorschau anzeigen, veröffentlichen und teilen.	2. August 2024
Neues Thema: Eine App mit mehreren Benutzern erstellen	Die Dokumentation zum Vorschauen, Veröffentlichen und Teilen wurde erweitert, um mehr Klarheit zu schaffen, der Benutzererfahrung im Service zu entsprechen und zusätzliche Informationen zu den Veröffentlichungsumgebungen und zur Anzeige der darin enthaltenen Apps bereitzustellen. Weitere Informationen finden Sie unter Eine App mit mehreren Benutzern erstellen.	2. August 2024

Aktualisiertes Thema: App Studio mit AWS Diensten verbinden	Es wurden Informationen zum Erstellen und Bereitstellen von IAM-Rollen für den Zugriff auf AWS Ressourcen beim Erstellen eines Connector s für andere AWS Dienste hinzugefügt. Weitere Informati onen finden Sie unter Connect zu AWS Diensten mithilfe des Connectors für andere AWS Dienste.	29. Juli 2024
Aktualisiertes Thema: Fügen Sie Anweisungen zum Erstellen eines AWS Administr atorbenutzers als Teil der Einrichtung hinzu	In der Dokumentation zur Einrichtung von App Studio wurden Anweisung en zum Erstellen eines Administratorbenutzers für die Verwaltung von AWS Ressourcen hinzugefügt. Außerdem wurde die gesamte Connector-Dokumentation aktualisiert, um die Verwendun g dieses Benutzers zu empfehlen.	24. Juli 2024
Neues Thema: Connect zu Amazon Bedrock herstellen	Es wurde ein Thema mit Anweisungen zum Erstellen eines Connectors für Amazon Bedrock hinzugefü gt. Entwickler können den Connector verwenden, um Apps zu erstellen, die Amazon Bedrock verwenden. Weitere Informationen finden Sie unter Connect zu Amazon Bedrock herstellen.	24. Juli 2024

Erstversion	Erste Version des AWS App Studio-Benutzerhandbuchs	10. Juli 2024

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.