



使用者指南

AWS App Studio



AWS App Studio: 使用者指南

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

Table of Contents

什麼是 AWS App Studio ?	1
您是第一次使用 App Studio 嗎?	1
概念	2
管理員角色	2
應用程式 (應用程式)	2
自動化	3
自動化動作	3
Builder 角色	3
元件	3
連接器	3
開發環境	3
實體	4
執行個體	4
頁面	4
觸發條件	4
App Studio 的運作方式	5
將您的應用程式連線至其他 服務	5
設定應用程式的資料模型	6
建置應用程式的 UI	6
實作應用程式的邏輯或行為	7
應用程式的開發生命週期	8
進一步了解	9
設定並登入 App Studio	10
第一次建立和設定 App Studio 執行個體	10
註冊 AWS 帳戶	10
建立管理使用者以管理 AWS 資源	10
在 中建立 App Studio 執行個體 AWS 管理主控台	11
接受加入 App Studio 的邀請	15
開始使用	16
教學課程：使用 AI 產生應用程式	16
先決條件	17
步驟 1：建立 應用程式	17
步驟 2：探索您的新應用程式	18
步驟 3：預覽您的應用程式	19

後續步驟	20
教學課程：從空白應用程式開始建置	20
先決條件	22
步驟 1：建立 應用程式	22
步驟 2：建立實體以定義應用程式的資料	23
步驟 3：設計使用者介面 (UI) 和邏輯	25
步驟 4：預覽應用程式	28
步驟 5：將應用程式發佈至測試環境	28
後續步驟	29
管理員文件	30
使用群組和角色管理使用者存取	30
角色和許可	30
檢視群組	31
新增使用者或群組	31
變更群組的角色	32
移除使用者或群組	33
使用連接器連線至其他 服務	33
連線至 AWS 服務	34
連線至第三方服務	75
檢視、編輯和刪除連接器	82
刪除 App Studio 執行個體	82
建置器文件	84
教學	84
使用 Amazon Bedrock 建置文字摘要器應用程式	84
與 Amazon S3 互動	91
調用 Lambda 函式 ()	99
使用生成式 AI 建置您的應用程式	101
產生您的應用程式	102
建置或編輯您的應用程式	102
產生資料模型	102
產生範例資料	102
設定 AWS 服務的動作	103
模擬回應	103
在建置時向 AI 尋求協助	103
建立、編輯和刪除應用程式	103
建立應用程式	103

匯入應用程式	104
複製應用程式	108
編輯或建置應用程式	109
編輯先前發佈的應用程式版本	109
重新命名應用程式	110
刪除應用程式	110
預覽、發佈和共用應用程式	111
預覽應用程式	111
發佈應用程式	112
共用已發佈的應用程式	116
回復至先前發佈的版本	116
匯出應用程式	117
頁面和元件：建置應用程式的使用者介面	118
管理頁面	118
管理元件	121
設定頁面的角色型可見性	122
在應用程式導覽中排序和組織頁面	124
使用應用程式主題變更應用程式中的顏色	125
元件參考	125
自動化和動作：定義應用程式的商業邏輯	166
自動化概念	167
建立、編輯和刪除自動化	168
新增、編輯和刪除自動化動作	169
自動化動作參考	171
實體和資料動作：設定應用程式的資料模型	188
設計資料模型時的最佳實務	188
建立實體	189
設定實體	192
刪除實體	198
受管資料實體	199
頁面和自動化參數	200
頁面參數	200
自動化參數	201
使用 JavaScript 撰寫表達式	206
基本語法	206
插補	207

串連	207
日期和時間	207
程式碼區塊	208
全域變數和函數	208
參考或更新 UI 元件值	208
使用資料表資料	210
存取自動化	211
資料相依性和時間考量	213
範例：訂單詳細資訊和客戶資訊	213
資料相依性和時間最佳實務	213
使用多個使用者建置應用程式	214
邀請建置者編輯應用程式	214
嘗試編輯另一個使用者正在編輯的應用程式	215
更新應用程式的內容安全設定	215
故障診斷和偵錯	218
設定、許可和加入	218
選擇為我建立帳戶執行個體選項時，App Studio 設定失敗	218
設定後無法存取 App Studio	218
不確定登入 App Studio 時要使用的使用者名稱或密碼	219
我在設定 App Studio 時收到系統錯誤	219
我找不到我的 App Studio 執行個體 URL	219
我無法在 App Studio 中修改群組或角色	219
如何從 App Studio 離職	218
疑難排解和偵錯應用程式	220
AI 建置器助理	220
在應用程式工作室	221
預覽應用程式	221
在測試環境中	222
在 CloudWatch 中使用日誌	223
連接器	225
發佈和共用應用程式	228
我在共用對話方塊中看不到新建立的應用程式角色	228
我的應用程式發佈完成時，我未收到電子郵件	228
我應用程式的最終使用者無法存取已發佈的應用程式	228
安全	229
安全考量和緩解措施	229

安全考量	229
安全風險緩解建議	230
資料保護	230
資料加密	231
傳輸中加密	232
金鑰管理	232
網際網路流量隱私權	232
App Studio 和 Identity and Access Management	232
身分型政策	234
資源型政策	234
政策動作	235
政策資源	235
政策條件索引鍵	235
ACL	236
ABAC	236
臨時憑證	236
主體許可	236
服務角色	236
服務連結角色	237
AWS 受管政策	237
服務連結角色	240
身分型政策範例	243
法規遵循驗證	246
恢復能力	246
基礎設施安全性	247
組態與漏洞分析	247
預防跨服務混淆代理人	247
跨區域資料傳輸	248
支援的瀏覽器	249
建置應用程式的支援和建議的瀏覽器	249
應用程式最終使用者的支援和建議的瀏覽器	249
更新瀏覽器設定以在 App Studio 上建置應用程式	249
配額	251
文件歷史紀錄	252
.....	cclix

什麼是 AWS App Studio ？

AWS App Studio 是採用生成式 AI 的服務，使用自然語言來協助您建立企業級應用程式。App Studio 向沒有軟體開發技能的技術專業人員開放應用程式開發，例如 IT 專案經理、資料工程師和企業架構師。透過 App Studio，您可以快速建置安全且完全由管理的應用程式 AWS，而無需操作專業知識。

建置器可以使用 App Studio 來建立和部署應用程式，以現代化內部業務流程。一些使用案例範例包括庫存管理和追蹤、宣告處理和複雜核准，以改善員工生產力和客戶成果。

主題

- [您是第一次使用 App Studio 嗎？](#)

您是第一次使用 App Studio 嗎？

如果您是第一次使用 App Studio，建議您先閱讀以下章節：

- 對於將設定 App Studio、管理使用者和存取，以及使用其他 AWS 或第三方服務設定連接器的管理員角色使用者，請參閱 [AWS App Studio 概念](#)和 [設定並登入 AWS App Studio](#)。
- 如需將建立和開發應用程式的建置器，請參閱 [AWS App Studio 概念](#)和 [AWS App Studio 入門](#)。

AWS App Studio 概念

熟悉重要的 App Studio 概念，以協助加速為您的團隊建立應用程式和自動化程序。這些概念包括用於管理員和建置器的 App Studio 術語。

主題

- [管理員角色](#)
- [應用程式 \(應用程式\)](#)
- [自動化](#)
- [自動化動作](#)
- [Builder 角色](#)
- [元件](#)
- [連接器](#)
- [開發環境](#)
- [實體](#)
- [執行個體](#)
- [頁面](#)
- [觸發條件](#)

管理員角色

Admin 是可以指派給 App Studio 中群組的角色。管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。

只有具有 Admin 角色的使用者才能存取 Admin Hub，其中包含管理角色、資料來源和應用程式的工具。

應用程式 (應用程式)

應用程式 (app) 是一種單一軟體程式，專為最終使用者開發，以完成特定任務。App Studio 中的應用程式包含資產，例如 UI 頁面和元件、自動化，以及使用者可以互動的資料來源。

自動化

自動化是您定義應用程式商業邏輯的方式。自動化的主要元件包括：啟動自動化的觸發、一或多個動作的序列、用來將資料傳遞至自動化的輸入參數，以及輸出。

自動化動作

自動化動作通常稱為動作，是構成自動化之邏輯的個別步驟。每個動作都會執行特定任務，無論是傳送電子郵件、建立資料記錄、叫用 Lambda 函數或呼叫 APIs。動作會從動作程式庫新增至自動化，並可分組為條件式陳述式或迴圈。

Builder 角色

Builder 是可以指派給 App Studio 中群組的角色。建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。

具有 Builder 角色的使用者可以存取 Builder Hub，其中包含有關資源的詳細資訊，例如建置器可存取的應用程式，以及有用的資訊，例如學習資源。

元件

元件是應用程式 UI 中的個別功能項目。元件包含在頁面中，有些元件可以做為其他元件的容器。元件結合 UI 元素與您希望該 UI 元素執行的業務邏輯。例如，一種類型的元件是表單，使用者可以在欄位中輸入資訊，提交後，該資訊會新增為資料庫記錄。

連接器

連接器是 App Studio 與其他 AWS 服務之間的連線，例如 AWS Lambda 和 Amazon Redshift 或第三方服務。建立和設定連接器後，建置器就可以在應用程式中使用連接器及其連接到 App Studio 的資源。

只有具有管理員角色的使用者才能建立、管理或刪除連接器。

開發環境

開發環境是建置應用程式的視覺化工具。此環境包含下列用於建置應用程式的索引標籤：

- 頁面：建置器使用[頁面](#)和[元件](#)設計其應用程式的位置。
- 自動化：建置器使用[自動化](#)設計其應用程式的商業邏輯。
- 資料：建置器使用[實體](#)設計其應用程式的資料模型的位置。

開發環境也包含偵錯主控台，以及 AI 聊天視窗，可在建置時取得情境說明。建置器可以從開發環境預覽其進行中的應用程式。

實體

實體是 App Studio 中的資料表。實體會直接與資料來源中的資料表互動。實體包括描述其中資料的欄位、尋找和傳回資料的查詢，以及將實體欄位連接至資料來源資料欄的映射。

執行個體

執行個體是所有 App Studio 資源的邏輯容器。它代表您、您的公司、團隊或組織，並包含您所有的 App Studio 資源，例如使用者和群組的應用程式、連接器和角色指派。較大的組織或企業通常有多個 App Studio 執行個體，例如：沙盒、測試和生產執行個體。您可以在設定 App Studio 時建立執行個體。

頁面

頁面是[元件](#)的容器，構成 App Studio 中應用程式的 UI。每個頁面都代表應用程式使用者介面 (UI) 的畫面，您的使用者將與其互動。頁面會在應用程式工作室的頁面索引標籤中建立和編輯。

觸發條件

觸發會決定自動化執行的時間和條件。有些觸發範例 `On click` 適用於按鈕和 `On select` 文字輸入。元件類型會決定該元件的可用觸發條件清單。觸發條件會新增至[元件](#)，並在應用程式工作室中設定。

AWS App Studio 的運作方式

使用 AWS App Studio 建置應用程式時，需要了解幾個關鍵概念。本主題涵蓋下列概念或資源的基本概念：

- 使用連接器連接到其他服務，在您的應用程式中使用其資源或 API 呼叫。例如，您可以使用連接器來存放和存取資料，或從應用程式傳送通知。
- 使用實體來設定應用程式的資料模型，將應用程式與外部資料來源連線。
- 使用頁面和元件來建置應用程式的使用者介面 (UI)。
- 使用自動化和動作來實作應用程式的邏輯或行為。
- App Studio 中的應用程式開發生命週期：建置、測試和發佈。

如需 App Studio 概念的詳細資訊，請參閱 [AWS App Studio 概念](#)。

下圖是如何組織 App Studio 及其資源的簡單圖表。



在 App Studio 中的應用程式中，頁面、自動化和實體都會彼此互動。您可以使用連接器將這些資源連接到外部服務，例如資料、儲存或通知供應商。若要成功建置應用程式，請務必了解所有這些概念和資源如何彼此互動。

將您的應用程式連線至其他 服務

使用 App Studio 建置應用程式的最大好處之一是能夠輕鬆整合您的應用程式與其他 服務。在 App Studio 中，您可以使用服務特有的連接器，以及您想要與應用程式搭配使用的資源或 API 呼叫，來連線至其他 服務。

您可以在 App Studio 執行個體層級建立連接器，而不是在個別應用程式中建立連接器。建立連接器後，您可以根據連線的服務和應用程式，在應用程式的各個部分使用連接器。

以下是使用連接器連線至其他 服務之應用程式中的功能範例：

- 幾乎所有應用程式中使用的最常見使用案例，是透過連線至 Amazon Redshift、Amazon DynamoDB 或 Amazon Aurora 等資料服務來存放和存取應用程式中使用 AWS 的資料。

- 允許上傳和檢視影像的應用程式，例如收據，可以使用 Amazon S3 來存放和存取影像檔案。
- 文字摘要器應用程式可以將文字輸入傳送至 Amazon Bedrock，並顯示傳回的摘要。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。建立連接器時，您必須包含您要使用的資源或 API 呼叫的相關適當登入資料和資訊。

設定應用程式的資料模型

您應用程式的資料是支援應用程式的資訊。在 App Studio 中，您可以建立和使用代表存放和使用之不同資料類型的實體。例如，在客戶會議的追蹤應用程式中，您可能有三個實體代表客戶會議、議程和出席者。

實體包含的欄位類型描述要存放的資料，例如整數或字串。雖然您使用實體來定義資料模型，但您必須將實體連線至外部資料儲存服務，例如 Amazon Redshift 或 Amazon DynamoDB 來儲存資料。您可以將實體視為 App Studio 應用程式與外部服務中資料之間的媒介。

您可以使用資料動作，從元件和自動化與應用程式中的資料互動。要使用的兩個最常見的資料動作是 `getAll` 動作和 `getByID` 動作。例如，您的應用程式可以使用 `getAll` 資料動作將資料填入資料表，並使用 `getByID` 動作將特定資料輸入的詳細資訊填入詳細資訊元件。

您也可以將範例資料新增至實體，以更輕鬆地測試應用程式，而無需呼叫外部服務。

建置應用程式的 UI

在 App Studio 中，您可以使用頁面和元件建置應用程式的 UI。頁面是應用程式的個別畫面，也是元件的容器。元件是您應用程式 UI 的建置區塊。元件有許多類型，例如資料表、表單、影像檢視器和按鈕。

下圖顯示應用程式工作室的分頁，您可以在其中新增或設定應用程式中的頁面和元件。下列關鍵區域會反白顯示並編號：

1. 左側分頁面板。這是您管理頁面、應用程式標頭和導覽設定的地方。您可以檢視應用程式的所有頁面和元件。
2. 畫布，顯示目前頁面的元件。您可以選擇畫布中的元件來設定其屬性。

3. 右側元件或屬性面板。如果未選取任何項目，則會顯示元件面板，其中會顯示可新增至頁面的元件清單。如果您選取頁面或元件，則會顯示屬性面板，您可以在其中設定頁面或元件。
4. 底部的錯誤和警告面板。這些面板會在您的應用程式中顯示任何錯誤或警告，這最常見於組態問題。您可以選擇面板來展開它，並查看訊息。



例如，使用者必須輸入資訊的應用程式可能具有下列頁面和元件：

- 輸入頁面，其中包含使用者用來填寫和提交資訊的表單元件。
- 包含資料表元件的清單檢視頁面，其中包含每個輸入的相關資訊。
- 詳細檢視頁面，其中包含詳細資訊元件，其中包含每個輸入的詳細資訊。

元件可以包含靜態資訊或資料，例如具有定義欄位的表單。它們也可以使用自動化來包含動態資訊，例如從 Amazon S3 儲存貯體擷取影像並將其顯示給使用者的影像檢視器。

請務必了解頁面參數的概念。您可以使用頁面參數，將資訊從一個頁面傳送到另一個頁面。頁面參數的使用案例的常見範例是搜尋和篩選，其中從一個頁面的搜尋詞彙會傳送到另一個頁面中要篩選的項目表格或清單。另一個使用案例範例是檢視項目詳細資訊，其中項目識別符會傳送到詳細的檢視器頁面。

實作應用程式的邏輯或行為

您可以將應用程式的邏輯或行為視為應用程式的功能。您可以定義當使用者選擇按鈕、提交資訊、導覽至新頁面或以其他方式互動時會發生的情況。在 App Studio 中，您可以使用自動化和動作來定義應用程式的邏輯。自動化是動作的容器，這是自動化功能的建置區塊。

下圖顯示應用程式工作室的自動化索引標籤，您可以在其中新增或設定自動化及其在應用程式中的動作。下列關鍵區域會反白顯示並編號：

- 左側自動化面板。這是您管理自動化的地方。您可以檢視應用程式的所有自動化和動作。
- 畫布，會顯示目前的自動化。它會顯示設定的自動化參數（本節稍後說明）和動作。您可以選擇畫布中的元件來設定其屬性。
- 右側動作和屬性面板。如果未選取任何項目，則會顯示動作面板。它會顯示可新增至自動化的動作清單。如果您選擇自動化，您可以檢視和設定其屬性，例如自動化的輸入和輸出。如果您選擇動作，您可以檢視和設定動作的屬性。

- 底部的錯誤和警告面板。此面板會顯示應用程式中的任何錯誤或警告（通常來自組態問題）。您可以選擇面板來展開它，並查看訊息。



自動化可以很簡單（例如新增數字並傳回結果），也可以更強大（例如將輸入傳送至另一個服務並傳回結果）。自動化的主要元件如下：

- 觸發，定義自動化執行的時間。例如，當使用者在 UI 中按下按鈕時。
- 自動化輸入，會將資訊傳送至自動化。您可以使用自動化參數定義自動化輸入。例如，如果您想要使用 Amazon Bedrock 將文字摘要傳回給使用者，您可以將文字設定為摘要為自動化參數。
- 動作，這是自動化功能的建置區塊。您可以將每個動作視為自動化中的一個步驟。動作可以呼叫 APIs、叫用自訂 JavaScript、建立資料記錄，以及執行其他函數。您也可以將動作分組為迴圈或條件，以進一步自訂功能。您也可以使用動作叫用其他自動化。
- 自動化輸出，您可以在元件或其他自動化中使用。例如，自動化輸出可以是文字元件中顯示的文字、影像檢視器元件中顯示的影像，或其他自動化的輸入。

應用程式的開發生命週期

應用程式的開發生命週期包括下列階段：建置、測試和發佈。它稱為循環，因為您在建立和迭代應用程式時，可能會在這些階段之間反覆執行。

下圖顯示 App Studio 中應用程式開發生命週期的簡化時間表：



App Studio 提供各種工具來支援應用程式的生命週期。這些工具包括以下三種不同的環境，如上圖所示：

- 預覽環境，您可以在其中預覽您的應用程式，查看它對最終使用者的外觀，並測試特定功能。使用預覽環境在應用程式上快速測試和迭代，而不需要發佈。預覽環境中的應用程式不會與外部服務通訊或傳輸資料。這表示您無法測試在預覽環境中依賴外部服務的互動和功能。
- 測試環境，您可以在其中測試應用程式的連線以及與外部服務的互動。您也可以將發佈至測試環境的版本分享給測試人員群組，藉此執行最終使用者測試。
- 生產環境，您可以在其中對新應用程式執行最終測試，然後再與最終使用者共用。共用應用程式後，發佈至生產環境的應用程式版本即為最終使用者將檢視和使用的版本。

進一步了解

現在您已了解應用程式開發如何在 App Studio 中運作的基本概念，您可以開始建置自己的應用程式，或深入了解概念和資源。

若要開始建置，建議您嘗試其中一個入門教學課程：

- 遵循以[教學課程：使用 AI 產生應用程式](#)了解如何使用 AI 建置器助理來開始建置應用程式。
- 遵循以[教學課程：從空白應用程式開始建置](#)了解如何從頭開始建置應用程式，同時了解基本概念。

若要進一步了解本主題提及的資源或概念，請參閱下列主題：

- [使用連接器將 App Studio 連接至其他 服務](#)
- [實體和資料動作：設定應用程式的資料模型](#)
- [頁面和元件：建置應用程式的使用者介面](#)
- [自動化和動作：定義應用程式的商業邏輯](#)
- [預覽、發佈和共用應用程式](#)

設定並登入 AWS App Studio

根據您的角色，設定 AWS App Studio 會有所不同：

- 第一次將 App Studio 設定為 AWS 或組織管理員：若要第一次將 App Studio 設定為管理員，如果您沒有 AWS 帳戶，請建立 App Studio 執行個體，並使用 IAM Identity Center 群組設定使用者存取權。建立執行個體後，在 App Studio 中具有管理員角色的任何人都可以進一步設定任務，例如，設定連接器以將其他服務（例如資料來源）連接到您的 App Studio 執行個體。如需第一次設定的相關資訊，請參閱 [第一次建立和設定 App Studio 執行個體](#)。
- 作為建置器入門：當您收到加入 App Studio 作為建置器的邀請時，您必須接受邀請，並提供密碼來啟用您的 IAM Identity Center 使用者憑證。之後，您可以登入 App Studio 並開始建置應用程式。如需接受邀請並加入 App Studio 執行個體的詳細資訊，請參閱 [接受加入 App Studio 的邀請](#)。

第一次建立和設定 App Studio 執行個體

註冊 AWS 帳戶

需要 AWS 帳戶才能設定 App Studio。使用 App Studio 只需要一個 AWS 帳戶 - 建置者和管理員不需要 AWS 帳戶即可使用 App Studio，因為存取是由 IAM Identity Center AWS 管理。

建立 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電或簡訊，並在電話鍵盤輸入驗證碼。

當您註冊時 AWS 帳戶，AWS 帳戶根使用者會建立。根使用者有權存取該帳戶中的所有 AWS 服務和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行 [需要根使用者存取權的任務](#)。

建立管理使用者以管理 AWS 資源

第一次建立 AWS 帳戶時，您會從一組預設登入資料開始，完整存取您帳戶中的所有 AWS 資源。此身分稱為 [AWS 帳戶根使用者](#)。若要建立要與 App Studio 搭配使用 AWS 的角色和資源，強烈建議您不要使用 AWS 帳戶根使用者。反之，我們建議您建立並使用管理使用者。

使用下列主題來建立管理使用者，以管理 AWS 角色和資源，以搭配 App Studio 使用。

- 如需單一獨立 AWS 帳戶，請參閱 [《IAM 使用者指南》中的建立您的第一個 IAM 使用者](#)。您可以提供任何使用者名稱，但必須具有 AdministratorAccess 許可政策。
- 如需透過 管理的多個 AWS 帳戶 AWS Organizations，請參閱 AWS IAM Identity Center 《使用者指南》中的 [為 IAM Identity Center 管理使用者設定 AWS 帳戶存取權](#)。

在 中建立 App Studio 執行個體 AWS 管理主控台

若要使用 App Studio，您必須從 中的 App Studio 登陸頁面建立執行個體 AWS 管理主控台。有兩種方法可用於建立 App Studio 執行個體：

1. **輕鬆建立：**使用此簡化方法，您只需設定一個使用者即可存取和使用 App Studio 做為設定的一部分。如果您正在評估組織或團隊的 App Studio，或者您只打算自行使用 App Studio，您應該使用此方法。您可以在設定後將更多使用者或群組新增至 App Studio。請注意，如果您有 IAM Identity Center 的組織執行個體，則無法使用此方法。
2. **標準建立：**使用此方法，您可以新增使用者或群組，並在 App Studio 中指派角色，做為設定的一部分。如果您想要在設定時將多個使用者新增至 App Studio，您應該使用此方法。

Note

您只能在所有 AWS 區域中建立一個 App Studio 執行個體。如果您有現有的執行個體，您必須在建立另一個執行個體之前將其刪除。如需詳細資訊，請參閱 [刪除 App Studio 執行個體](#)。

Easy create

若要在 中輕鬆建立 App Studio AWS 管理主控台 執行個體

1. 在 <https://console.aws.amazon.com/appstudio/> 開啟 App Studio 主控台。
2. 導覽至您要在其中建立 App Studio 執行個體 AWS 的區域。
3. 選擇開始使用。
4. 選擇輕鬆建立，然後選擇下一步。
5. 設定 App Studio 的後續步驟取決於您是否擁有 IAM Identity Center 帳戶執行個體。若要尋找 IAM Identity Center 執行個體的詳細資訊，包括不同類型的以及如何尋找您擁有的類型，請參

閱 [《IAM Identity Center 使用者指南》](#) 中的管理 IAM Identity Center 的組織和帳戶執行個體。

AWS

- 如果您有 IAM Identity Center 的帳戶執行個體：
 - a. 在帳戶許可中，檢閱啟用 App Studio 所需的許可。如果您的帳戶沒有必要的許可，您將無法啟用 App Studio。您必須取得新增至您帳戶的必要許可，或切換到具有這些許可的帳戶。
 - b. 在新增使用者中，搜尋並選取 IAM Identity Center 帳戶執行個體中將存取 App Studio 之使用者的電子郵件地址。此使用者在 App Studio 執行個體中將具有管理員角色。如果您沒有看到想要提供 App Studio 存取權的使用者，您可能需要將他們新增至 IAM Identity Center 執行個體。
- 如果您沒有 IAM Identity Center 的帳戶執行個體：

Note

設定 App Studio 會自動使用您在設定程序期間設定的使用者建立 IAM Identity Center 帳戶執行個體。設定完成後，您可以在 IAM Identity Center 主控台中新增或管理使用者和群組，網址為 <https://console.aws.amazon.com/singlesignon/>。

- a. 在帳戶許可中，檢閱啟用 App Studio 所需的許可。如果您的帳戶沒有必要的許可，您將無法啟用 App Studio。您必須取得新增至您帳戶的必要許可，或切換到具有這些許可的帳戶。
 - b. 在新增使用者中，為存取 App Studio 的使用者提供電子郵件地址、名字、姓氏和使用名稱。此使用者在 App Studio 執行個體中將具有管理員角色。
6. 在服務存取和角色中，檢閱設定 App Studio 為服務提供必要許可時自動建立的服務角色和服務連結角色。選擇檢視許可以查看授予服務角色的確切許可，或選擇檢視政策以查看連接到服務連結角色的許可政策。
 7. 在確認中，選擇陳述式的核取方塊以確認陳述式。
 8. 選擇設定以建立執行個體。

Note


若要在設定後將更多使用者或群組新增至您的 App Studio 執行個體，您必須將其新增至 IAM Identity Center 執行個體。

Standard create

AWS 管理主控台 使用標準方法在 中建立 App Studio 執行個體

1. 在 <https://console.aws.amazon.com/appstudio/> 開啟 App Studio 主控台。
2. 導覽至您要在其中建立 App Studio 執行個體 AWS 的區域。
3. 選擇開始使用。
4. 選擇標準建立，然後選擇下一步。
5. 設定 App Studio 的步驟取決於您是否擁有 IAM Identity Center 執行個體，以及執行個體的類型。若要尋找 IAM Identity Center 執行個體的詳細資訊，包括不同類型的以及如何尋找您擁有的類型，請參閱 [《IAM Identity Center 使用者指南》中的管理 IAM Identity Center 的組織和帳戶執行個體](#)。AWS
 - 如果您有 IAM Identity Center 的組織執行個體：
 - 在使用單一登入設定對 App Studio 的存取中，選取現有的 IAM Identity Center 群組，為他們提供對 App Studio 的存取。App Studio 群組將根據指定的組態建立。新增至管理員群組的群組成員將具有管理員角色，而新增至建置器群組的群組成員將在 App Studio 中具有建置器角色。角色的定義如下：
 - 管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
 - 建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。
 - 如果您有 IAM Identity Center 執行個體的帳戶執行個體：
 - a. 在帳戶許可中，檢閱啟用 App Studio 所需的許可。如果您的帳戶沒有必要的許可，您將無法啟用 App Studio。您必須取得新增至您帳戶的必要許可，或切換到具有這些許可的帳戶。

- b. 在使用單一登入設定對 App Studio 的存取中，在 IAM Identity Center 帳戶中，選擇使用現有的帳戶執行個體
 - c. 在AWS 區域中，選擇 IAM Identity Center 帳戶執行個體所在的 Region。
 - d. 選取現有的 IAM Identity Center 群組，讓他們能夠存取 App Studio。App Studio 群組將根據指定的組態建立。新增至管理員群組的群組成員將具有管理員角色，而新增至建置器群組的群組成員將在 App Studio 中具有建置器角色。角色的定義如下：
 - 管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
 - 建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。
- 如果您沒有 IAM Identity Center 執行個體：

 Note

設定 App Studio 會自動使用您在設定程序期間設定的群組建立 IAM Identity Center 帳戶執行個體。設定完成後，您可以在 IAM Identity Center 主控台中新增或管理使用者和群組，網址為 <https://console.aws.amazon.com/singlesignon/>。

- a. 在帳戶許可中，檢閱啟用 App Studio 所需的許可。如果您的帳戶沒有必要的許可，您將無法啟用 App Studio。您必須取得新增至您帳戶的必要許可，或切換到具有這些許可的帳戶。
- b. 在使用單一登入設定 App Studio 的存取權中，在 IAM Identity Center 帳戶中，選擇為我建立帳戶執行個體。
- c. 在建立使用者和群組並將其新增至 App Studio 中，提供名稱並將使用者新增至管理員群組和建置器群組。新增至管理員群組的使用者將擁有 App Studio 中的管理員角色，而新增至建置器群組的使用者將擁有建置器角色。角色的定義如下：
 - 管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
 - 建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。

⚠ Important

您必須將自己新增為管理員群組的使用者，才能設定 App Studio，並在設定後擁有管理員存取權。

6. 在服務存取和角色中，檢閱設定 App Studio 為服務提供必要許可時自動建立的服務角色和服務連結角色。選擇檢視許可以查看授予服務角色的確切許可，或選擇檢視政策以查看連接到服務連結角色的許可政策。
7. 在確認中，選取陳述式的核取方塊以確認陳述式。
8. 選擇設定以建立執行個體。

接受加入 App Studio 的邀請

App Studio 的存取權由 IAM Identity Center 管理。這表示想要使用 App Studio 的每個使用者都必須在 IAM Identity Center 中設定使用者，並屬於管理員新增至 App Studio 的群組。當管理員邀請您加入 IAM Identity Center 時，您會收到一封電子郵件，要求您接受邀請並啟用您的使用者登入資料。啟用登入資料後，您可以使用這些登入資料登入 App Studio。

接受 IAM Identity Center 存取 App Studio 的邀請

1. 當您收到邀請電子郵件時，請依照步驟提供密碼，並在 IAM Identity Center 中啟用您的使用者登入資料。如需詳細資訊，請參閱[接受加入 IAM Identity Center 的邀請](#)。
2. 啟用使用者登入資料後，請使用它們登入您的 App Studio 執行個體。

AWS App Studio 入門

下列入門教學課程會逐步引導您在 App Studio 中建置第一個應用程式。

- 建議：若要使用生成式 AI 來描述您要建立的應用程式，並自動建立它及其資源，請參閱 [教學課程：使用 AI 產生應用程式](#)。
- 若要從空白應用程式開始建置，請參閱 [教學課程：從空白應用程式開始建置](#)。

教學課程：使用 AI 產生應用程式

AWS App Studio 包含整個服務的生成式 AI 功能，可協助加速應用程式建置。在本教學課程中，您將了解如何使用自然語言描述您的應用程式，以使用 AI 產生應用程式。

使用 AI 產生應用程式是開始建置的好方法，因為許多應用程式的資源都是為您建立的。使用現有資源從產生的應用程式開始建置，通常比從空白應用程式開始建置更容易。

Note

您可以使用 [AWS App Studio \(預覽\)](#) 檢視部落格文章 [使用自然語言建置企業級應用程式](#)，以檢視包含映像的類似演練。部落格文章也包含有關設定和配置管理員相關資源的資訊，但您可以視需要跳到有關建置應用程式的部分。

當 App Studio 使用 AI 產生應用程式時，它會使用根據您描述的應用程式量身打造的下列資源來建立應用程式：

- 頁面和元件：元件是您應用程式使用者介面的建置區塊。它們代表視覺化元素，例如資料表、表單和按鈕。每個元件都有自己的一組屬性，您可以自訂元件以符合您的特定需求。頁面是元件的容器。
- 自動化：您可以使用自動化來定義控制應用程式行為的邏輯和工作流程。例如，您可以使用自動化來建立、更新、讀取或刪除資料表中的資料列，或與 Amazon S3 儲存貯體中的物件互動。您也可以使用它們來處理資料驗證、通知等任務，或與其他系統的整合。
- 實體：資料是為您的應用程式提供支援的資訊。產生的應用程式會建立類似於資料表的實體。實體代表您需要存放和使用的不同資料類型，例如客戶、產品或訂單。您可以使用 App Studio 連接器，將這些資料模型連接到各種資料來源，包括 AWS 服務和外部 APIs。

內容

- [先決條件](#)
- [步驟 1：建立應用程式](#)
- [步驟 2：探索您的新應用程式](#)
 - [探索頁面和元件](#)
 - [探索自動化和動作](#)
 - [使用實體探索資料](#)
- [步驟 3：預覽您的應用程式](#)
- [後續步驟](#)

先決條件

開始之前，請先檢閱並完成下列先決條件：

- 存取 AWS App Studio。如需詳細資訊，請參閱[設定並登入 AWS App Studio](#)。
- 選用：檢閱[AWS App Studio 概念](#)以熟悉重要的 App Studio 概念。

步驟 1：建立應用程式

產生應用程式的第一步是將您要建立的應用程式描述給 App Studio 中的 AI 助理。您可以檢閱要產生的應用程式，並在產生應用程式之前視需要反覆執行。

使用 AI 產生您的應用程式

1. 登入 App Studio。
2. 在左側導覽中，選擇建置器中樞，然後選擇 + 建立應用程式。
3. 選擇使用 AI 產生應用程式。
4. 在應用程式名稱欄位中，為您的應用程式提供名稱。
5. 在選取資料來源對話方塊中，選擇略過。
6. 您可以在文字方塊中描述應用程式，或在範例提示中選擇自訂，以開始定義要產生的應用程式。在您描述應用程式後，App Studio 會產生應用程式需求和詳細資訊供您檢閱。這包括使用案例、使用者流程和資料模型。
7. 視需要使用文字方塊與您的應用程式進行反覆運算，直到您對需求和詳細資訊感到滿意為止。
8. 當您準備好產生應用程式並開始建置時，請選擇產生應用程式。
9. 或者，您可以檢視簡短的视频，其中詳細說明如何導覽您的新應用程式。

10. 選擇編輯應用程式以輸入應用程式的開發環境。

步驟 2：探索您的新應用程式

在開發環境中，您會找到下列資源：

- 您用來檢視或編輯應用程式的畫布。畫布會根據選取的資源而變更。
- 畫布頂端的導覽標籤。這些索引標籤會在下列清單中說明：
 - 頁面：您可以在其中使用頁面和元件來設計應用程式的 UI。
 - 自動化：您可以在自動化中使用動作來定義應用程式的商業邏輯。
 - 資料：您在其中定義實體、其欄位、範例資料和資料動作，以定義應用程式的資料模型。
 - 應用程式設定：您可以在其中定義應用程式設定，包括應用程式角色，用於為最終使用者定義以角色為基礎的頁面可見性。
- 左側導覽功能表，其中包含您檢視之標籤的資源。
- 右側選單，列出頁面和自動化索引標籤中所選資源的資源和屬性。
- 偵錯主控台，會在建置器底部顯示警告和錯誤。您產生的應用程式中可能存在錯誤。這可能是因為自動化需要設定的連接器來執行動作，例如使用 Amazon Simple Email Service 傳送電子郵件。
- 詢問 AI 聊天視窗，以取得 AI 建置器助理的情境協助。

讓我們進一步了解頁面、自動化和資料索引標籤。

探索頁面和元件

頁面索引標籤會顯示為您產生的頁面及其元件。

每個頁面都代表應用程式使用者介面 (UI) 的畫面，您的使用者將與其互動。在這些頁面上，您可以找到各種元件（例如資料表、表單和按鈕）來建立所需的配置和功能。

使用左側導覽選單花一些時間檢視頁面及其元件。當您選取頁面或元件時，您可以在右側功能表中選擇屬性。

探索自動化和動作

自動化索引標籤會顯示為您產生的自動化及其動作。

自動化會定義應用程式的商業邏輯，例如建立、檢視、更新或刪除資料項目、傳送電子郵件，甚至叫用 APIs 或 Lambda 函數。

使用左側導覽選單花一些時間檢視自動化。當您選擇自動化時，您可以在右側的屬性功能表上檢視其屬性。自動化包含下列資源：

- 自動化是由個別動作組成，這些動作是您應用程式商業邏輯的建置區塊。您可以在左側導覽功能表或所選自動化的畫布中檢視自動化的動作。選取動作時，您可以在右側的屬性功能表上檢視其屬性。
- 自動化參數是將資料傳遞至自動化的方式。參數做為預留位置，在執行自動化時以實際值取代。這可讓您每次使用相同的自動化搭配不同的輸入。
- 自動化輸出是您設定自動化結果的地方。根據預設，自動化沒有輸出，因此若要使用自動化的結果產生元件或其他自動化，您必須在此處定義它們。

如需詳細資訊，請參閱[自動化概念](#)。

使用實體探索資料

資料索引標籤會顯示為您產生的實體。

實體代表保存應用程式資料的資料表，類似於資料庫中的資料表。它們不會將應用程式的使用者介面 (UI) 和自動化直接連接到資料來源，而是先連接到實體。實體充當實際資料來源與 App Studio 應用程式之間的媒介。這提供單一位置來管理和存取您的資料。

花一些時間檢視從左側導覽功能表中選擇產生的實體。您可以檢閱下列詳細資訊：

- 組態索引標籤會顯示實體名稱及其欄位，代表實體的資料欄。
- 資料動作索引標籤會顯示使用實體產生的資料動作。元件和自動化可以使用資料動作從您的實體擷取資料。
- 範例資料索引標籤顯示範例資料，您可以用來在開發環境中測試應用程式（不會與外部服務通訊）。如需環境的詳細資訊，請參閱[應用程式環境](#)。
- Connection 索引標籤會顯示實體所連線之外部資料來源的相關資訊。App Studio 提供使用 DynamoDB 資料表的受管資料儲存解決方案。如需詳細資訊，請參閱[AWS App Studio 中的受管資料實體](#)。

步驟 3：預覽您的應用程式

您可以在 App Studio 中預覽應用程式，以查看使用者的外觀。您也可以使用它並檢查偵錯面板中的日誌來測試其功能。

應用程式預覽環境不支援顯示即時資料或與具有連接器之外部資源的連線，例如資料來源。反之，您可以使用範例資料和模擬輸出來測試功能。

預覽您的應用程式以進行測試

1. 在應用程式建置器的右上角，選擇預覽。
2. 與您應用程式的頁面互動。

後續步驟

現在您已建立第一個應用程式，以下是一些後續步驟：

- 如需包含映像的另一個入門演練，請參閱部落格文章[使用 AWS App Studio 建置自然語言的企業級應用程式（預覽）](#)。
- 應用程式使用連接器來傳送和接收資料，或與外部服務（AWS 服務和第三方服務）通訊。必須進一步了解連接器，以及如何設定連接器來建置應用程式。請注意，您必須擁有管理員角色才能管理連接器。如需詳細資訊，請參閱[使用連接器將 App Studio 連接至其他服務](#)。
- 若要進一步了解預覽、發佈和最終將應用程式分享給最終使用者，請參閱[預覽、發佈和共用應用程式](#)。
- 繼續探索和更新您為一些實作體驗產生的應用程式。
- 若要進一步了解如何建置應用程式，請參閱[建置器文件](#)。具體而言，下列主題可能有助於探索：
 - [自動化動作參考](#)
 - [元件參考](#)
 - [與 Amazon Simple Storage Service 與元件和自動化互動](#)
 - [安全考量和緩解措施](#)

教學課程：從空白應用程式開始建置

在本教學課程中，您將使用 AWS App Studio 建置內部客戶會議請求應用程式。您將了解如何在 App Studio 中建置應用程式，同時專注於實際使用案例和實作範例。此外，您將了解如何定義資料結構、UI 設計和應用程式部署。

Note

本教學課程詳細說明如何從頭開始建置應用程式，從空白應用程式開始。一般而言，透過提供您要建立的應用程式描述，使用 AI 協助產生應用程式及其資源會更快、更輕鬆。如需詳細資訊，請參閱[教學課程：使用 AI 產生應用程式](#)。

了解如何使用 App Studio 建置應用程式的關鍵是了解以下四個核心概念及其如何共同運作：元件、自動化、資料和連接器。

- **元件**：元件是您應用程式使用者介面的建置區塊。它們代表視覺化元素，例如資料表、表單和按鈕。每個元件都有自己的一組屬性，您可以自訂這些屬性以符合您的特定需求。
- **自動化**：透過自動化，您可以定義控制應用程式行為的邏輯和工作流程。您可以使用自動化來建立、更新、讀取或刪除資料表中的資料列，或與 Amazon S3 儲存貯體中的物件互動。您也可以使用它們來處理資料驗證、通知等任務，或與其他系統的整合。
- **資料**：資料是為您的應用程式提供支援的資訊。在 App Studio 中，您可以定義稱為實體的資料模型。實體代表您需要存放和處理的不同資料類型，例如客戶會議請求、議程或出席者。您可以使用 App Studio 連接器，將這些資料模型連接到各種資料來源，包括 AWS 服務和外部 APIs。
- **連接器**：App Studio 提供具有各種資料來源的連線，包括 Aurora、DynamoDB 和 Amazon Redshift 等 AWS 服務。資料來源也包括第三方服務，例如 Salesforce，或使用 OpenAPI 或一般 API 連接器的其他許多服務。您可以使用 App Studio 連接器，輕鬆地將來自這些企業級服務和外部應用程式的資料和功能整合到您的應用程式中。

隨著教學課程的進行，您將探索元件、資料和自動化的關鍵概念如何結合起來，以建置您的內部客戶會議請求應用程式。

以下是說明您將在本教學課程中做什麼的高階步驟：

1. **從資料開始**：許多應用程式從資料模型開始，因此本教學課程也從資料開始。若要建置客戶會議請求應用程式，請先建立 MeetingRequests 實體。此實體代表儲存所有相關會議請求資訊的資料結構，例如客戶名稱、會議日期、議程和出席者。此資料模型是應用程式的基礎，並支援您將建置的各種元件和自動化。
2. **建立您的使用者介面 (UI)**：設定資料模型後，教學課程會引導您建置使用者介面 (UI)。在 App Studio 中，您可以透過新增頁面和新增元件來建置 UI。您可以將資料表、詳細資訊檢視和行事曆等元件新增至會議請求儀表板頁面。這些元件旨在顯示存放在 MeetingRequests 實體中的資料並與之互動。這可讓您的使用者檢視、管理和排程客戶會議。您也將建立會議請求建立頁面。此頁面包含用於收集資料的表單元件，以及用於提交資料的按鈕元件。
3. **使用自動化新增商業邏輯**：若要增強應用程式的功能，您將設定一些元件以啟用使用者互動。一些範例正在導覽至頁面或在 MeetingRequests 實體中建立新的會議請求記錄。
4. **增強驗證和表達式**：為確保資料的完整性和準確性，請將驗證規則新增至表單元件。這將有助於確保使用者在建立新的會議請求記錄時提供完整且有效的資訊。此外，您將使用表達式來參考和操作應用程式中的資料，以便在整個使用者介面中顯示動態和內容資訊。

5. 預覽和測試：在部署應用程式之前，您將有機會預覽和徹底測試應用程式。這可讓您驗證元件、資料和自動化是否都能順暢地協同運作。這可為您的使用者提供流暢且直覺式的體驗。
6. 發佈應用程式：最後，您將部署完成的內部客戶會議請求應用程式，並讓使用者存取。透過 App Studio 中低程式碼方法的強大功能，您將建置符合組織特定需求的自訂應用程式，而不需要廣泛的程式設計專業知識。

內容

- [先決條件](#)
- [步驟 1：建立 應用程式](#)
- [步驟 2：建立實體以定義應用程式的資料](#)
 - [建立受管實體](#)
 - [將欄位新增至您的實體](#)
- [步驟 3：設計使用者介面 \(UI\) 和邏輯](#)
 - [新增會議請求儀表板頁面](#)
 - [新增會議請求建立頁面](#)
- [步驟 4：預覽應用程式](#)
- [步驟 5：將應用程式發佈至測試環境](#)
- [後續步驟](#)

先決條件

開始之前，請先檢閱並完成下列先決條件：

- 存取 AWS App Studio。如需詳細資訊，請參閱[設定並登入 AWS App Studio](#)。
- 選用：檢閱[AWS App Studio 概念](#)以熟悉重要的 App Studio 概念。
- 選用：了解基本 Web 開發概念，例如 JavaScript 語法。
- 選用：熟悉 AWS 服務。

步驟 1：建立 應用程式

1. 登入 App Studio。
2. 在左側導覽中，選擇建置器中樞，然後選擇 + 建立應用程式。

3. 選擇從頭開始。
4. 在應用程式名稱欄位中，提供應用程式的名稱，例如 **Customer Meeting Requests**。
5. 如果系統要求您選取資料來源或連接器，請選擇略過以進行本教學課程。
6. 選擇下一步繼續進行。
7. (選用)：觀看影片教學課程，以取得在 App Studio 中建置應用程式的快速概觀。
8. 選擇編輯應用程式，讓您進入 App Studio 應用程式建置器。

步驟 2：建立實體以定義應用程式的資料

實體代表保存應用程式資料的資料表，類似於資料庫中的資料表。它們不會直接連接到資料來源的應用程式使用者介面 (UI) 和自動化，而是先連接到實體。實體做為實際資料來源和 App Studio 應用程式之間的媒介，並提供單一位置來管理和存取您的資料。

建立實體有四種方式。在本教學課程中，您將使用 App Studio 受管實體。

建立受管實體

建立受管實體也會建立 App Studio 管理的對應 DynamoDB 資料表。在 App Studio 應用程式中對實體進行變更時，DynamoDB 資料表會自動更新。使用此選項，您不需要手動建立、管理或連線至第三方資料來源，或指定從實體欄位到資料表欄的映射。

建立實體時，您必須定義主索引鍵欄位。主索引鍵可做為實體中每個記錄或資料列的唯一識別符。這可確保可以輕鬆識別和擷取每個記錄，而無需模稜兩可。主索引鍵包含下列屬性：

- 主索引鍵名稱：實體主索引鍵欄位的名稱。
- 主索引鍵資料類型：主索引鍵欄位的類型。在 App Studio 中，支援的主要金鑰類型為文字字串和數字浮動。文字主索引鍵 (例如 *meetingName*) 會有一種字串類型，數值主索引鍵 (例如 *meetingId*) 會有一種 Float 類型。

主索引鍵是實體的重要元件，因為它會強制執行資料完整性、防止資料重複，並啟用有效的資料擷取和查詢。

建立受管實體

1. 從頂端列功能表中選擇資料。
2. 選擇 + 建立實體。

3. 選擇建立 App Studio 受管實體。
4. 在實體名稱欄位中，提供實體的名稱。針對本教學，輸入 **MeetingRequests**。
5. 在主索引鍵欄位中，輸入要提供給實體中主索引鍵資料欄的主索引鍵名稱標籤。針對本教學，輸入 **requestID**。
6. 針對主索引鍵資料類型，選擇浮動。
7. 選擇 Create entity (建立實體)。

將欄位新增至您的實體

對於每個欄位，您將指定顯示名稱，這是應用程式使用者可見的標籤。顯示名稱可以包含空格和特殊字元，但在實體中必須是唯一的。顯示名稱可做為易於使用的欄位標籤，並協助使用者輕鬆識別和了解其用途。

接著，您將提供系統名稱，這是應用程式在內部用來參考欄位的唯一識別符。系統名稱應該簡潔，沒有空格或特殊字元。系統名稱允許應用程式變更欄位的資料。它可做為應用程式內欄位的唯一參考點。

最後，您將選取最能代表您要在欄位中存放的資料類型，例如字串（文字）、布林值 (true/false)、日期、小數、浮點數、整數或 DateTime。定義適當的資料類型可確保資料完整性，並啟用欄位值的適當處理和處理。例如，如果您要在會議請求中存放客戶名稱，您可以選取String資料類型來容納文字值。

將欄位新增至您的MeetingRequests實體

- 選擇 + 新增欄位以新增下列四個欄位：
 - a. 使用下列資訊新增代表客戶名稱的欄位：
 - 顯示名稱：**Customer name**
 - 系統名稱：**customerName**
 - 資料類型：**String**
 - b. 使用下列資訊新增代表會議日期的欄位：
 - 顯示名稱：**Meeting date**
 - 系統名稱：**meetingDate**
 - 資料類型：**DateTime**
 - c. 使用下列資訊新增代表會議議程的欄位：

- 顯示名稱：**Agenda**
 - 系統名稱：**agenda**
 - 資料類型：**String**
- d. 新增欄位，以使用下列資訊代表會議出席者：
- 顯示名稱：**Attendees**
 - 系統名稱：**attendees**
 - 資料類型：**String**

您可以將範例資料新增至實體，以便在發佈應用程式之前用來測試和預覽應用程式。透過新增最多 500 列模擬資料，您可以模擬真實案例，並檢查您的應用程式如何處理和顯示各種類型的資料，而無需依賴實際資料或連接到外部服務。這可協助您在開發程序初期識別和解決任何問題或不一致。這可確保您的應用程式在處理實際資料時如預期般運作。

將範例資料新增至您的實體

1. 選擇橫幅中的範例資料索引標籤。
2. 選擇產生更多範例資料。
3. 選擇儲存。

或者，選擇橫幅中的連線，以檢閱連接器的詳細資訊，以及為您建立的 DynamoDB 資料表。

步驟 3：設計使用者介面 (UI) 和邏輯

新增會議請求儀表板頁面

在 App Studio 中，每個頁面代表使用者將互動的應用程式使用者介面 (UI) 畫面。在這些頁面中，您可以新增各種元件，例如資料表、表單和按鈕，以建立所需的配置和功能。

新建立的應用程式隨附預設頁面，因此您將重新命名該應用程式，而不是新增新應用程式，以用作簡單的會議請求儀表板頁面。

重新命名預設頁面

1. 在頂端列導覽選單中，選擇頁面。
2. 在左側面板中，按兩下 Page1，將其重新命名為 **MeetingRequestsDashboard**，然後按 Enter 鍵。

現在，將資料表元件新增至用來顯示會議請求的頁面。

將資料表元件新增至會議請求儀表板頁面

1. 在右側元件面板中，找到資料表元件並將其拖曳到畫布上。
2. 選擇畫布中的資料表以進行選取。
3. 在右側屬性面板中，更新下列設定：
 - a. 選擇鉛筆圖示，將資料表重新命名為 **meetingRequestsTable**。
 - b. 在來源下拉式清單中，選擇實體。
 - c. 在資料動作下拉式清單中，選擇您建立的實體 (**MeetingRequests**)，然後選擇 + 新增資料動作。
4. 如果出現提示，請選擇 `getAll`。

Note

`getAll` 資料動作是一種特定類型的資料動作，可從指定的實體擷取所有記錄（資料列）。例如，當您將 `getAll` 資料動作與資料表元件建立關聯時，資料表會自動填入來自連線實體的所有資料，並將每個記錄顯示為資料表中的資料列。

新增會議請求建立頁面

接下來，建立一個頁面，其中包含最終使用者將用於建立會議請求的表單。您也可以新增在 `MeetingRequests` 實體中建立記錄的提交按鈕，然後將最終使用者導覽回 `MeetingRequestsDashboard` 頁面。

新增會議請求建立頁面

1. 在頂端橫幅中，選擇頁面。
2. 在左側面板中，選擇 + 新增。
3. 在右側屬性面板中，選取鉛筆圖示並將頁面重新命名為 **CreateMeetingRequest**。

現在已新增頁面，您會將表單新增至最終使用者用來輸入資訊以在 `MeetingRequests` 實體中建立會議請求的頁面。App Studio 提供從現有實體產生表單的方法，會根據實體的欄位自動填入表單欄位，也會產生提交按鈕，以在實體中使用表單輸入建立記錄。

在會議請求建立頁面上從實體自動產生表單

1. 在元件功能表右側，尋找表單元件並將其拖曳到畫布上。
2. 選取產生表單。
3. 從下拉式清單中選取MeetingRequests實體。
4. 選擇 Generate (產生)。
5. 選擇畫布上的提交按鈕以進行選取。
6. 在右側屬性面板的觸發區段中，選擇 + 新增。
7. 選擇導覽。
8. 在右側屬性面板中，將動作名稱變更為描述性名稱，例如 **Navigate to MeetingRequestsDashboard**。
9. 將導覽類型變更為頁面。在導覽至下拉式清單中，選擇 **MeetingRequestsDashboard**。

現在我們有了會議請求建立頁面和表單，我們希望從MeetingRequestsDashboard頁面輕鬆導覽至此頁面，以便檢閱儀表板的最終使用者可以輕鬆建立會議請求。使用下列程序，在導覽至MeetingRequestsDashboard頁面的CreateMeetingRequest頁面上建立按鈕。

新增按鈕以從 導覽MeetingRequestsDashboard至 CreateMeetingRequest

1. 在頂端橫幅中，選擇頁面。
2. 選擇MeetingRequestsDashboard頁面。
3. 在右側元件面板中，尋找按鈕元件，將其拖曳到畫布上，並將其放在資料表上方。
4. 選擇新增的按鈕來選取它。
5. 在右側屬性面板中，更新下列設定：
 - a. 選取鉛筆圖示，將按鈕重新命名為 **createMeetingRequestButton**。
 - b. 按鈕標籤：**Create Meeting Request**。這是最終使用者將看到的名稱。
 - c. 在圖示下拉式清單中，選取 + Plus。
 - d. 建立將最終使用者導覽至MeetingRequestsDashboard頁面的觸發：
 1. 在觸發區段中，選擇 + 新增。
 2. 在動作類型中，選取導覽。
 3. 選擇您剛建立的觸發條件以進行設定。
 4. 在動作名稱中，提供描述性名稱，例如 **NavigateToCreateMeetingRequest**。

5. 在導覽類型下拉式清單中，選取頁面。
6. 在導覽至下拉式清單中，選取CreateMeetingRequest頁面。

步驟 4：預覽應用程式

您可以在 App Studio 中預覽應用程式，以查看使用者將如何顯示應用程式。此外，您可以使用它並檢查偵錯面板中的日誌來測試其功能。

應用程式預覽環境不支援顯示即時資料。它也不支援與具有連接器的外部資源連線，例如資料來源。反之，您可以使用範例資料和模擬輸出來測試功能。

預覽您的應用程式以進行測試

1. 在應用程式建置器的右上角，選擇預覽。
2. 與MeetingRequestsDashboard頁面互動，並測試資料表、表單和按鈕。

步驟 5：將應用程式發佈至測試環境

現在您已完成建立、設定和測試應用程式，是時候將其發佈至測試環境，以執行最終測試，然後與使用者共用。

將您的應用程式發佈至測試環境

1. 在應用程式建置器的右上角，選擇發佈。
2. 新增測試環境的版本描述。
3. 檢閱並選取有關 SLA 的核取方塊。
4. 選擇 開始使用。發佈最多可能需要 15 分鐘。
5. (選用) 當您準備好時，您可以選擇共用並依照提示給予其他人存取權。

Note

若要共用應用程式，管理員必須已建立最終使用者群組。

測試後，再次選擇發佈，將應用程式提升為生產環境。如需不同應用程式環境的詳細資訊，請參閱 [應用程式環境](#)。

後續步驟

現在您已建立第一個應用程式，以下是一些後續步驟：

1. 繼續建置教學課程應用程式。現在您已設定資料、某些頁面和自動化，您可以新增其他頁面並新增元件，以進一步了解建置應用程式。
2. 若要進一步了解如何建置應用程式，請參閱 [建置器文件](#)。具體而言，下列主題可能有助於探索：
 - [自動化動作參考](#)
 - [元件參考](#)
 - [與 Amazon Simple Storage Service 與元件和自動化互動](#)
 - [安全考量和緩解措施](#)

此外，下列主題包含教學課程中討論之概念的詳細資訊：

- [預覽、發佈和共用應用程式](#)
- [在 App Studio 應用程式中建立實體](#)

管理員文件

下列主題包含的資訊可協助在 App Studio 中管理第三方服務連線和存取、使用者和角色的使用者。

主題

- [在 App Studio 中管理存取和角色](#)
- [使用連接器將 App Studio 連接至其他 服務](#)
- [刪除 App Studio 執行個體](#)

在 App Studio 中管理存取和角色

App Studio 中管理員的責任之一是管理存取、角色和許可。下列主題包含 App Studio 中角色的相關資訊，以及如何新增使用者、移除使用者或變更其角色。

使用 IAM Identity Center 群組管理對 AWS App Studio 的存取。若要將使用者新增至您的 App Studio 執行個體，您必須：

- 將它們新增至新增至 App Studio 的現有 IAM Identity Center 群組。
- 將它們新增至未新增至 App Studio 的新或現有 IAM Identity Center 群組，然後將其新增至 App Studio。

由於角色會套用至群組，IAM Identity Center 群組應該代表您要指派給群組成員的存取權限（或角色）。如需 IAM Identity Center 的詳細資訊，包括管理使用者和群組的相關資訊，請參閱 [IAM Identity Center 使用者指南](#)。

角色和許可

App Studio 中有三個角色。下列清單包含每個角色及其描述。

- **管理員**：管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
- **建置器**：建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。
- **應用程式使用者**：應用程式使用者可以存取和使用已發佈的應用程式，但無法存取您的 App Studio 執行個體來建置應用程式或管理資源。

在 App Studio 中，角色會指派給群組，因此新增的 IAM Identity Center 群組的每個成員都會獲指派指派給群組的角色。

檢視群組

執行下列步驟，以檢視新增至 App Studio 執行個體的群組。

Note

您必須是管理員，才能在 App Studio 執行個體中檢視群組。

檢視新增至 App Studio 執行個體的群組

- 在導覽窗格中，在管理區段中選擇角色。系統會將您導向至顯示現有群組清單的頁面，以及每個群組指派的角色。

如需管理群組的資訊，請參閱 [新增使用者或群組](#)、[變更群組的角色](#) 或 [從 App Studio 移除使用者或群組](#)。

新增使用者或群組

若要將使用者新增至 App Studio，您必須將使用者新增至 IAM Identity Center 群組，並將該群組新增至 App Studio。執行下列步驟，透過新增 IAM Identity Center 群組和指派角色，將使用者新增至 App Studio。

Note

您必須是管理員，才能將使用者新增至您的 App Studio 執行個體。

將使用者或群組新增至您的 App Studio 執行個體

- 若要將使用者新增至您的 App Studio 執行個體，您必須將使用者新增至已新增至 App Studio 的現有 IAM Identity Center 群組，或建立新的 IAM Identity Center 群組、將新使用者新增至其中，並將新群組新增至 App Studio。

如需有關管理 IAM Identity Center 使用者和群組的資訊，請參閱 [AWS IAM Identity Center 《使用者指南》](#) 中的 [在 IAM Identity Center 中管理身分](#)。

2. 如果您已將使用者新增至已新增至 App Studio 的現有 IAM Identity Center 群組，則新使用者在完成其 IAM Identity Center 許可的設定後，可以使用指定的許可存取 App Studio。如果您建立了新的 IAM Identity Center 群組，請執行下列步驟，將群組新增至 App Studio，並為群組的成員指定角色。
3. 在導覽窗格中，在管理區段中選擇角色。
4. 在角色頁面上，選擇 + 新增群組。這會開啟新增群組對話方塊，供您輸入群組的相關資訊。
5. 在新增群組對話方塊中，輸入下列資訊：
 - 在下拉式清單中選擇現有的 IAM Identity Center 群組。
 - 選取群組的角色。
 - 管理員：管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
 - 建置器：建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。
 - 應用程式使用者：應用程式使用者可以存取和使用已發佈的應用程式，但無法存取您的 App Studio 執行個體來建置應用程式或管理資源。
6. 選擇指派，將群組新增至 App Studio，並提供其成員已設定的角色。

變更群組的角色

請依照下列步驟，在 App Studio 中變更指派給群組的角色。變更群組的角色會變更該群組中每個成員的角色。

Note

您必須是管理員，才能在 App Studio 中變更群組的角色。

變更群組的角色

1. 在導覽窗格中，在管理區段中選擇角色。系統會將您導向至顯示現有群組清單的頁面，以及每個群組指派的角色。
2. 選擇省略號圖示 (...)，然後選擇變更角色。
3. 在變更角色對話方塊中，選取群組的新角色：

- **管理員**：管理員可以在 App Studio 中管理使用者和群組、新增和管理連接器，以及管理建置器建立的應用程式。此外，具有 Admin 角色的使用者具有 Builder 角色隨附的所有許可。
- **建置器**：建置器可以建立和建置應用程式。建置器無法管理使用者或群組、新增或編輯連接器執行個體，或管理其他建置器的應用程式。
- **應用程式使用者**：應用程式使用者可以存取和使用已發佈的應用程式，但無法存取您的 App Studio 執行個體來建置應用程式或管理資源。

4. 選擇變更 變更群組的角色。

從 App Studio 移除使用者或群組

您無法從 App Studio 移除 IAM Identity Center 群組。執行下列指示會改為將群組的角色降級為 App User。群組的成員仍然可以存取已發佈的 App Studio 應用程式。

若要移除 App Studio 及其應用程式的所有存取權，您必須在 AWS IAM Identity Center 主控台中刪除 IAM Identity Center 群組或使用者。如需有關管理 IAM Identity Center 使用者和群組的資訊，請參閱 AWS IAM Identity Center [《使用者指南》](#) 中的 [在 IAM Identity Center 中管理身分](#)。

Note

您必須是管理員，才能在 App Studio 中降級群組的存取權。

移除群組

1. 在導覽窗格中，在管理區段中選擇角色。系統會將您導向至顯示現有群組清單的頁面，以及每個群組指派的角色。
2. 選擇省略號圖示 (...)，然後選擇撤銷角色。
3. 在撤銷角色對話方塊中，選擇撤銷，將群組的角色降級為應用程式使用者。

使用連接器將 App Studio 連接至其他 服務

連接器是 App Studio 與其他 AWS 服務之間的連線，例如 AWS Lambda 和 Amazon Redshift 或第三方服務。建立和設定連接器後，建置器就可以在應用程式中使用連接器及其連接到 App Studio 的資源。

只有具有管理員角色的使用者才能建立、管理或刪除連接器。

主題

- [連線至 AWS 服務](#)
- [連線至第三方服務](#)
- [檢視、編輯和刪除連接器](#)

連線至 AWS 服務

主題

- [連線至 Amazon Redshift](#)
- [連線至 Amazon DynamoDB](#)
- [連線至 AWS Lambda](#)
- [連線至 Amazon Simple Storage Service \(Amazon S3\)](#)
- [連線至 Amazon Aurora](#)
- [連線至 Amazon Bedrock](#)
- [連線至 Amazon Simple Email Service](#)
- [AWS 使用其他服務連接器連線至 AWS 服務](#)
- [搭配 CMKs 使用加密的資料來源](#)

連線至 Amazon Redshift

若要將 App Studio 與 Amazon Redshift 連線，讓建置器能夠在應用程式中存取和使用 Amazon Redshift 資源，您必須執行下列步驟：

1. [步驟 1：建立和設定 Amazon Redshift 資源](#)
2. [步驟 2：使用適當的 Amazon Redshift 許可建立 IAM 政策和角色](#)
3. [步驟 3：建立 Amazon Redshift 連接器](#)

步驟 1：建立和設定 Amazon Redshift 資源

使用下列程序來建立和設定要與 App Studio 搭配使用的 Amazon Redshift 資源。

設定 Amazon Redshift 以搭配 App Studio 使用

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/redshiftv2/>：// 開啟 Amazon Redshift 主控台。

建議使用在 中建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。

2. 建立 Redshift Serverless 資料倉儲或佈建叢集。如需詳細資訊，請參閱《Amazon Redshift 使用者指南》中的 [使用 Redshift Serverless 建立資料倉儲](#) 或 [建立叢集](#)。
3. 佈建完成後，請選擇查詢資料以開啟查詢編輯器。連線至您的資料庫。
4. 變更下列設定：
 1. 將隔離工作階段切換設定為 OFF。這是必要的，以便您可以查看其他使用者所做的資料變更，例如從執行中的 App Studio 應用程式。
 2. 選擇「齒輪」圖示。選擇 Account settings (帳戶設定)。增加與的並行連線上限10。這是可連線至 Amazon Redshift 資料庫之查詢編輯器工作階段的數量限制。它不適用於其他用戶端，例如 App Studio 應用程式。
5. 在public結構描述下建立資料表。這些資料表INSERT中的任何初始資料。
6. 在查詢編輯器中執行下列命令：

下列命令會建立資料庫使用者，並將其與 App Studio 所使用的名為 *AppBuilderDataAccessRole* 的 IAM 角色連線。您將在後續步驟中建立 IAM 角色，而此處的名稱必須符合提供給該角色的名稱。

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

下列命令會將所有資料表的所有許可授予 App Studio。

Note

針對最佳安全實務，您應該將此處的許可範圍縮小為適當資料表所需的最低許可。如需 GRANT 命令的詳細資訊，請參閱《Amazon Redshift 資料庫開發人員指南》中的 [GRANT](#)。

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

步驟 2：使用適當的 Amazon Redshift 許可建立 IAM 政策和角色

若要搭配 App Studio 使用 Amazon Redshift 資源，管理員必須建立 IAM 政策和角色，以授予 App Studio 存取資源的許可。IAM 政策控制建置器可以使用的資料範圍，以及可以針對該資料呼叫哪些操作，例如建立、讀取、更新或刪除。接著，IAM 政策會連接至 App Studio 所使用的 IAM 角色。

我們建議每個服務和政策至少建立一個 IAM 角色。例如，如果建置器在 Amazon Redshift 中建立兩個由不同資料表支援的應用程式，則管理員應該建立兩個 IAM 政策和角色，一個用於 Amazon Redshift 中的每個資料表。

步驟 2a：使用適當的 Amazon Redshift 許可建立 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

使用適當的 Amazon Redshift 許可建立 IAM 政策

1. 使用具有建立 [IAM 政策許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 在政策編輯器中，選擇 JSON 選項。
5. 在 JSON 政策文件中輸入或貼上。下列索引標籤包含佈建和無伺服器 Amazon Redshift 的範例政策。

Note

下列政策適用於使用萬用字元 (*) 的所有 Amazon Redshift 資源*。為了實現最佳安全實務，您應該將萬用字元取代為您要搭配 App Studio 使用之資源的 Amazon Resource Name (ARN)。

Provisioned

JSON

```
{
  "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

JSON

```

{
  "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. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **RedshiftServerlessForAppStudio** 或 **RedshiftProvisionedForAppStudio**，以及描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon Redshift 資源

現在，建立使用您先前建立之政策的 IAM 角色。App Studio 將使用此政策來存取已設定的 Amazon Redshift 資源。

建立 IAM 角色，讓 App Studio 存取 Amazon Redshift 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇角色
3. 選擇建立角色。
4. 在信任的實體類型中，選擇自訂信任政策。
5. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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"
        }
    }
}
]
}

```

選擇下一步。

6. 在新增許可中，搜尋並選取您在上一個步驟中建立的策略 (**RedshiftServerlessForAppStudio** 或 **RedshiftProvisionedForAppStudio**)。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。

7. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。

Important

此處的角色名稱必須符合 [步驟 1：建立和設定 Amazon Redshift 資源](#) (*AppBuilderDataAccessRole*) 中 GRANT 命令中使用的角色名稱。

8. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：

- 金鑰：IsAppStudioDataAccessRole
- 值：true

9. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會 [在 App Studio 中建立 Amazon Redshift 連接器](#) 時需要它。

步驟 3：建立 Amazon Redshift 連接器

現在您已設定 Amazon Redshift 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以用來將其應用程式連線至 Amazon Redshift。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

建立 Amazon Redshift 的連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 選擇 Amazon Redshift 連接器。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：提供連接器的名稱。
 - 描述：提供連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon Redshift 資源](#)。如需 IAM 的詳細資訊，請參閱《[IAM 使用者指南](#)》。
 - 區域：選擇 Amazon Redshift 資源所在的 AWS 區域。
 - 運算類型：選擇您是使用 Amazon Redshift Serverless 還是佈建叢集。
 - 叢集或工作群組選擇：如果選擇佈建，請選擇您要連線至 App Studio 的叢集。如果選擇無伺服器，請選擇工作群組。
 - 資料庫選擇：選擇您要連線至 App Studio 的資料庫。
 - 可用的資料表：選取要連線至 App Studio 的資料表。
6. 選擇下一步。檢閱連線資訊，然後選擇建立。
7. 新建立的連接器會出現在連接器清單中。

連線至 Amazon DynamoDB

若要將 App Studio 與 DynamoDB 連線，讓建置器能夠在應用程式中存取和使用 DynamoDB 資源，您必須執行下列步驟：

1. [步驟 1：建立和設定 DynamoDB 資源](#)
2. [步驟 2：使用適當的 DynamoDB 許可建立 IAM 政策和角色](#)

3. [建立 DynamoDB 連接器](#)

步驟 1：建立和設定 DynamoDB 資源

使用下列程序來建立和設定要與 App Studio 搭配使用的 DynamoDB 資源。

設定 DynamoDB 以搭配 App Studio 使用

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/dynamodb/> 開啟 DynamoDB 主控台。

建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。

2. 在左側導覽窗格中，選擇 Tables (資料表)。
3. 選擇 Create Table (建立資料表)。
4. 輸入資料表的名稱和索引鍵。
5. 選擇 Create Table (建立資料表)。
6. 建立資料表之後，請新增一些項目，以便在資料表連線至 App Studio 時顯示。
 - a. 選擇您的資料表，選擇動作，然後選擇探索項目。
 - b. 在傳回的項目中，選擇建立項目。
 - c. (選用)：選擇新增屬性，將更多屬性新增至資料表。
 - d. 輸入每個屬性的值，然後選擇建立項目。

步驟 2：使用適當的 DynamoDB 許可建立 IAM 政策和角色

若要搭配 App Studio 使用 DynamoDB 資源，管理員必須建立 IAM 政策和角色，以授予 App Studio 存取資源的許可。IAM 政策控制建置器可以使用的資料範圍，以及可以針對該資料呼叫哪些操作，例如建立、讀取、更新或刪除。接著，IAM 政策會連接至 App Studio 所使用的 IAM 角色。

我們建議每個服務和政策至少建立一個 IAM 角色。例如，如果建置器在 DynamoDB 中建立兩個由相同資料表支援的應用程式，一個只需要讀取存取權，另一個需要讀取、建立、更新和刪除；管理員應該建立兩個 IAM 角色，一個使用唯讀許可，另一個具有 DynamoDB 中適用資料表的完整 CRUD 許可。

步驟 2a：使用適當的 DynamoDB 許可建立 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

使用適當的 DynamoDB 許可建立 IAM 政策

1. 使用具有建立 [IAM 政策許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 在政策編輯器中，選擇 JSON 選項。
5. 在 JSON 政策文件中輸入或貼上。下列索引標籤包含僅供讀取和完整存取 DynamoDB 資料表的範例政策，以及包含使用 AWS KMS 客戶受管金鑰 (CMK) 加密之 DynamoDB 資料表 AWS KMS 許可的政策範例。

Note

下列政策適用於使用萬用字元 (*) 的所有 DynamoDB 資源*。為了實現最佳安全實務，您應該將萬用字元取代為您要搭配 App Studio 使用之資源的 Amazon Resource Name (ARN)。

Read only

下列政策會授予已設定 DynamoDB 資源的讀取存取權。

JSON

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

```
}
```

Full access

下列政策會授予所設定 DynamoDB 資源的建立、讀取、更新和刪除存取權。

JSON

```
{
  "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

下列政策透過提供 AWS KMS 許可，授予已設定加密 DynamoDB 資源的讀取存取權。您必須以 AWS KMS 金鑰的 ARN 取代 ARN。

JSON

```
{
  "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

下列政策透過提供 AWS KMS 許可，授予已設定加密 DynamoDB 資源的讀取存取權。您必須以 AWS KMS 金鑰的 ARN 取代 ARN。

JSON

```

{
  "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. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **ReadOnlyDDBForAppStudio** 或 **FullAccessDDBForAppStudio**，以及描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 DynamoDB 資源

現在，建立使用您先前建立之政策的 IAM 角色。App Studio 將使用此政策來存取設定的 DynamoDB 資源。

建立 IAM 角色，讓 App Studio 存取 DynamoDB 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。
4. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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"
        }
      }
    }
  ]
}
```

選擇下一步。

5. 在新增許可中，搜尋並選取您在上一個步驟中建立的政策 (**ReadOnlyDDBForAppStudio** 或 **FullAccessDDBForAppStudio**)。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。

6. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
7. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：
 - 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會 [在 App Studio 中建立 DynamoDB 連接器](#) 時需要它。

建立 DynamoDB 連接器

現在您已設定 DynamoDB 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以用來將其應用程式連線至 DynamoDB。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

為 DynamoDB 建立連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 從連接器類型清單中選擇 Amazon DynamoDB。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 DynamoDB 連接器的名稱。
 - 描述：輸入 DynamoDB 連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2b：建立 IAM 角色，讓 App Studio 存取 DynamoDB 資源](#)。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。
 - 區域：選擇 DynamoDB 資源所在的 AWS 區域。
 - 可用的資料表：選取要連線至 App Studio 的資料表。
6. 選擇下一步。檢閱連線資訊，然後選擇建立。
7. 新建立的連接器會出現在連接器清單中。

連線至 AWS Lambda

若要將 App Studio 與 Lambda 連線，讓建置器能夠在應用程式中存取和使用 Lambda 資源，您必須執行下列步驟：

1. [步驟 1：建立和設定 Lambda 函數](#)
2. [步驟 2：建立 IAM 角色，讓 App Studio 存取 Lambda 資源](#)

3. [步驟 3：建立 Lambda 連接器](#)

步驟 1：建立和設定 Lambda 函數

如果您沒有現有的 Lambda 函數，您必須先建立它們。若要進一步了解如何建立 Lambda 函數，請參閱 [AWS Lambda 開發人員指南](#)。

步驟 2：建立 IAM 角色，讓 App Studio 存取 Lambda 資源

若要搭配 App Studio 使用 Lambda 資源，管理員必須建立 IAM 角色，以授予 App Studio 存取資源的許可。IAM 角色控制應用程式可以從 Lambda 存取的資源或操作。

我們建議每個服務和政策至少建立一個 IAM 角色。

建立 IAM 角色，讓 App Studio 存取 Lambda 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者[建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。
4. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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"
      }
    }
  }
]
}

```

選擇下一步。

5. 在新增許可中，搜尋並選取授予角色適當許可的政策。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。對於 Lambda，您可以考慮新增AWSLambdaRole政策，這會授予叫用 Lambda 函數的許可。

如需搭配 Lambda 使用 IAM 政策的詳細資訊，包括受管政策的清單及其說明，請參閱《AWS Lambda 開發人員指南》中的[適用於的 Identity and Access Management AWS Lambda](#)。

選擇下一步。

6. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
7. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：
 - 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會在[App Studio 中建立 Lambda 連接器](#)時需要它。

步驟 3：建立 Lambda 連接器

現在您已設定 Lambda 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以用來將其應用程式連線至 Lambda。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

建立 Lambda 的連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 從連接器類型清單中選擇 AWS 其他服務。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 Lambda 連接器的名稱。
 - 描述：輸入 Lambda 連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2：建立 IAM 角色，讓 App Studio 存取 Lambda 資源](#)。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。
 - 服務：選擇 Lambda。
 - 區域：選擇 Lambda 資源所在的 AWS 區域。
6. 選擇建立。
7. 新建立的連接器會出現在連接器清單中。

連線至 Amazon Simple Storage Service (Amazon S3)

若要將 App Studio 與 Amazon S3 連線，讓建置器能夠在應用程式中存取和使用 Amazon S3 資源，請執行下列步驟：

1. [步驟 1：建立和設定 Amazon S3 資源](#)
2. [步驟 2：使用適當的 Amazon S3 許可建立 IAM 政策和角色](#)
3. [步驟 3：建立 Amazon S3 連接器](#)

在您完成步驟並建立具有適當許可的連接器之後，建置器可以使用連接器來建立與 Amazon S3 資源互動的應用程式。如需在 App Studio 應用程式中與 Amazon S3 互動的詳細資訊，請參閱 [與 Amazon Simple Storage Service 與元件和自動化互動](#)。

步驟 1：建立和設定 Amazon S3 資源

根據您的應用程式需求和現有資源，您可能需要建立 Amazon S3 儲存貯體，供應用程式寫入和讀取。如需建立 Amazon S3 資源的資訊，包括儲存貯體，請參閱 [《Amazon Simple Storage Service 使用者指南》](#) 中的 [Amazon S3 入門](#)。

若要在應用程式中使用 [S3 上傳](#) 元件，您必須將跨來源資源共用 (CORS) 組態新增至您要上傳的任何 Amazon S3 儲存貯體。CORS 組態提供 App Studio 將物件推送至儲存貯體的許可。下列程序詳細說明如何使用主控台將 CORS 組態新增至 Amazon S3 儲存貯體。如需 CORS 及其設定的詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的 [使用跨來源資源共用 \(CORS\)](#)。

在主控台中將 CORS 組態新增至 Amazon S3 儲存貯體

1. 導覽至 <https://console.aws.amazon.com/s3/> 中的儲存貯體。
2. 選擇許可索引標籤。
3. 在跨來源資源共用 (CORS) 中，選擇編輯。
4. 新增下列程式碼片段：

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

5. 選擇儲存變更。

步驟 2：使用適當的 Amazon S3 許可建立 IAM 政策和角色

若要搭配 App Studio 使用 Amazon S3 資源，管理員必須建立 IAM 政策和角色，以授予 App Studio 存取資源的許可。IAM 政策控制建置器可以使用的資料範圍，以及可以針對該資料呼叫哪些操作，例如建立、讀取、更新或刪除。接著，IAM 政策會連接至 App Studio 所使用的 IAM 角色。

我們建議每個服務和政策至少建立一個 IAM 角色。例如，如果建置器在 Amazon S3 中建立兩個由不同儲存貯體支援的應用程式，管理員應該建立兩個 IAM 政策和角色，每個儲存貯體各一個。

步驟 2a：使用適當的 Amazon S3 許可建立 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

使用適當的 Amazon S3 許可建立 IAM 政策

1. 使用具有建立 [IAM 政策許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 在政策編輯器中，選擇 JSON 選項。
5. 在 JSON 政策文件中輸入或貼上。下列索引標籤包含唯讀和完整存取 Amazon S3 資源的範例政策。

Note

下列政策適用於使用萬用字元 (*) 的所有 Amazon S3 資源*。為了實現最佳安全實務，您應該將萬用字元取代為要與 App Studio 搭配使用的資源的 Amazon Resource Name (ARN)，例如儲存貯體或資料夾。

Read only

下列政策會將唯讀存取權（取得和列出）授予已設定的 Amazon S3 儲存貯體或資料夾。

JSON

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

```
    }  
  ]  
}
```

Full access

下列政策會將完整存取權 (put、get、list 和 delete) 授予已設定的 Amazon S3 儲存貯體或資料夾。

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "S3FullAccessForAppStudio",  
      "Effect": "Allow",  
      "Action": [  
        "s3:PutObject",  
        "s3:GetObject",  
        "s3:ListBucket",  
        "s3:DeleteObject"  
      ],  
      "Resource": "*"   
    }  
  ]  
}
```

6. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **AWSAppStudioS3FullAccess** 和描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon S3 資源

若要搭配 App Studio 使用 Amazon S3 資源，管理員必須建立 IAM 角色，以授予 App Studio 存取資源的許可。IAM 角色控制建置器可以使用的資料範圍，以及可以針對該資料呼叫哪些操作，例如建立、讀取、更新或刪除。

我們建議每個服務和政策至少建立一個 IAM 角色。

建立 IAM 角色，讓 App Studio 存取 Amazon S3 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。
4. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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":
"111111111-2222-3333-4444-5555555555"
        }
      }
    }
  ]
}
```

選擇下一步。

5. 在新增許可中，搜尋並選取您在上一個步驟中建立的政策 (**S3ReadOnlyForAppStudio** 或 **S3FullAccessForAppStudio**)。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。
6. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
7. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：
 - 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色並記下產生的 Amazon Resource Name (ARN)，在下一個步驟中，您將需要它在 App Studio 中建立 Amazon S3 連接器。

步驟 3：建立 Amazon S3 連接器

現在您已設定 Amazon S3 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以用來將其應用程式連線至 Amazon S3。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

建立 Amazon S3 的連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 選擇 Amazon S3 連接器。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 Amazon S3 連接器的名稱。
 - 描述：輸入 Amazon S3 連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon S3 資源](#)。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。

- 區域：選擇 Amazon S3 資源所在的 AWS 區域。
6. 選擇建立。
 7. 新建立的連接器會出現在連接器清單中。

連線至 Amazon Aurora

若要將 App Studio 與 Aurora 連線，讓建置器能夠在應用程式中存取和使用 Aurora 資源，您必須執行下列步驟：

1. [步驟 1：建立和設定 Aurora 資源](#)
2. [步驟 2：建立具有適當 Aurora 許可的 IAM 政策和角色](#)
3. [步驟 3：在 App Studio 中建立 Aurora 連接器](#)

App Studio 支援下列 Aurora 版本：

- Aurora MySQL Serverless V1：5.7.2
- Aurora PostgreSQL Serverless V1：11.18、13.9
- Aurora MySQL Serverless V2：13.11 或更新版本、14.8 或更新版本，以及 15.3 或更新版本
- Aurora PostgreSQL Serverless V2：13.11 或更高版本、14.8 或更高版本，以及 15.3 或更高版本

步驟 1：建立和設定 Aurora 資源

若要搭配 App Studio 使用 Aurora 資料庫，您必須先建立資料庫並適當設定資料庫。App Studio 支援兩種 Aurora 資料庫類型：Aurora PostgreSQL 和 Aurora MySQL。若要比較類型，請參閱 [MySQL 和 PostgreSQL 之間的差異為何？](#)。選擇適當的索引標籤，並依照程序設定 Aurora 以搭配 App Studio 應用程式使用。

Aurora PostgreSQL

使用下列程序來建立和設定要與 App Studio 搭配使用的 Aurora PostgreSQL 資料庫叢集。

設定 Aurora 以搭配 App Studio 使用

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/rds/>：// 開啟 Amazon RDS 主控台。
2. 選擇建立資料庫。

3. 選擇 Aurora (PostgreSQL 相容)。
4. 在可用版本中，選擇大於或等於版本 13.11、14.8 和的任何版本 15.3。
5. 在設定中，輸入資料庫叢集識別符。
6. 在執行個體組態中，選擇 Serverless v2，然後選擇適當的容量。
7. 在連線中，選取啟用 RDS 資料 API。
8. 在資料庫身分驗證中，選取 IAM 資料庫身分驗證。
9. 在其他組態中，在初始資料庫名稱中，輸入資料庫的初始資料庫名稱。

Aurora MySQL

使用下列程序來建立和設定要與 App Studio 搭配使用的 Aurora MySQL 資料庫叢集。

Aurora MySQL 不支援從支援資料 API 或無伺服器 v1 的 UI 建立版本。若要建立支援資料 API 的 Aurora MySQL 叢集，您必須使用 AWS CLI。

Note

若要將 Aurora MySQL 資料庫與 App Studio 搭配使用，它們必須位於虛擬私有雲端 (VPC) 中。如需詳細資訊，請參閱《Amazon Aurora 使用者指南》中的 [在 VPC 中使用資料庫叢集](#)。

設定 Aurora MySQL 以搭配 App Studio 使用

1. 如有必要，AWS CLI 請依照《AWS Command Line Interface 使用者指南》中的安裝 [或更新至最新版本的指示來安裝 AWS CLI](#)。
2. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/rds/> : // 開啟 Amazon RDS 主控台。
3. 在左側導覽中，選擇子網路群組。
4. 選擇 Create DB Subnet Group (建立資料庫子網路群組)。
5. 填寫資訊並建立太陽網群組。如需子網路群組和使用 VPCs 的詳細資訊，請參閱《Amazon Aurora 使用者指南》中的在 [VPC 中使用資料庫叢集](#)。
6. 執行下列 AWS CLI 命令：

```
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
```

取代下列欄位：

- 將 *db_name* 取代為所需的資料庫名稱。
- 將 *db_cluster_identifier* 取代為所需的資料庫叢集識別符。
- (選用) 視需要取代 `scaling-configuration` 欄位中的數字。
- 以所需的使用者名稱取代 *userName*。
- 以所需的密碼取代 *userPass*。
- 在 `availability-zones`，從您建立的子網路群組新增可用區域。
- 將 *subnet-group-name* 取代為您建立的子網路群組名稱。

步驟 2：建立具有適當 Aurora 許可的 IAM 政策和角色

若要將 Aurora 資源與 App Studio 搭配使用，管理員必須建立 IAM 政策並將其連接到 IAM 角色，該角色用於授予 App Studio 存取所設定資源的許可。IAM 政策和角色控制建置器可以使用的資料範圍，以及可以針對該資料呼叫哪些操作，例如建立、讀取、更新或刪除。

我們建議每個服務和政策至少建立一個 IAM 角色。

步驟 2a：建立具有適當 Aurora 許可的 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

建立具有適當 Aurora 許可的 IAM 政策

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 `中` 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。

4. 在政策編輯器中，選擇 JSON 選項。
5. 將現有的程式碼片段取代為下列程式碼片段，將 AWS **111122223333** 取代為包含 Amazon Redshift 和 Aurora 資源的帳號。

JSON

```
{
  "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. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **Aurora_AppStudio** 和 描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 Aurora 資源

現在，建立使用您先前建立之政策的 IAM 角色。App Studio 將使用此政策來存取已設定的 Aurora 資源。

建立 IAM 角色，讓 App Studio 存取 Aurora 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。

- 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為 AWS 帳戶 ID。
- 將 **11111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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":
            "11111111-2222-3333-4444-5555555555"
        }
      }
    }
  ]
}
```

選擇下一步。

- 在新增許可中，搜尋並選取您先前建立的政策 (**Aurora_AppStudio**)。選擇政策旁的 + 會展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。

- 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
- 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：

- 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會在 [App Studio 中建立 Aurora 連接器](#) 時需要它。

步驟 3：在 App Studio 中建立 Aurora 連接器

現在您已設定 Aurora 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以用來將其應用程式連線至 Aurora。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

為 Aurora 建立連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 選擇 Amazon Aurora 連接器。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 Aurora 連接器的名稱。
 - 描述：輸入 Aurora 連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN) [步驟 2b：建立 IAM 角色，讓 App Studio 存取 Aurora 資源](#)。如需 IAM 的詳細資訊，請參閱《[IAM 使用者指南](#)》。
 - 秘密 ARN：輸入資料庫叢集的秘密 ARN。如需有關在何處尋找秘密 ARN 的資訊，請參閱《[Amazon Aurora 使用者指南](#)》中的 [檢視資料庫 cluster 秘密的詳細資訊](#)。
 - 區域：選擇 Aurora 資源所在的 AWS 區域。
 - 資料庫 ARN：輸入資料庫叢集的 ARN。您可以在資料庫叢集的組態索引標籤中找到 ARN，類似於秘密 ARN。
 - 資料庫類型：選擇與在 中建立的資料庫類型相符的資料庫類型 MySQL 或 PostgreSQL [步驟 1：建立和設定 Aurora 資源](#)。

- 資料庫名稱：輸入資料庫的名稱，也可以在資料庫叢集的組態索引標籤中找到。
 - 可用的資料表：使用此連接器選取要搭配 App Studio 使用的資料表。
6. 選擇下一步以檢閱或定義實體映射。
 7. 選擇建立以建立 Aurora 連接器。新建立的連接器會出現在連接器清單中。

連線至 Amazon Bedrock

若要將 App Studio 與 Amazon Bedrock 連線，讓建置器可以在應用程式中存取和使用 Amazon Bedrock，您必須執行下列步驟：

1. [步驟 1：啟用 Amazon Bedrock 模型](#)
2. [步驟 2：使用適當的 Amazon Bedrock 許可建立 IAM 政策和角色](#)
3. [步驟 3：建立 Amazon Bedrock 連接器](#)

步驟 1：啟用 Amazon Bedrock 模型

使用下列程序來啟用 Amazon Bedrock 模型。

啟用 Amazon Bedrock 模型

1. 登入 AWS 管理主控台，並在 <https://console.aws.amazon.com/bedrock/> 開啟 Amazon Bedrock 主控台。
2. 在左側導覽窗格中，選擇模型存取。
3. 啟用您要使用的模型。如需詳細資訊，請參閱[管理對 Amazon Bedrock 基礎模型的存取](#)。

步驟 2：使用適當的 Amazon Bedrock 許可建立 IAM 政策和角色

若要搭配 App Studio 使用 Amazon Bedrock 資源，管理員必須建立 IAM 政策和角色，以授予 App Studio 存取資源的許可。IAM 政策會控制可以針對這些資源呼叫哪些資源和哪些操作，例如 InvokeModel。接著，IAM 政策會連接至 App Studio 所使用的 IAM 角色。

步驟 2a：使用適當的 Amazon Bedrock 許可建立 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

使用適當的 Amazon Bedrock 許可建立 IAM 政策

1. 使用具有建立 [IAM 政策許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 在政策編輯器中，選擇 JSON 選項。
5. 在 JSON 政策文件中輸入或貼上。下列範例政策會在所有 Amazon Bedrock 資源 `InvokeModel` 上使用萬用字元 (*) 提供。

為了實現最佳安全實務，您應該將萬用字元取代為您要搭配 App Studio 使用之資源的 Amazon Resource Name (ARN)。

JSON

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

6. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **BedrockAccessForAppStudio** 和描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon Bedrock

若要將 Amazon Bedrock 與 App Studio 搭配使用，管理員必須建立 IAM 角色，以授予 App Studio 存取資源的許可。IAM 角色控制 App Studio 應用程式要使用的許可範圍，並在建立連接器時使用。我們建議每個服務和政策至少建立一個 IAM 角色。

建立 IAM 角色，讓 App Studio 存取 Amazon Bedrock

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。
4. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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":
            "111111111-2222-3333-4444-5555555555"
        }
      }
    }
  ]
}
```

選擇下一步。

5. 在新增許可中，搜尋並選取您在上一個步驟中建立的政策 (**BedrockAccessForAppStudio**)。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。
6. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
7. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：
 - 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，在下一個步驟中，您需要在 App Studio 中建立 Amazon Bedrock 連接器。

步驟 3：建立 Amazon Bedrock 連接器

現在您已設定 Amazon Bedrock 資源和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以使用該連接器將其應用程式連接到 Amazon Bedrock。


Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

建立 Amazon Bedrock 的連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. AWS 從連接器類型清單中選擇其他服務。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 Amazon Bedrock 連接器的名稱。
 - 描述：輸入 Amazon Bedrock 連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon Bedrock](#)。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。

- 服務：選擇 Bedrock 執行時間。

 Note

Bedrock 執行期用於對 Amazon Bedrock 中託管的模型提出推論請求，而 Bedrock 用於管理、訓練和部署模型。

- 區域：選擇 Amazon Bedrock 資源所在的 AWS 區域。
6. 選擇建立。
 7. 新建立的連接器會出現在連接器清單中。

連線至 Amazon Simple Email Service

若要將 App Studio 與 Amazon SES 連線，讓建置器使用它從其應用程式傳送電子郵件通知，您必須執行下列步驟：

1. [步驟 1：設定 Amazon SES 資源](#)
2. [步驟 2：使用適當的 Amazon SES 許可建立 IAM 政策和角色](#)
3. [步驟 3：建立 Amazon SES 連接器](#)

步驟 1：設定 Amazon SES 資源

如果您尚未設定，您必須先將 Amazon SES 設定為使用它來傳送電子郵件。若要進一步了解如何設定 Amazon SES，請參閱 [《Amazon Simple Email Service 開發人員指南》](#) 中的 Amazon Simple Email Service 入門。

步驟 2：使用適當的 Amazon SES 許可建立 IAM 政策和角色

若要搭配 App Studio 使用 Amazon SES 資源，管理員必須建立 IAM 角色，以授予 App Studio 存取資源的許可。IAM 角色控制可在 App Studio 應用程式中使用的 Amazon SES 函數或資源。

我們建議每個服務和政策至少建立一個 IAM 角色。

步驟 2a：使用適當的 Amazon SES 許可建立 IAM 政策

您建立並與 App Studio 搭配使用的 IAM 政策應僅包含應用程式遵循最佳安全實務之適當資源的最低必要許可。

使用適當的 Amazon SES 許可建立 IAM 政策

1. 使用具有建立 [IAM 政策許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇政策。
3. 選擇建立政策。
4. 在政策編輯器中，選擇 JSON 選項。
5. 在下列 JSON 政策文件中輸入或貼上。

Note

下列政策適用於使用萬用字元 (*) 的所有 Amazon SES 資源*。為了實現最佳安全實務，您應該將萬用字元取代為您要搭配 App Studio 使用之資源的 Amazon Resource Name (ARN)。

JSON

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

6. 選擇下一步。
7. 在檢閱和建立頁面上，提供政策名稱，例如 **SESForAppStudioPolicy** 和描述（選用）。
8. 選擇建立政策以建立政策。

步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon SES

現在，建立使用您先前建立之政策的 IAM 角色。App Studio 將使用此政策來存取 Amazon SES。

建立 IAM 角色以授予 App Studio 對 Amazon SES 的存取權

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 [中](#) 建立的管理使用者 [建立管理使用者以管理 AWS 資源](#)。
2. 在左側導覽窗格中，選擇角色
3. 選擇建立角色。
4. 在信任的實體類型中，選擇自訂信任政策。
5. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **11111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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":
            "11111111-2222-3333-4444-5555555555"
        }
      }
    }
  ]
}
```

選擇下一步。

6. 在新增許可中，搜尋並選取您在上一個步驟中建立的政策 (**SESForAppStudioPolicy**)。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。

選擇下一步。

7. 在名稱、檢閱和建立頁面上，提供角色名稱和描述。
8. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：
 - 金鑰：IsAppStudioDataAccessRole
 - 值：true
9. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會在 [App Studio 中建立 Amazon SES 連接器](#) 時需要它。

步驟 3：建立 Amazon SES 連接器

現在您已設定 Amazon SES 和 IAM 政策和角色，請使用該資訊在 App Studio 中建立連接器，建置器可以使用該連接器在其應用程式中使用 Amazon SES。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

建立 Amazon SES 的連接器

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
3. 選擇 + 建立連接器。
4. 從連接器類型清單中選擇 AWS 其他服務。
5. 填寫下列欄位來設定您的連接器：
 - 名稱：輸入 Amazon SES 連接器的名稱。
 - 描述：輸入 Amazon SES 連接器的描述。

- IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN)[步驟 2b：建立 IAM 角色，讓 App Studio 存取 Amazon SES](#)。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。
 - 服務：選擇 Simple Email Service。
 - 區域：選擇 Amazon SES 資源所在的 AWS 區域。
6. 選擇建立。
 7. 新建立的連接器會出現在連接器清單中。

AWS 使用其他服務連接器連線至 AWS 服務

雖然 App Studio 提供某些特定 AWS 服務的特定連接器，但您也可以 AWS 使用其他服務 AWS 連接器連線到其他服務。

Note

如果 AWS 可用，建議使用服務特定的連接器。

若要將 App Studio 與 AWS 服務連線，讓建置器能夠在應用程式中存取和使用服務的資源，您必須執行下列步驟：

1. [建立 IAM 角色，讓 App Studio 存取 AWS 資源](#)
2. [建立其他服務 AWS 連接器](#)

建立 IAM 角色，讓 App Studio 存取 AWS 資源

若要搭配 App Studio 使用 AWS 服務和資源，管理員必須建立 IAM 角色，以授予 App Studio 存取資源的許可。IAM 角色控制建置器可存取的資源範圍，以及可以針對資源呼叫哪些操作。我們建議每個服務和政策至少建立一個 IAM 角色。

建立 IAM 角色，讓 App Studio 存取 AWS 資源

1. 使用具有建立 [IAM 角色許可的使用者登入 IAM 主控台](#)。建議使用在 中建立的管理使用者[建立管理使用者以管理 AWS 資源](#)。
2. 在主控台的導覽窗格中，選擇 Roles (角色)，然後選擇 Create role (建立角色)。
3. 在信任的實體類型中，選擇自訂信任政策。
4. 將預設政策取代為下列政策，以允許 App Studio 應用程式在您的帳戶中擔任此角色。

您必須取代政策中的下列預留位置。您可以在 App Studio 的帳戶設定頁面中找到要使用的值。

- 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
- 將 **111111111-2222-3333-4444-5555555555** 取代為您的 App Studio 執行個體 ID，在 App Studio 執行個體的帳戶設定中列為執行個體 ID。

JSON

```
{
  "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":
            "111111111-2222-3333-4444-5555555555"
        }
      }
    }
  ]
}
```

選擇下一步。

5. 在新增許可中，搜尋並選取授予角色適當許可的政策。選擇政策旁的 + 將展開政策，以顯示其授予的許可，並選擇核取方塊會選取政策。如需 IAM 的詳細資訊，請參閱 [《IAM 使用者指南》](#)。

選擇下一步。

6. 在角色詳細資訊中，提供名稱和描述。
7. 在步驟 3：新增標籤中，選擇新增標籤以新增下列標籤，以提供 App Studio 存取：

- 金鑰：IsAppStudioDataAccessRole
 - 值：true
8. 選擇建立角色，並記下產生的 Amazon Resource Name (ARN)，您會在 [App Studio 中建立其他服務 AWS 連接器](#) 時需要它。

建立其他服務 AWS 連接器

現在您已設定 IAM 角色，請使用該資訊在 App Studio 中建立連接器，建置器可以使用該連接器將其應用程式連接到 服務和資源。

Note

您必須在 App Studio 中具有管理員角色，才能建立連接器。

AWS 使用其他服務連接器連線至 AWS 服務

1. 導覽至 App Studio。
2. 在左側導覽窗格中，在管理區段中選擇連接器。
3. 選擇 + 建立連接器。
4. 在支援的 AWS 服務清單的連接器區段中選擇其他服務。AWS
5. 填寫下列欄位來設定您的 AWS 服務連接器：
 - 名稱：提供連接器的名稱。
 - 描述：提供連接器的描述。
 - IAM 角色：從在 中建立的 IAM 角色輸入 Amazon Resource Name (ARN) [建立 IAM 角色](#)，讓 [App Studio 存取 AWS 資源](#)。
 - 服務：選取您要連線至 App Studio AWS 的服務。
 - 區域：選取 AWS 資源所在的 AWS 區域。
6. 選擇建立。新建立的連接器會出現在連接器清單中。

搭配 CMKs 使用加密的資料來源

本主題包含有關設定 App Studio 並將其連線至使用 [AWS KMS 客戶受管金鑰 \(CMK\)](#) 加密的資料來源的資訊。

內容

- [使用加密的受管資料儲存資料表](#)
- [使用加密的 DynamoDB 資料表](#)

使用加密的受管資料儲存資料表

使用下列程序來加密 App Studio 應用程式中受管儲存實體使用的 DynamoDB 資料表。如需受管資料實體的詳細資訊，請參閱 [AWS App Studio 中的受管資料實體](#)。

使用加密的受管資料儲存資料表


1. 如有必要，請在 App Studio 的應用程式中建立受管資料實體。如需詳細資訊，請參閱 [使用 App Studio 受管資料來源建立實體](#)。
2. 透過執行下列步驟，將具有使用 CMK 加密和解密資料表資料許可的政策陳述式新增至 IAM AppStudioManagedStorageDDBAccess 角色：
 - a. 前往 <https://console.aws.amazon.com/iam/> 開啟 IAM 主控台。

Important

您必須使用與建立 App Studio 執行個體相同的帳戶。

- b. 在 IAM 主控台的導覽窗格中，選擇角色。
 - c. 選擇 AppStudioManagedStorageDDBAccess。
 - d. 在許可政策中，選擇新增許可，然後選擇建立內嵌政策。
 - e. 選擇 JSON 並以下列政策取代內容，並取代以下內容：
 - 將 **111122223333** 取代為用於設定 App Studio 執行個體的帳戶的 AWS 帳號，在 App Studio 執行個體的帳戶設定中列為帳戶 AWS ID。
 - 將 **CMK_id** 取代為 CMK ID。若要尋找，請參閱 [尋找金鑰 ID 和金鑰 ARN](#)。
3. 透過執行下列步驟，加密 App Studio 受管資料實體所使用的 DynamoDB 資料表：
 - a. 在 <https://console.aws.amazon.com/dynamodbv2/> 開啟 Amazon DynamoDB 主控台。
 - b. 選擇您要加密的資料表。您可以在 App Studio 中對應實體的連線索引標籤中找到資料表名稱。
 - c. 選擇其他設定。

- d. 在加密中，選擇管理加密。
 - e. 選擇存放在您的帳戶，由您擁有和管理，然後選取您的 CMK。
4. 透過重新發佈您的應用程式，並確保讀取和寫入資料在測試和生產環境中都正常運作，以及在另一個實體中使用此資料表來測試您的變更。

 Note

根據預設，任何新增的受管資料實體都會使用 DynamoDB 受管金鑰，而且必須依照先前步驟，使用 CMK 更新為。

使用加密的 DynamoDB 資料表

使用下列程序來設定要在 App Studio 應用程式中使用的加密 DynamoDB 資料表。

使用加密的 DynamoDB 資料表

1. 請依照 中的指示 [步驟 1：建立和設定 DynamoDB 資源](#) 進行下列變更：
 - 設定要加密的資料表。如需詳細資訊，請參閱《Amazon DynamoDB 開發人員指南》中的 [指定新資料表的加密金鑰](#)。
2. 遵循 中的指示 [步驟 2：使用適當的 DynamoDB 許可建立 IAM 政策和角色](#)，然後使用 新增政策陳述式以更新新角色的許可政策，允許它透過執行下列步驟，使用 CMK 加密和解密資料表資料：
 - a. 如有必要，請在 IAM 主控台中導覽至您的角色。
 - b. 在許可政策中，選擇新增許可，然後選擇建立內嵌政策。
 - c. 選擇 JSON 並以下列政策取代內容，並取代以下內容：
 - 將 `team_account_id` 取代為您的 App Studio 團隊 ID，可在您的帳戶設定中找到。
 - 將 `CMK_id` 取代為 CMK ID。若要尋找，請參閱 [尋找金鑰 ID 和金鑰 ARN](#)。
3. 遵循 [建立 DynamoDB 連接器](#) 和 中的指示，使用您先前建立的角色來建立連接器。
4. 透過發佈使用 DynamoDB 連接器和資料表的應用程式來測試組態，以進行測試或生產。確保讀取和寫入資料有效，並且使用此資料表建立另一個實體也有效。

Note

建立任何新的 DynamoDB 資料表時，您必須依照上述步驟，將它們設定為使用 CMK 加密。

連線至第三方服務

主題

- [OpenAPI Connector 與 API Connector](#)
- [連線至第三方服務和 APIs \(一般 \)](#)
- [使用 OpenAPI 連線至 服務](#)
- [連線至 Salesforce](#)

OpenAPI Connector 與 API Connector

若要從 App Studio 應用程式將 API 請求傳送至第三方服務，您必須建立並設定應用程式用來驗證服務的連接器，並設定 API 呼叫。App Studio 同時提供 API Connector 和 OpenAPI Connector 連接器類型來完成此操作，如下所述：

- API Connector：用於設定任何類型的 REST API 的身分驗證和請求資訊。
- OpenAPI 連接器：用於設定已採用 OpenAPI 規格 (OAS) APIs 的身分驗證和請求資訊。遵守 OAS 的 APIs 提供數種優點，包括標準化、安全性、控管和文件。

App Studio 建議 OpenAPI Connector 針對任何遵循 OAS 的 APIs 使用，並提供 OpenAPI 規格檔案。如需 OpenAPI 的詳細資訊，請參閱 [什麼是 OpenAPI?](#) 在 Swagger 文件中。

連線至第三方服務和 APIs (一般)

使用下列程序在 App Studio 中建立一般 API Connector。API Connector 用於為 App Studio 應用程式提供第三方服務、資源或操作的存取權。

使用 API Connector 連線至第三方服務

1. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。

2. 選擇 + 建立連接器。
3. 選擇 API Connector。現在，填寫下列欄位來設定您的連接器。
4. 連接器名稱：提供連接器的名稱。
5. 連接器描述：為您的連接器提供描述。
6. 基本 URL：第三方連線的網站或主機。例如 `www.slack.com`。
7. 身分驗證方法：選擇使用目標服務進行身分驗證的方法。
 - 無：在沒有身分驗證的情況下存取目標服務。
 - 基本：使用從要連線的服務取得的使用者名稱和密碼來存取目標服務。
 - 承載權杖：使用從服務使用者帳戶或 API 設定取得的身分驗證權杖的權杖值來存取目標服務。
 - OAuth 2.0：使用 OAuth 2.0 通訊協定存取目標服務，這會授予 App Studio 存取服務和資源的權限，而無需共用任何登入資料或身分。若要使用 OAuth 2.0 身分驗證方法，您必須先從連接 App Studio 的服務建立應用程式，以取得必要資訊。使用該資訊，填寫下列欄位：
 - a. 用戶端登入資料流程：非常適合 system-to-system 互動，其中應用程式在沒有使用者互動的情況下代表自己運作。例如，根據使用者新增的新記錄自動更新 Salesforce 記錄的 CRM 應用程式，或在報告中擷取和顯示交易資料的應用程式。
 1. 在用戶端 ID 中，輸入從目標服務中建立的 OAuth 應用程式取得的 ID。
 2. 在用戶端秘密中，輸入從目標服務中建立的 OAuth 應用程式取得的秘密。
 3. 在存取字符 URL 中，輸入從目標服務中建立的 OAuth 應用程式取得的字符 URL。
 4. 或者，在範圍中，輸入應用程式的範圍。範圍是應用程式所需的許可或存取層級。請參閱目標服務的 API 文件以了解其範圍，並僅設定 App Studio 應用程式所需的範圍。
 - b. 授權碼流程：適用於需要代表使用者採取行動的應用程式。例如，使用者登入並檢視和更新支援票證的客戶支援應用程式，或每個團隊成員登入以檢視和管理其銷售資料的銷售應用程式。
 1. 在用戶端 ID 中，輸入從目標服務中建立的 OAuth 應用程式取得的 ID。
 2. 在用戶端秘密中，輸入從目標服務中建立的 OAuth 應用程式取得的秘密。
 3. 在授權 URL 中，輸入目標服務的授權 URL。
 4. 在存取字符 URL 中，輸入從目標服務中建立的 OAuth 應用程式取得的字符 URL。

5. 或者，在範圍中，輸入應用程式的範圍。範圍是應用程式所需的許可或存取層級。請參閱目標服務的 API 文件以了解其範圍，並僅設定 App Studio 應用程式所需的範圍。
8. 標頭：新增 HTTP 標頭，用於提供請求或回應的中繼資料。您可以同時新增索引鍵和值，或僅提供建置器可在應用程式中提供值的索引鍵。
9. 查詢參數：新增用於傳遞選項、篩選條件或資料的查詢參數，做為請求 URL 的一部分。如同標頭，您可以同時提供金鑰和值，或僅提供建置器可在應用程式中提供值的金鑰。
10. 選擇建立。新建立的連接器會出現在連接器清單中。

現在連接器已建立，建置器可以在其應用程式中使用它。

使用 OpenAPI 連線至 服務

若要使用 OpenAPI 將 App Studio 與服務連線，讓建置器能夠建置應用程式，以傳送請求並從服務接收回應，請執行下列步驟：

1. [取得 OpenAPI 規格檔案並收集服務資訊](#)
2. [建立 OpenAPI 連接器](#)

取得 OpenAPI 規格檔案並收集服務資訊

若要使用 OpenAPI 將服務連線至 App Studio，請執行下列步驟：

1. 前往您要連線至 App Studio 的服務，並尋找 OpenAPI 規格 JSON 檔案。

Note

App Studio 支援符合 OpenAPI 規格 3.0.0 版或更新版本的 OpenAPI 規格檔案。

2. 收集必要的資料以設定 OpenAPI 連接器，包括下列項目：
 - 連接到服務的基本 URL。
 - 身分驗證登入資料，例如字符或使用名稱/密碼。
 - 如果適用，則為任何標頭。
 - 如果適用，則為任何查詢參數。

建立 OpenAPI 連接器

建立 OpenAPI 的連接器

1. 導覽至 App Studio。
 2. 在左側導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
 3. 選擇 + 建立連接器。
 4. 從連接器類型清單中選擇 OpenAPI Connector。現在，填寫下列欄位來設定您的連接器。
 5. 名稱：輸入 OpenAPI 連接器的名稱。
 6. 描述：輸入 OpenAPI 連接器的描述。
 7. 基本 URL：輸入連接到服務的基本 URL。
 8. 身分驗證方法：選擇使用目標服務進行身分驗證的方法。
 - 無：在沒有身分驗證的情況下存取目標服務。
 - 基本：使用從要連線的服務取得的使用者名稱和密碼來存取目標服務。
 - 承載權杖：使用從服務使用者帳戶或 API 設定取得的身分驗證權杖的權杖值來存取目標服務。
 - OAuth 2.0：使用 OAuth 2.0 通訊協定存取目標服務，這會授予 App Studio 存取服務和資源的權限，而無需共用任何登入資料或身分。若要使用 OAuth 2.0 身分驗證方法，您必須先從連接 App Studio 的服務建立應用程式，以取得必要資訊。使用該資訊，填寫下列欄位：
 - a. 用戶端登入資料流程：
 1. 在用戶端 ID 中，輸入目標服務的 ID。
 2. 在用戶端秘密中，輸入目標服務的秘密。
 3. 在存取字符 URL 中，輸入目標服務的字符 URL。
 4. 或者，在範圍中，輸入應用程式的範圍。範圍是應用程式所需的許可或存取層級。請參閱目標服務的 API 文件以了解其範圍，並僅設定 App Studio 應用程式所需的範圍。
- 新增要在每次呼叫時與服務一起傳送的任何變數，然後選擇驗證連線以測試身分驗證和連線。
- b. 授權碼流程：
 1. 在用戶端 ID 中，輸入目標服務的 ID。
 2. 在用戶端秘密中，輸入目標服務的秘密。

3. 在授權 URL 中，輸入目標服務的授權 URL。
 4. 在存取字符 URL 中，輸入目標服務的字符 URL。
 5. 或者，在範圍中，輸入應用程式的範圍。範圍是應用程式所需的許可或存取層級。請參閱目標服務的 API 文件以了解其範圍，並僅設定 App Studio 應用程式所需的範圍。
9. 變數：新增要在每次呼叫時傳送至服務的變數。在組態期間新增的變數會安全地存放，且只能在使用連線的應用程式執行期間存取。
 10. 標頭：新增 HTTP 標頭，用於提供請求或回應的中繼資料。您可以同時新增索引鍵和值，或僅提供建置器可在應用程式中提供值的索引鍵。
 11. 查詢參數：新增用於傳遞選項、篩選條件或資料的查詢參數，做為請求 URL 的一部分。如同標頭，您可以同時提供金鑰和值，或僅提供建置器可在應用程式中提供值的金鑰。
 12. OpenAPI 規格檔案：透過拖放上傳 OpenAPI 規格 JSON 檔案，或選擇選取檔案以導覽本機檔案系統，然後選擇要上傳的檔案。

新增後，系統會處理 檔案，並顯示可用選項的清單。選取連接器的必要操作。

13. 選擇建立。新建立的連接器會出現在連接器清單中。

現在連接器已建立，建置器可以在其應用程式中使用它。

連線至 Salesforce

若要將 App Studio 與 Salesforce 連線，讓建置器能夠在應用程式中存取和使用 Salesforce 資源，您必須在 Salesforce 中建立和設定連線的應用程式，並在 App Studio 中建立 Salesforce 連接器。

將 Salesforce 與 App Studio 連線

1. 在 App Studio 的導覽窗格中，選擇管理區段中的連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的一些詳細資訊。
2. 選擇 + 建立連接器。
3. 從連接器類型清單中選擇 Salesforce，以開啟連接器建立頁面。
4. 請記下重新導向 URL，您將在下列步驟中使用此 URL 來設定 Salesforce。
5. 下一個步驟是在 Salesforce 中建立連線的應用程式。在另一個索引標籤或視窗中，導覽至您的 Salesforce 執行個體。
6. 在快速尋找方塊中，搜尋 **App Manager** 並選取 App Manager。
7. 選擇新的連線應用程式。

8. 在連線的應用程式名稱和 API 名稱中，輸入應用程式的名稱。它不必與您的 App Studio 應用程式名稱相符。
9. 視需要提供聯絡資訊。
10. 在 API（啟用 OAuth 設定）區段中，啟用啟用 OAuth 設定。
11. 在回呼 URL 中，輸入您先前從 App Studio 記下的重新導向 URL。
12. 在選取的 OAuth 範圍中，從清單中新增必要的許可範圍。App Studio 可以與 Salesforce REST APIs 互動，對五個物件執行 CRUD 操作：帳戶、案例、聯絡人、潛在客戶和機會。建議新增完整存取（完整），以確保您的 App Studio 應用程式具有所有相關許可或範圍。
13. 停用支援授權流程的程式碼交換 (PKCE) 延伸需要證明金鑰選項。App Studio 不支援 PKCE。
14. 啟用需要秘密的 Web 伺服器流程和需要秘密的重新整理字符流程，以遵循最佳安全實務。
15. App Studio 支援下列兩種身分驗證流程：
 - 用戶端登入資料流程：非常適合 server-to-server 互動，其中應用程式在沒有使用者互動的情況下代表自己運作。例如，列出沒有 Salesforce 存取權的臨時員工團隊的所有潛在客戶資訊。
 - 授權碼流程：適用於代表使用者執行動作的應用程式，例如個人資料存取或動作。例如，列出每個銷售經理的潛在客戶來源或由他們擁有，以透過此應用程式執行其他任務。
 - 對於用戶端登入資料流程：
 - a. 啟用啟用用戶端登入資料流程。檢閱並確認訊息。
 - b. 儲存應用程式。
 - c. 您必須選取執行使用者，雖然流程中沒有使用者互動。透過選取執行使用者，Salesforce 會代表使用者傳回存取權杖。
 1. 在 App Manager 中，從應用程式清單中選擇 App Studio 應用程式的箭頭，然後選擇管理。
 2. 選擇編輯政策
 3. 在用戶端登入資料流程中，新增適當的使用者。
 - 針對授權碼流程，啟用啟用授權碼和登入資料流程
16. Salesforce 提供用戶端 ID 和用戶端秘密，必須在下列步驟中用來在 App Studio 中設定連接器。
 - a. 在 App Manager 中，選擇 App Studio 應用程式的箭頭，然後選擇檢視。
 - b. 在 API（啟用 OAuth 設定）區段中，選擇管理消費者詳細資訊。這可能會傳送驗證金鑰的電子郵件，您需要輸入此金鑰以進行確認。

- c. 請注意消費者金鑰 (用戶端 ID) 和消費者秘密 (用戶端秘密)。
17. 返回 App Studio，填寫下列欄位來設定和建立您的連接器。
18. 在名稱中，輸入 Salesforce 連接器的名稱。
19. 在描述中，輸入 Salesforce 連接器的描述。
20. 在基本 URL 中，輸入 Salesforce 執行個體的基本 URL。看起來應該像這樣：`https://hostname.salesforce.com/services/data/v60.0`，將####取代為您的 Salesforce 執行個體名稱。
21. 在身分驗證方法中，確保已選取 OAuth 2.0。
22. 在 OAuth 2.0 流程中，選取 OAuth 身分驗證方法並填寫相關欄位：
 - 針對system-to-system整合，選取用戶端登入資料流程以用於代表自己的應用程式。
 - a. 在用戶端 ID 中，輸入先前從 Salesforce 取得的取用者金鑰。
 - b. 在用戶端秘密中，輸入先前從 Salesforce 取得的消費者秘密。
 - c. 在存取字符 URL 中，輸入 OAuth 2.0 字符端點。看起來應該像這樣：`https://hostname/services/oauth2/token`，將####取代為您的 Salesforce 執行個體名稱。如需詳細資訊，請參閱 [Salesforce OAuth 端點](#) 文件。
 - d. 選擇驗證連線以測試身分驗證和連線。
 - 選取授權碼流程，以用於代表使用者執行動作的應用程式。
 - a. 在用戶端 ID 中，輸入先前從 Salesforce 取得的取用者金鑰。
 - b. 在用戶端秘密中，輸入先前從 Salesforce 取得的消費者秘密。
 - c. 在授權 URL 中，輸入授權端點。看起來應該像這樣：`https://hostname/services/oauth2/authorize`，將####取代為您的 Salesforce 執行個體名稱。如需詳細資訊，請參閱 [Salesforce OAuth 端點](#) 文件。
 - d. 在存取字符 URL 中，輸入 OAuth 2.0 字符端點。看起來應該像這樣：`https://hostname/services/oauth2/token`，將####取代為您的 Salesforce 執行個體名稱。如需詳細資訊，請參閱 [Salesforce OAuth 端點](#) 文件。
23. 在操作中，選取連接器將支援的 Salesforce 操作。此清單中的操作是預先定義的，並代表 Salesforce 中的常見任務，例如從常見物件建立、擷取、更新或刪除記錄。
24. 選擇建立。新建立的連接器會出現在連接器清單中。

檢視、編輯和刪除連接器

若要檢視、編輯或刪除現有的連接器

1. 在導覽窗格中，在管理區段中選擇連接器。您將前往顯示現有連接器清單的頁面，其中包含每個連接器的下列詳細資訊：
 - 名稱：建立期間提供的連接器名稱。
 - 描述：建立期間提供的連接器描述。
 - 連接至：連接器連接至 App Studio 的服務。API 的值代表與第三方服務的連線。
 - 建立者：建立連接器的使用者。
 - 建立日期：連接器建立的日期。
2. 若要檢視連接器的詳細資訊，或編輯或刪除連接器，請使用下列指示：
 - 若要查看特定連接器的詳細資訊，請選擇該連接器的檢視。
 - 若要編輯連接器，請選擇檢視旁的下拉式選單，然後選擇編輯。
 - 若要刪除連接器，請選擇檢視旁的下拉式選單，然後選擇刪除。

刪除 App Studio 執行個體

使用本主題中的程序來刪除您的 App Studio 執行個體。如果您在其他服務中建立資源以搭配 App Studio 使用，請視需要檢閱並刪除這些資源，以避免產生費用。

您可能因為下列原因而想要刪除 App Studio 執行個體：

- 您不想再使用 App Studio。
- 您想要在不同區域中建立 App Studio 執行個體 AWS。由於 App Studio 一次僅支援在一個區域中擁有執行個體，因此您必須刪除任何現有的執行個體，才能建立另一個執行個體。

Warning

刪除 App Studio 執行個體也會刪除所有 App Studio 資源，例如應用程式和連接器。刪除執行個體無法復原。

刪除您的 App Studio 執行個體

1. 在 <https://console.aws.amazon.com/appstudio/> 開啟 App Studio 主控台。
2. 選取您 App Studio 執行個體所在的區域。
3. 在導覽窗格中，選擇執行個體。
4. 選擇動作以開啟具有其他執行個體動作的下拉式清單。
5. 選擇刪除 App Studio 執行個體。
6. 輸入 **confirm**，然後選擇 Delete (刪除)。
7. 您的執行個體刪除可能需要一段時間才能處理。刪除後，您會收到確認電子郵件。收到電子郵件後，您可以視需要建立另一個執行個體。

建置器文件

下列主題包含的資訊可協助 App Studio 中建立、編輯和發佈應用程式的使用者。

主題

- [教學](#)
- [使用生成式 AI 建置您的 App Studio 應用程式](#)
- [建立、編輯和刪除應用程式](#)
- [預覽、發佈和共用應用程式](#)
- [頁面和元件：建置應用程式的使用者介面](#)
- [自動化和動作：定義應用程式的商業邏輯](#)
- [實體和資料動作：設定應用程式的資料模型](#)
- [頁面和自動化參數](#)
- [使用 JavaScript 在 App Studio 中撰寫表達式](#)
- [資料相依性和時間考量](#)
- [使用多個使用者建置應用程式](#)
- [檢視或更新應用程式的內容安全設定](#)

教學

主題

- [使用 Amazon Bedrock 建置 AI 文字摘要器應用程式](#)
- [與 Amazon Simple Storage Service 與元件和自動化互動](#)
- [在 App Studio 應用程式中叫用 Lambda 函數](#)

使用 Amazon Bedrock 建置 AI 文字摘要器應用程式

在本教學課程中，您將在 App Studio 中建置應用程式，該應用程式使用 Amazon Bedrock 提供最終使用者輸入文字的簡潔摘要。應用程式包含一個簡單的使用者介面，使用者可以輸入他們想要摘要的任何文字。這可能是會議備註、文章內容、調查結果或任何其他文字資訊。使用者輸入文字後，可以按下按鈕將文字傳送至 Amazon Bedrock，這會使用 Claude 3 Sonnet 模型處理文字並傳回摘要版本。

內容

- [先決條件](#)
- [步驟 1：建立和設定 IAM 角色和 App Studio 連接器](#)
- [步驟 2：建立應用程式](#)
- [步驟 3：建立和設定自動化](#)
- [步驟 4：建立頁面和元件](#)
 - [重新命名預設頁面](#)
 - [將元件新增至頁面](#)
 - [設定頁面元件](#)
- [步驟 5：將應用程式發佈至測試環境](#)
- [\(選用\) 清除](#)

先決條件

開始之前，請先檢閱並完成下列先決條件：

- 存取 AWS App Studio。請注意，您必須擁有管理員角色，才能在本教學課程中建立連接器。
- 選用：檢閱 [AWS App Studio 概念](#) 和 [教學課程：從空白應用程式開始建置](#) 以熟悉重要的 App Studio 概念。

步驟 1：建立和設定 IAM 角色和 App Studio 連接器

若要提供 App Studio 對 Amazon Bedrock 模型的存取權，您必須：

1. 啟用您要在應用程式中使用的 Amazon Bedrock 模型。在本教學課程中，您將使用 Claude 3 Sonnet，因此請務必啟用該模型。
2. 建立具有 Amazon Bedrock 適當許可的 IAM 角色。
3. 使用將在應用程式中使用的 IAM 角色建立 App Studio 連接器。

前往 [連線至 Amazon Bedrock](#) 以取得詳細說明，並在遵循步驟並建立連接器後返回本教學課程。

步驟 2：建立應用程式

使用下列程序在 App Studio 中建立空應用程式，您將建置到文字摘要器應用程式中。

1. 登入 App Studio。

2. 導覽至建置器中樞，然後選擇 + 建立應用程式。
3. 選擇從頭開始。
4. 在應用程式名稱欄位中，提供應用程式的名稱，例如 **Text Summarizer**。
5. 如果您被要求選取資料來源或連接器，請選擇略過以進行本教學課程。
6. 選擇下一步繼續進行。
7. (選用)：觀看影片教學課程，以取得在 App Studio 中建置應用程式的快速概觀。
8. 選擇編輯應用程式，這將帶您進入應用程式工作室。

步驟 3：建立和設定自動化

您可以在自動化中定義 App Studio 應用程式的邏輯和行為。自動化包含稱為動作的個別步驟、用於將資料從其他資源傳遞至動作的參數，以及可供其他自動化或元件使用的輸出。在此步驟中，您將建立自動化，以使用下列項目處理與 Amazon Bedrock 的互動：

- 輸入：將文字輸入從使用者傳遞至自動化的參數。
- 動作：一個將文字輸入傳送至 Amazon Bedrock 並傳回輸出文字摘要的 GenAI 提示動作。
- 輸出：自動化輸出，其中包含來自 Amazon Bedrock 的已處理摘要，可用於您的應用程式。

建立和設定自動化，將提示傳送至 Amazon Bedrock 並處理和傳回摘要

1. 選擇畫布頂端的自動化索引標籤。
2. 選擇 + 新增自動化。
3. 在右側面板中，選擇屬性。
4. 選擇鉛筆圖示來更新自動化名稱。輸入 **InvokeBedrock**，然後按 Enter。
5. 將參數新增至自動化，以透過執行下列步驟，將文字提示輸入從使用者傳遞至要在請求中用於 Amazon Bedrock 的自動化：
 - a. 在畫布的參數方塊中，選擇 + 新增。
 - b. 在名稱中，輸入 **input**。
 - c. 在描述中，輸入任何描述，例如 **Text to be sent to Amazon Bedrock**。
 - d. 在類型中，選取字串。
 - e. 選擇新增以建立參數。
6. 執行下列步驟來新增 GenAI 提示動作：

- a. 在右側面板中，選擇動作。
 - b. 選擇 GenAI 提示以新增動作。
7. 透過執行下列步驟來設定動作：
- a. 從畫布中選擇動作以開啟右側屬性功能表。
 - b. 選擇鉛筆圖示、輸入名稱，然後按下 Enter 鍵，將動作重新命名 **PromptBedrock** 為。
 - c. 在連接器中，選取在 中建立的連接器 [步驟 1：建立和設定 IAM 角色和 App Studio 連接器](#)。
 - d. 在模型中，選擇您要用來處理提示的 Amazon Bedrock 模型。在本教學課程中，您將選擇 Claude 3.5 Sonnet。
 - e. 在使用者提示中，輸入 `{{params.input}}`。這代表您先前建立的 input 參數，並將包含應用程式使用者輸入的文字。
 - f. 在系統提示中，輸入您要傳送至 Amazon Bedrock 的系統提示指示。在本教學課程中，請輸入下列項目：

```
You are a highly efficient text summarizer. Provide a concise summary of the prompted text, capturing the key points and main ideas.
```
 - g. 選擇請求設定以展開，並更新下列欄位：
 - 在溫度中，輸入 0。範本會在 0 到 10 的範圍內決定輸出的隨機性或創造性。數字越高，回應的創造性就越高。
 - 在權杖上限中，輸入 4096 以限制回應的長度。
8. 此自動化的輸出將是摘要文字，但預設自動化不會建立輸出。執行下列步驟，設定自動化以建立自動化輸出：
- a. 在左側導覽中，選擇 InvokeBedrock 自動化。
 - b. 在右側屬性選單的輸出中，選擇 + 新增。
 - c. 在輸出中，輸入 `{{results.PromptBedrock.text}}`。此表達式會傳回 `processResults` 動作的內容。

步驟 4：建立頁面和元件

在 App Studio 中，每個頁面代表使用者將互動的應用程式使用者介面 (UI) 畫面。在這些頁面中，您可以新增各種元件，例如資料表、表單、按鈕等，以建立所需的配置和功能。

重新命名預設頁面

本教學課程中的文字摘要程式應用程式只會包含一個頁面。新建立的應用程式隨附預設頁面，因此您將重新命名該應用程式，而不是新增應用程式。

重新命名預設頁面

1. 在頂端列導覽選單中，選擇頁面。
2. 在左側面板中，選擇Page1，然後選擇右側面板中的屬性面板。
3. 選擇鉛筆圖示，輸入 **TextSummarizationTool**，然後按 Enter。
4. 在導覽標籤中，輸入 **TextSummarizationTool**。

將元件新增至頁面

在本教學課程中，文字摘要器應用程式有一個頁面，其中包含下列元件：

- 最終使用者用來輸入要摘要提示的文字輸入元件。
- 用來將提示傳送至 Amazon Bedrock 的按鈕元件。
- 文字區域元件，顯示來自 Amazon Bedrock 的摘要。

將文字輸入元件新增至使用者用來輸入要摘要之文字提示的頁面。

新增文字輸入元件

1. 在右側元件面板中，找到文字輸入元件並將其拖曳到畫布上。
2. 選擇畫布中的文字輸入以進行選取。
3. 在右側屬性面板中，更新下列設定：
 - a. 選擇鉛筆圖示，將文字輸入重新命名為 **inputPrompt**。
 - b. 在標籤中，輸入 **Prompt**。
 - c. 在預留位置中，輸入 **Enter text to be summarized**。

現在，新增使用者將選擇將提示傳送至 Amazon Bedrock 的 Button 元件。

新增按鈕元件

1. 在右側元件面板中，找到按鈕元件並將其拖曳到畫布上。

2. 選擇畫布中的按鈕來選取它。
3. 在右側屬性面板中，更新下列設定：
 - a. 選擇鉛筆圖示，將按鈕重新命名為 **sendButton**。
 - b. 在按鈕標籤中，輸入 **Send**。

現在，新增文字區域元件，以顯示 Amazon Bedrock 傳回的摘要。

新增文字區域元件

1. 在右側元件面板中，找到文字區域元件並將其拖曳到畫布上。
2. 選擇畫布中的文字區域以進行選取。
3. 在右側屬性面板中，更新下列設定：
 - a. 選擇鉛筆圖示，將按鈕重新命名為 **textSummary**。
 - b. 在標籤中，輸入 **Summary**。

設定頁面元件

現在應用程式包含含有元件的頁面，下一個步驟是設定元件以執行適當的行為。若要設定元件，例如按鈕，以在與元件互動時採取動作，您必須新增觸發。對於本教學課程中的應用程式，您將在 `sendButton` 按鈕中新增兩個觸發條件，以執行下列動作：

- 第一個觸發會將 `textPrompt` 元件中的文字傳送至要分析的 Amazon Bedrock。
- 第二個觸發會在 `textSummary` 元件中顯示從 Amazon Bedrock 傳回的摘要。

新增將提示傳送至 Amazon Bedrock 的觸發條件

1. 選擇畫布中的按鈕來選取它。
2. 在右側屬性面板的觸發區段中，選擇 + 新增。
3. 選擇調用自動化。
4. 選擇為設定它而建立的 `InvokeAutomation1` 觸發條件。
5. 在動作名稱中，輸入 **invokeBedrockAutomation**。
6. 在調用自動化中，選取先前建立的 `InvokeBedrock` 自動化。

7. 在參數方塊中，在先前建立的輸入參數中，輸入 `{{ui.inputPrompt.value}}`，傳遞 `inputPrompt` 文字輸入元件中的內容。
8. 選擇面板頂端的向左箭頭，以返回元件屬性選單。

現在，您已設定觸發程序，以叫用自動化，在按下按鈕時將請求傳送至 Amazon Bedrock，下一個步驟是設定第二個觸發程序，在 `textSummary` 元件中顯示結果。

新增在文字區域元件中顯示 Amazon Bedrock 結果的觸發條件

1. 在按鈕的右側屬性面板的觸發區段中，選擇 + 新增。
2. 選擇執行元件動作。
3. 選擇為設定而建立的 `Runcomponentaction1` 觸發條件。
4. 在動作名稱中，輸入 `setTextSummary`。
5. 在元件中，選取 `textSummary` 元件。
6. 在動作中，選取設定值。
7. 在將值設定為 中，輸入 `{{results.invokeBedrockAutomation}}`。

步驟 5：將應用程式發佈至測試環境

一般而言，當您建置應用程式時，最好預覽應用程式，以查看其外觀並對其功能進行初始測試。不過，由於應用程式不會與預覽環境中的外部服務互動，因此您將改為將應用程式發佈至測試環境，以測試傳送請求並從 Amazon Bedrock 接收回應。

將您的應用程式發佈至測試環境

1. 在應用程式建置器的右上角，選擇發佈。
2. 新增測試環境的版本描述。
3. 檢閱並選取有關 SLA 的核取方塊。
4. 選擇 開始使用。發佈最多可能需要 15 分鐘。
5. (選用) 當您準備好時，您可以選擇共用並依照提示給予其他人存取權。如需共用 App Studio 應用程式的詳細資訊，請參閱 [共用已發佈的應用程式](#)。

測試應用程式後，再次選擇發佈，將應用程式提升為生產環境。請注意，在共用最終使用者之前，生產環境中的應用程式無法使用。如需不同應用程式環境的詳細資訊，請參閱 [應用程式環境](#)。

(選用) 清除

您現在已成功完成教學課程，並使用 Amazon Bedrock 在 App Studio 中建置文字摘要應用程式。您可以繼續使用您的應用程式，也可以清除在本教學課程中建立的資源。下列清單包含要清除的資源清單：

- 在 App Studio 中建立的 Amazon Bedrock 連接器。如需詳細資訊，請參閱[檢視、編輯和刪除連接器](#)。
- App Studio 中的文字摘要程式應用程式。如需詳細資訊，請參閱[刪除應用程式](#)。
- 在 IAM 主控台中建立的 IAM 角色。如需詳細資訊，請參閱 AWS Identity and Access Management 《使用者指南》中的[刪除角色或執行個體描述檔](#)。
- 如果您請求模型存取以使用 Claude 3 Sonnet 並想要還原存取，請參閱《[Amazon Bedrock 使用者指南](#)》中的[管理對 Amazon Bedrock 基礎模型的存取](#)。

與 Amazon Simple Storage Service 與元件和自動化互動

您可以從 App Studio 應用程式叫用各種 Amazon S3 操作。例如，您可以建立簡單的管理員面板來管理您的使用者和訂單，並顯示來自 Amazon S3 的媒體。雖然您可以使用叫用 AWS 動作叫用任何 Amazon S3 操作，但您可以新增四個專用 Amazon S3 動作至應用程式中的自動化，以在 Amazon S3 儲存貯體和物件上執行常見操作。四個動作及其操作如下所示：

- Put Object：使用 Amazon S3 PutObject 操作來新增 Amazon S3 儲存貯體的物件。
- 取得物件：使用 Amazon S3 GetObject 操作從 Amazon S3 儲存貯體擷取物件。
- 列出物件：使用 Amazon S3 ListObjects 操作列出 Amazon S3 儲存貯體中的物件。
- 刪除物件：使用 Amazon S3 DeleteObject 操作從 Amazon S3 儲存貯體刪除物件。

除了動作之外，您還可以將 S3 上傳元件新增至應用程式中的頁面。使用者可以使用此元件選擇要上傳的檔案，而元件會呼叫 Amazon S3 PutObject 將檔案上傳至設定的儲存貯體和資料夾。本教學課程將使用此元件取代獨立的 Put Object 自動化動作。（獨立動作應在更複雜的案例中使用，這些案例涉及上傳之前或之後要採取的其他邏輯或動作。）

先決條件

本指南假設您已完成下列先決條件：

1. 建立並設定 Amazon S3 儲存貯體、IAM 角色和政策，以及 Amazon S3 連接器，以成功整合 Amazon S3 與 App Studio。若要建立連接器，您必須具有管理員角色。如需詳細資訊，請參閱[連線至 Amazon Simple Storage Service \(Amazon S3\)](#)。

建立空白應用程式

透過執行下列步驟，建立要在本指南中使用的空白應用程式。

建立空白應用程式

1. 在導覽窗格中，選擇我的應用程式。
2. 選擇 + 建立應用程式。
3. 在建立應用程式對話方塊中，為您的應用程式命名，選擇從頭開始，然後選擇下一步。
4. 在連線至現有資料對話方塊中，選擇略過以建立應用程式。
5. 選擇編輯要帶到新應用程式的畫布，您可以在其中使用元件、自動化和資料來設定應用程式的外觀和功能。

建立頁面

在應用程式中建立三個頁面以收集或顯示資訊。

建立頁面

1. 如有必要，請選擇畫布頂端的分頁。
2. 在左側導覽中，您的應用程式已建立單一頁面。選擇 + 新增兩次以建立另外兩個頁面。導覽窗格應會顯示總共三個頁面。
3. 執行下列步驟來更新 Page1 頁面的名稱：
 - a. 選擇省略號圖示，然後選擇頁面屬性。
 - b. 在右側屬性功能表中，選擇鉛筆圖示以編輯名稱。
 - c. 輸入 **FileList**，然後按 Enter。
4. 重複上述步驟以更新第二頁和第三頁，如下所示：
 - 將Page2 重新命名為 **UploadFile**。
 - 將 Page3 重新命名為 **FailUpload**。

現在，您的應用程式應該有三個頁面，名為 FileList、UploadFile 和 FailUpload，顯示在左側頁面面板中。

接著，您將建立和設定與 Amazon S3 互動的自動化。

建立和設定自動化

建立與 Amazon S3 互動的應用程式自動化。使用下列程序建立下列自動化：

- 列出 Amazon S3 儲存貯體中物件的 `getFiles` 自動化，用於填充資料表元件。
- `deleteFile` 自動化會從 Amazon S3 儲存貯體刪除物件，其將用於將刪除按鈕新增至資料表元件。
- `viewFile` 自動化會從 Amazon S3 儲存貯體取得物件並顯示它，用於顯示從資料表元件中選取之單一物件的更多詳細資訊。

建立 `getFiles` 自動化

建立自動化，列出指定 Amazon S3 儲存貯體中的檔案。

1. 選擇畫布頂端的自動化索引標籤。
2. 選擇 + 新增自動化。
3. 在右側面板中，選擇屬性。
4. 選擇鉛筆圖示來更新自動化名稱。輸入 **getFiles**，然後按 Enter。
5. 透過執行下列步驟來新增清單物件動作：
 - a. 在右側面板中，選擇動作。
 - b. 選擇列出物件以新增動作。動作應該命名為 `ListObjects1`。
6. 透過執行下列步驟來設定動作：
 - a. 從畫布中選擇動作以開啟右側屬性功能表。
 - b. 針對連接器，選擇您從先決條件建立的 Amazon S3 連接器。
 - c. 針對組態，輸入下列文字，將 *bucket_name* 取代為您先決條件中建立的儲存貯體：

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

Note

您可以使用 Prefix 欄位，將回應限制為以指定字串開頭的物件。

7. 此自動化的輸出將用於將 Amazon S3 儲存貯體中的物件填入資料表元件。不過，根據預設，自動化不會建立輸出。執行下列步驟，設定自動化以建立自動化輸出：
 - a. 在左側導覽中，選擇 `getFiles` 自動化。
 - b. 在右側屬性功能表的自動化輸出中，選擇 + 新增輸出。
 - c. 針對輸出，輸入 `{{results.ListObjects1.Contents}}`。此表達式會傳回 動作的內容，現在可以用來填入資料表元件。

建立 `deleteFile` 自動化

建立從指定的 Amazon S3 儲存貯體刪除物件的自動化。

1. 在左側自動化面板中，選擇 + 新增。
2. 選擇 + 新增自動化。
3. 在右側面板中，選擇屬性。
4. 選擇鉛筆圖示來更新自動化名稱。輸入 `deleteFile`，然後按 Enter。
5. 透過執行下列步驟，新增用來將資料傳遞至自動化的自動化參數：
 - a. 在屬性功能表右側，在自動化參數中，選擇 + 新增。
 - b. 選擇鉛筆圖示以編輯自動化參數。將參數名稱更新為 `fileName`，然後按 Enter 鍵。
6. 透過執行下列步驟來新增刪除物件動作：
 - a. 在右側面板中，選擇動作。
 - b. 選擇刪除物件以新增動作。動作應該命名為 `DeleteObject1`。
7. 透過執行下列步驟來設定動作：
 - a. 從畫布中選擇動作以開啟右側屬性功能表。
 - b. 針對連接器，選擇您從先決條件建立的 Amazon S3 連接器。
 - c. 針對組態，輸入下列文字，將 `bucket_name` 取代為您在先決條件中建立的儲存貯體：

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

建立viewFile自動化

建立從指定的 Amazon S3 儲存貯體擷取單一物件的自動化。稍後，您將使用檔案檢視器元件來設定此自動化，以顯示物件。

1. 在左側自動化面板中，選擇 + 新增。
2. 選擇 + 新增自動化。
3. 在右側面板中，選擇屬性。
4. 選擇鉛筆圖示來更新自動化名稱。輸入 **viewFile**，然後按 Enter。
5. 透過執行下列步驟，新增用來將資料傳遞至自動化的自動化參數：
 - a. 在屬性功能表右側，在自動化參數中，選擇 + 新增。
 - b. 選擇鉛筆圖示以編輯自動化參數。將參數名稱更新為 **fileName**，然後按 Enter 鍵。
6. 透過執行下列步驟來新增取得物件動作：
 - a. 在右側面板中，選擇動作。
 - b. 選擇取得物件以新增動作。動作應該命名為 GetObject1。
7. 透過執行下列步驟來設定動作：
 - a. 從畫布中選擇動作以開啟右側屬性功能表。
 - b. 針對連接器，選擇您從先決條件建立的 Amazon S3 連接器。
 - c. 針對組態，輸入下列文字，將 *bucket_name* 取代為您先決條件中建立的儲存貯體：

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

8. 根據預設，自動化不會建立輸出。執行下列步驟，設定自動化以建立自動化輸出：
 - a. 在左側導覽中，選擇 viewFile 自動化。
 - b. 在右側屬性功能表的自動化輸出中，選擇 + 新增輸出。
 - c. 針對輸出，輸入 **{{results.GetObject1.Body.transformToWebStream()}}**。此表達式會傳回動作的內容。

Note

您可以透過 S3 GetObject 下列方式讀取 的回應：

- `transformToWebStream`：傳回必須取用的串流，才能擷取資料。如果用作自動化輸出，自動化會處理此問題，而輸出可以用作影像或 PDF 檢視器元件的資料來源。它也可以用作另一個操作的輸入，例如 `S3 PutObject`。
- `transformToString`：傳回自動化的原始資料，如果您的檔案包含文字內容，例如 JSON 資料，則應用於 JavaScript 動作。必須等待，例如：`await results.GetObject1.Body.transformToString();`
- `transformToByteArray`：傳回 8 位元未簽署整數的陣列。此回應提供位元組陣列的目的，允許儲存和處理二進位資料。必須等待，例如：`await results.GetObject1.Body.transformToByteArray();`

接下來，您將新增元件到您先前建立的頁面，並使用自動化進行設定，以便使用者可以使用您的應用程式來檢視和刪除檔案。

新增和設定頁面元件

現在您已建立定義應用程式商業邏輯和功能的自動化，您將建立元件並連接兩者。

將元件新增至 FileList 頁面

您先前建立的 FileList 頁面將用於顯示已設定 Amazon S3 儲存貯體中的檔案清單，以及從清單中選擇之任何檔案的詳細資訊。若要這麼做，您將執行下列動作：

1. 建立資料表元件以顯示檔案清單。您將設定資料表的資料列，以填入您先前建立的 `getFiles` 自動化輸出。
2. 建立 PDF 檢視器元件以顯示單一 PDF。您將設定元件，使用您先前建立的 `viewFile` 自動化來從儲存貯體擷取檔案，以檢視從資料表中選取的檔案。

將元件新增至 FileList 頁面

1. 選擇畫布頂端的分頁。
2. 在左側頁面面板中，選擇 FileList 頁面。
3. 在右側元件頁面上，尋找資料表元件，並將其拖曳至畫布的中心。
4. 選擇您剛新增至頁面的資料表元件。
5. 在右側屬性功能表中，選擇來源下拉式清單，然後選取自動化。

6. 選擇自動化下拉式清單，然後選取 `getFiles` 自動化。資料表將使用 `getFiles` 自動化的輸出做為其內容。
7. 新增要填入檔案名稱的資料欄。
 - a. 在右側屬性功能表的資料欄旁，選擇 + 新增。
 - b. 選擇剛新增之 `Column1` 資料欄右側的箭頭圖示。
 - c. 對於資料欄標籤，將資料欄重新命名為 **Filename**。
 - d. 針對數值，輸入 `{{currentRow.Key}}`。
 - e. 選擇面板頂端的箭頭圖示，以返回主要屬性面板。
8. 新增資料表動作以刪除資料列中的檔案。
 - a. 在屬性功能表右側的動作旁，選擇 + 新增。
 - b. 在動作中，將按鈕重新命名為 **Delete**。
 - c. 選擇剛重新命名的刪除動作右側的箭頭圖示。
 - d. 在按一下時，選擇 + 新增動作，然後選擇叫用自動化。
 - e. 選擇新增的動作以進行設定。
 - f. 針對動作名稱，輸入 **DeleteRecord**。
 - g. 在叫用自動化中，選取 `deleteFile`。
 - h. 在參數文字方塊中，輸入 `{{currentRow.Key}}`。
 - i. 針對數值，輸入 `{{currentRow.Key}}`。
9. 在右側面板中，選擇元件以檢視元件選單。有兩種顯示檔案的選項：
 - 影像檢視器，可檢視具有 `.png`、`.jpeg` 或 `.jpg` 副檔名的檔案。
 - 用於檢視 PDF 檔案的 PDF 檢視器元件。

在本教學課程中，您將新增和設定 PDF 檢視器元件。

10. 新增 PDF 檢視器元件。
 - a. 在右側元件頁面上，尋找 PDF 檢視器元件，並將其拖曳至資料表元件下方的畫布。
 - b. 選擇剛新增的 PDF 檢視器元件。
 - c. 在右側屬性功能表中，選擇來源下拉式清單，然後選取自動化。
 - d. 選擇自動化下拉式清單，然後選取 `viewFile` 自動化。資料表將使用 `viewFile` 自動化的輸出做為其內容。

- e. 在參數文字方塊中，輸入 `{{ui.table1.selectedRow["Filename"]}}`。
- f. 在右側面板中，也有檔案名稱欄位。此欄位的值會用作 PDF 檢視器元件的標頭。輸入與上一個步驟相同的文字：`{{ui.table1.selectedRow["Filename"]}}`。

將元件新增至 UploadFile 頁面

UploadFile 頁面將包含檔案選取器，可用於選取檔案並將其上傳至設定的 Amazon S3 儲存貯體。您將新增 S3 上傳元件至頁面，使用者可以使用該頁面來選取和上傳檔案。

1. 在左側頁面面板中，選擇 UploadFile 頁面。
2. 在右側元件頁面上，尋找 S3 上傳元件，並將其拖曳至畫布的中心。
3. 選擇您剛新增至頁面的 S3 上傳元件。
4. 在屬性功能表右側，設定元件：
 - a. 在連接器下拉式清單中，選取在先決條件中建立的 Amazon S3 連接器。
 - b. 在儲存貯體中，輸入 Amazon S3 儲存貯體的名稱。
 - c. 針對檔案名稱，輸入 `{{ui.s3Upload1.files[0]?.nameWithExtension}}`。
 - d. 對於檔案大小上限，請在文字方塊中輸入 `5`，並確保已在下拉式清單中選取 **MB**。
 - e. 在觸發區段中，透過執行下列步驟，新增成功或失敗上傳後執行的動作：

若要新增成功上傳後執行的動作：

1. 在成功時，選擇 + 新增動作，然後選取導覽。
2. 選擇新增的動作以進行設定。
3. 對於導覽類型，選擇頁面。
4. 針對導覽至，選擇 **FileList**。
5. 選擇面板頂端的箭頭圖示，以返回主要屬性面板。

若要新增在上傳失敗後執行的動作：

1. 在失敗時，選擇 + 新增動作，然後選取導覽。
2. 選擇新增的動作以進行設定。
3. 對於導覽類型，選擇頁面。
4. 針對導覽至，選擇 **FailUpload**。

5. 選擇面板頂端的箭頭圖示，以返回主要屬性面板。

將元件新增至 FailUpload 頁面

FailUpload 頁面是簡單的頁面，其中包含文字方塊，可通知使用者其上傳失敗。

1. 在左側頁面面板中，選擇 FailUpload 頁面。
2. 在右側元件頁面上，尋找文字元件，並將其拖曳至畫布的中心。
3. 選擇您剛新增至頁面的文字元件。
4. 在屬性功能表右側的值中，輸入 **Failed to upload, try again.**

更新您的應用程式安全設定

App Studio 中的每個應用程式都有內容安全設定，可用來限制外部媒體或資源，或您可以上傳物件至 Amazon S3 中的哪些網域。預設設定為封鎖所有網域。若要從應用程式上傳物件至 Amazon S3，您必須更新設定，以允許您要上傳物件的網域。

允許網域將物件上傳至 Amazon S3

1. 選擇應用程式設定索引標籤。
2. 選擇內容安全設定索引標籤。
3. 針對 Connect 來源，選擇允許所有連線。
4. 選擇儲存。

後續步驟：預覽和發佈要測試的應用程式

應用程式現在已準備好進行測試。如需預覽和發佈應用程式的詳細資訊，請參閱[預覽、發佈和共用應用程式](#)。

在 App Studio 應用程式中叫用 Lambda 函數

本教學課程說明如何將 App Studio 連線至 Lambda，並從應用程式叫用 Lambda 函數。

先決條件

本指南假設您已完成下列先決條件：

1. 已建立 App Studio 應用程式。如果您沒有應用程式，您可以建立要在教學課程中使用的空白應用程式。如需詳細資訊，請參閱[建立應用程式](#)。

Note

雖然您不需要 Lambda 函數來遵循本教學課程並了解如何進行設定，但擁有一個可能有助於確保您已正確設定應用程式。本教學課程不包含建立 Lambda 函數的相關資訊。如需詳細資訊，請參閱[AWS Lambda 開發人員指南](#)。

建立 Lambda 連接器

若要在 App Studio 應用程式中使用 Lambda 函數，您必須使用連接器將 App Studio 連線至 Lambda，以提供函數的存取權。您必須是管理員，才能在 App Studio 中建立連接器。如需建立 Lambda 連接器的詳細資訊，包括建立連接器的步驟，請參閱[連線至 AWS Lambda](#)。

建立和設定自動化

自動化用於定義應用程式的邏輯，並由動作組成。若要在應用程式中叫用 Lambda 函數，請先將叫用 Lambda 動作新增至自動化。使用下列步驟來建立自動化，並將調用 Lambda 動作新增至自動化。

1. 編輯應用程式時，請選擇自動化索引標籤。
2. 選擇 + 新增自動化。
3. 在右側動作選單中，選擇叫用 Lambda 將步驟新增至自動化。
4. 在畫布中選擇新的 Lambda 步驟，以檢視和設定其屬性。
5. 在右側屬性功能表中，執行下列步驟來設定步驟：
 - a. 在連接器中，選取為將 App Studio 連線至 Lambda 函數而建立的連接器。
 - b. 在函數名稱中，輸入 Lambda 函數的名稱。
 - c. 在函數事件中，輸入要傳遞給 Lambda 函數的事件。下列清單提供一些常見的使用案例範例：
 - 傳遞自動化參數的值，例如檔案名稱或其他字串：`varName: params.paramName`
 - 傳遞先前動作的結果：`varName: results.actionName1.data[0].fieldName`
 - 如果您在迴圈動作內新增調用 Lambda 動作，您可以從每個重複項目傳送類似於參數的欄位：`varName: currentItem.fieldName`

- d. 模擬輸出欄位可用於提供模擬輸出，以在預覽時測試應用程式，其中連接器未處於作用中狀態。

設定 UI 元素以執行自動化

現在您的自動化已設定動作來叫用 Lambda 函數，您可以設定 UI 元素來執行自動化。在本教學課程中，您將建立一個按鈕，在按一下時執行自動化。

Tip

您也可以使用叫用自動化動作從其他自動化執行自動化。

從按鈕執行自動化

1. 編輯應用程式時，請選擇分頁索引標籤。
2. 在右側功能表中，選擇按鈕元件，將按鈕新增至頁面。
3. 選擇新按鈕進行設定。
4. 在屬性右側選單的觸發中，選擇 + 新增，然後選擇叫用自動化。
5. 選擇新的自動化調用觸發條件以進行設定。
6. 在叫用自動化中，選取叫用 Lambda 函數的自動化，並設定您要傳送至自動化的任何參數。

現在，在您應用程式中選擇此按鈕的任何使用者都會導致設定的自動化執行。

後續步驟：預覽和發佈要測試的應用程式

您的應用程式現在已準備好進行測試。在開發環境中預覽您的應用程式時，連接器不會處於作用中狀態，因此您無法在預覽時測試自動化，因為它使用連接器來連線 AWS Lambda。若要測試依賴連接器的應用程式功能，您必須將應用程式發佈至測試環境。如需預覽和發佈應用程式的詳細資訊，請參閱[預覽、發佈和共用應用程式](#)。

使用生成式 AI 建置您的 App Studio 應用程式

AWS App Studio 提供整合的生成式 AI 功能，以加速開發並簡化常見任務。您可以利用生成式 AI 來產生和編輯應用程式、資料模型、範例資料，甚至在建置應用程式時取得情境說明。

產生您的應用程式

若要加速啟動，您可以使用 AI 提供的自然語言提示來產生整個應用程式。此功能可讓您描述所需的應用程式功能，AI 會自動建置資料模型、使用者介面、工作流程和連接器。如需使用 AI 產生應用程式的詳細資訊，請參閱 [建立應用程式](#)。

建置或編輯您的應用程式

編輯應用程式時，您可以使用聊天功能來描述您要進行的變更，並自動更新您的應用程式。您可以從現有的範例提示中選擇，或輸入您自己的提示。聊天可用來新增、編輯和移除支援的元件，也可以建立和設定自動化和動作。使用下列程序來使用 AI 來編輯或建置您的應用程式。

使用 AI 編輯您的應用程式

1. 如有必要，請編輯您的應用程式以導覽至應用程式工作室。
2. （選用）選取您要使用 AI 編輯的頁面或元件。
3. 選擇左下角的建置 AI 以開啟聊天。
4. 輸入您要進行的變更，或從範例提示中選擇。
5. 檢閱要進行的變更。如果您想要進行變更，請選擇確認。否則，請輸入另一個提示。
6. 檢閱變更的摘要。

產生資料模型

您可以根據提供的實體名稱，自動產生具有欄位、資料類型和資料動作的實體。如需建立實體的詳細資訊，包括使用 GenAI 建立實體，請參閱 [在 App Studio 應用程式中建立實體](#)。

您也可以透過下列方式更新現有實體：

- 將更多欄位新增至實體。如需詳細資訊，請參閱 [新增、編輯或刪除實體欄位](#)。
- 將資料動作新增至實體。如需詳細資訊，請參閱 [建立資料動作](#)。

產生範例資料

您可以根據實體的欄位產生實體的範例資料。這有助於在連接外部資料來源之前測試您的應用程式，或在開發環境中測試您的應用程式，這不會與外部資料來源通訊。如需詳細資訊，請參閱 [新增或刪除範例資料](#)。

一旦您將應用程式發佈至測試或生產，您的即時資料來源和連接器就會用於這些環境。

設定 AWS 服務的動作

與 Amazon Simple Email Service 等 AWS 服務整合時，您可以使用 AI 根據所選服務產生預先填入欄位的範例組態。若要試用，請在叫用 AWS 自動化動作的屬性功能表中，選擇雙邊箭頭來展開組態欄位。然後，選擇產生範例組態。

模擬回應

您可以產生 AWS 服務動作的模擬回應。這有助於在開發環境中測試您的應用程式，這不會與外部資料來源通訊。

在建置時向 AI 尋求協助

在應用程式工作室中，您可以找到有關支援資源或屬性的 Ask AI for help 按鈕。使用此項目可取得與目前檢視或所選元件相關的內容建議、文件和指引。詢問有關 App Studio、應用程式建置最佳實務或特定應用程式使用案例的一般問題，以接收量身打造的資訊和建議。

建立、編輯和刪除應用程式

內容

- [建立應用程式](#)
- [匯入應用程式](#)
 - [App Studio 提供的可匯入應用程式](#)
- [複製應用程式](#)
- [編輯或建置應用程式](#)
- [編輯先前發佈的應用程式版本](#)
- [重新命名應用程式](#)
- [刪除應用程式](#)

建立應用程式

使用下列程序在 App Studio 中建立應用程式。

建立應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式，以導覽至您的應用程式清單。

2. 選擇 + 建立應用程式。
3. 在建立應用程式對話方塊中，為您的應用程式命名，然後選擇下列其中一個應用程式建立方法：
 - 使用 AI 產生應用程式：選擇此選項以自然語言描述您的應用程式，並讓 AI 為您產生應用程式及其資源。
 - 從頭開始：選擇此選項以從空白應用程式開始建置。
4. 選擇下一步。
5. 如果您選擇使用 AI 產生應用程式：
 - a. 在連線至現有資料對話方塊中，選取提供 App Studio 存取資料來源的連接器，然後選取 Tablese，然後選擇下一步，將任何現有資料來源新增至您的應用程式。在此處新增資料來源有助於 AI 為您產生最佳化應用程式。您可以略過此步驟，稍後再選擇略過來新增資料來源。
 - b. 在短暫延遲（幾分鐘）後，系統會將您導向使用 AI 產生應用程式頁面，您可以在其中描述您要建立的應用程式。
 - c. 您可以在聊天中開始描述您的應用程式，也可以選擇和自訂提供的範例提示。
 - d. 分析您的提示後，請檢閱應用程式需求和概觀。使用聊天功能請求任何變更，或選擇重新開始以從空白提示開始。
 - e. 準備就緒時，選擇產生應用程式。
 - f. 產生後，請選擇預覽應用程式，在另一個索引標籤中預覽您的應用程式。當您準備好開始編輯時，可以選擇編輯應用程式。瀏覽應用程式的頁面、自動化和資料，以熟悉它。檢閱底部偵錯面板中的任何錯誤或警告。若要了解如何使用 AI 產生應用程式，請參閱 [教學課程：使用 AI 產生應用程式](#)。如需有關在 App Studio 中建置如何運作的一般資訊，請參閱 [AWS App Studio 的運作方式](#)。
6. 如果您選擇從頭開始：
 - a. 在連線至現有資料對話方塊中，選取提供 App Studio 存取資料來源的連接器，然後選取 Tablese，然後選擇下一步，將任何現有資料來源新增至您的應用程式。您可以略過此步驟，稍後再選擇略過來新增資料來源。
 - b. 建立應用程式後，請選擇編輯應用程式以開始編輯應用程式。若要了解如何從空白應用程式建置，請參閱 [教學課程：從空白應用程式開始建置](#)。如需有關在 App Studio 中建置如何運作的一般資訊，請參閱 [AWS App Studio 的運作方式](#)。

匯入應用程式

您可以將匯出應用程式的副本匯入 App Studio 執行個體。您可以從其他 App Studio 執行個體匯入已匯出的應用程式，或從 App Studio 提供的目錄匯入應用程式。從 App Studio 應用程式目錄匯入應用程式

可協助您開始使用具有類似功能的應用程式，或透過探索匯入的應用程式，協助您了解 App Studio 中的應用程式建置。

當您將應用程式匯入執行個體時，會在執行個體中建立原始應用程式的複本。建立新應用程式後，您會導覽至應用程式的開發環境，您可以在其中預覽應用程式並瀏覽應用程式的功能。

Warning

當您匯入應用程式時，您會從應用程式匯入所有邏輯，這可能會導致不想要或非預期的行為。例如，可能會有破壞性查詢從您連線至應用程式的資料庫刪除資料。我們建議您徹底檢閱應用程式及其組態，並在非生產資產上測試它，然後再將生產資料連接到它。

匯入應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式，以導覽至您的應用程式清單。
2. 選擇 + 建立應用程式旁的下拉式箭頭。
3. 選擇匯入應用程式。
4. 在匯入應用程式對話方塊中的匯入程式碼中，輸入您要匯入之應用程式的匯入程式碼。如需 App Studio 提供的可匯入應用程式清單，包括應用程式描述和匯入代碼，請參閱 [App Studio 提供的可匯入應用程式](#)。
5. 選擇匯入以匯入應用程式，然後前往匯入應用程式的開發環境以檢視或編輯應用程式。如需在 App Studio 中建置應用程式的資訊，請參閱 [AWS App Studio 的運作方式](#)

App Studio 提供的可匯入應用程式

App Studio 提供可匯入執行個體的應用程式，協助您了解應用程式建置。若要查看 App Studio 中如何實作特定應用程式功能，您可以預覽應用程式，然後在開發環境中瀏覽其組態。

下表包含應用程式清單、其匯入代碼，以及應用程式的簡短描述。每個應用程式都包含一個README頁面，其中包含您可以在匯入應用程式後檢視的應用程式相關資訊。

應用程式名稱	Description	匯入程式碼
Swag 請求調查	內部 swag 請求應用程式，旨在供員工訂購品牌公司商品。員工可以選取項目並指定大	Swag 請求Survey/ec4f5faf-e2f8-42ee-ab8d-6723d8ca21b2

應用程式名稱	Description	匯入程式碼
	<p>小，並透過簡單的表單提交其請求。此應用程式會透過內建儲存體處理所有資料，無需外部連線。</p>	
衝刺追蹤	<p>團隊可用來組織和追蹤軟體開發工作的衝刺管理應用程式。使用者可以建立衝刺、新增任務、指派工作，以及透過專用衝刺、追蹤和任務檢視監控進度。此應用程式會透過內建儲存體處理所有資料，無需外部連線。</p>	衝刺Tracking/8f31e160-771f-48d7-87b0-374e285e2fbc
Amazon Review 情緒追蹤器	<p>此應用程式是一種客戶意見回饋分析工具，可從產品評論產生情緒分數，以協助企業了解客戶滿意度。應用程式包含用於快速測試的範例資料產生公用程式，並需要 Amazon Bedrock 連接器來提供 AI 技術的洞見，同時在內建儲存系統中維護所有其他資料。</p>	Amazon Review 情緒Tracker /60f0dae4-f8e2-4c20-9583-fa456f5ebfab

應用程式名稱	Description	匯入程式碼
發票和接收處理	<p>此接收處理應用程式透過自動化手動資料輸入和簡化文件核准工作流程來節省時間並減少錯誤。解決方案需要 Amazon Textract、Amazon S3 和 Amazon SES 連接器。它使用 Amazon Textract 從存放在 Amazon S3 中的收據分析和擷取資料，然後使用 Amazon SES 處理擷取的資訊並透過電子郵件傳送給核准者。</p>	<p>發票和接收Processing/98bde3ae-e454-4b18-a1e6-6f23e8b2a4f1</p>
檢查和庫存稽核	<p>用於管理倉儲檢查和設備追蹤的應用程式。使用者可以依房間位置執行通過/失敗設備評估、監控庫存層級，以及檢視檢查歷史記錄。應用程式提供集中式系統，用於追蹤設施檢查和設備狀態。此應用程式會透過內建儲存體處理所有資料，無需外部連線。</p>	<p>檢查和庫存Audit/cf570a06-1c5e-4dd7-9ea8-5c04723d687f</p>
產品採用追蹤器	<p>管理產品開發的全方位應用程式，可集中客戶意見回饋、功能請求和客戶會議備註。團隊可以追蹤客戶互動、組織需求，並產生 AI 支援的報告以進行每季藍圖規劃。應用程式包含範例資料公用程式，並利用 Amazon Bedrock 取得 AI 洞見，以及利用 Amazon Aurora PostgreSQL 進行資料管理。</p>	<p>產品採用Tracker/9b3a4437-bb50-467f-ae9e-d108776b7ca1</p>

應用程式名稱	Description	匯入程式碼
快速內嵌	示範應用程式，可讓使用者在使用基礎資料時檢視分析。應用程式包含兩種在 App Studio 中內嵌 Amazon Quick 儀表板的方法：適用於已註冊和匿名使用者的 API 型方法（需要 Quick 連接器），以及適用於公有儀表板的 iFrame 整合。	Quicksight Embedding /0cdc15fc-ca8b-41b7-869e-ed13c9072bc8
Kitchen 接收器	顯示進階 App Studio 開發秘訣和最佳實務的參考應用程式。包括建置者可以在自己的應用程式中學習和實作的狀態管理、CSV 資料處理、瀏覽器 API 整合和 UI 模式的工作範例。這些範例都不需要外部連線。	App Studio Kitchen Sink/1cfe6b2f-544c-4611-b82c-80eadc76a0c8

複製應用程式

應用程式擁有者和共同擁有者可以複製其應用程式，以建立應用程式的確切複本。如果您想要保留目前狀態以供測試，或使用複製的應用程式做為新應用程式的入門者，複製應用程式會很有幫助。

複製應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式。系統會將您導向至顯示您有權存取之應用程式清單的頁面。
2. 在您要複製的應用程式的動作欄中，選擇下拉式清單。
3. 選擇 Duplicate (複製)。如果無法使用複製選用項目，您可能不是應用程式的擁有者或共同擁有者。
4. 選擇性地提供重複應用程式的名稱。預設名稱為 *Current_App_Name COPY*。
5. 選擇 Duplicate (複製)。

編輯或建置應用程式

使用下列程序在 App Studio 中編輯應用程式。

編輯（建置）應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式。系統會將您導向至顯示您有權存取之應用程式清單的頁面。
2. 在您要編輯的應用程式的動作欄中，選擇編輯。這將帶您前往開發環境，您可以在其中使用元件、自動化和資料來設定應用程式的外觀和功能。如需建置應用程式的資訊，請參閱 [AWS App Studio 入門](#)。

編輯先前發佈的應用程式版本

使用下列程序來編輯先前發佈的 App Studio 應用程式版本。選擇編輯先前發佈的版本後，您可以在開發環境中編輯應用程式，或將其發佈至測試，然後發佈至生產。

Warning

先前發佈的版本會取代開發環境中應用程式的進行中版本。任何未發佈的應用程式變更都將遺失。

如果您不小心發佈了不必要的變更或變更，而這些變更會中斷使用者的應用程式，而且您想要從先前的應用程式版本進一步建置或編輯，則編輯先前發佈的版本會很有用。

Note

如果您偵測到已發佈應用程式的問題，並需要立即發佈先前運作的版本，或者您想要發佈先前的版本，但保留應用程式在開發環境中的最新更新，則應改為轉返應用程式。如需詳細資訊，請參閱 [回復至先前發佈的版本](#)。

編輯先前發佈的應用程式版本

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇發佈按鈕旁的下拉箭頭，然後選擇發佈中心。

3. 選擇版本歷史記錄，以查看先前發佈的應用程式版本清單。
4. 尋找您要編輯的版本，然後選擇編輯。
5. 檢閱資訊，然後選擇還原。
6. 您選擇編輯的版本現在是開發環境中的目前版本。您可以進行變更，或依原樣選擇發佈，將其發佈至測試環境。發佈至測試後，您可以視需要再次發佈至生產環境。

重新命名應用程式

使用下列程序重新命名 App Studio 中的應用程式。您可以在建置應用程式時，從應用程式清單或從開發環境重新命名應用程式。

重新命名應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式。系統會將您導向至顯示您有權存取之應用程式清單的頁面。
2. 您可以從此清單或在編輯時從開發環境重新命名應用程式。
 - 若要從此清單中重新命名：
 - a. 在您要重新命名的應用程式的動作欄中選擇下拉式清單，然後選擇重新命名。
 - b. 為您的應用程式命名，然後選擇重新命名。
 - 若要從開發環境重新命名：
 - a. 在您要編輯的應用程式的動作欄中，選擇編輯。
 - b. 在開發環境中，選擇應用程式名稱並進行更新，然後按下 Enter 鍵或離開文字欄位以儲存變更。

刪除應用程式

使用下列程序刪除 App Studio 中的應用程式。

刪除 應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式。系統會將您導向至顯示您有權存取之應用程式清單的頁面。
2. 在要刪除的應用程式的動作欄中，選擇下拉式清單。
3. 選擇 刪除。

4. 在刪除應用程式對話方塊中，仔細檢閱有關刪除應用程式的資訊。如果您想要刪除應用程式，請選擇刪除。

預覽、發佈和共用應用程式

主題

- [預覽應用程式](#)
- [發佈應用程式](#)
- [共用已發佈的應用程式](#)
- [回復至先前發佈的版本](#)
- [匯出應用程式](#)

預覽應用程式

您可以在 App Studio 中預覽應用程式，查看使用者將看到的應用程式，並在偵錯面板中使用它並檢查日誌來測試其功能。

應用程式預覽環境不支援顯示即時資料或與具有連接器之外部資源的連線，例如資料來源。若要測試預覽環境中的功能，您可以在自動化中使用模擬輸出，並在實體中使用範例資料。若要使用即時資料檢視應用程式，您必須發佈應用程式。如需詳細資訊，請參閱[發佈應用程式](#)。

預覽或開發環境不會更新在其他環境中發佈的應用程式。如果應用程式尚未發佈，在發佈和共用之前，使用者將無法存取它。如果應用程式已發佈並共用，使用者仍會存取已發佈的版本，而不是預覽環境中使用的版本。

預覽您的應用程式

1. 如有必要，請導覽至您要預覽之應用程式的應用程式 Studio：
 - a. 在導覽窗格中，選擇建置區段中的我的應用程式。
 - b. 針對應用程式選擇編輯。
2. 選擇預覽以開啟應用程式的預覽環境。
3. (選用) 選擇靠近畫面底部的標頭，展開偵錯面板。您可以在篩選日誌區段中選擇訊息類型，依訊息類型篩選面板。您可以選擇清除主控台來清除面板的日誌。

4. 在預覽環境中，您可以透過導覽應用程式頁面、使用其元件，並選擇其按鈕來啟動傳輸資料的自動化來測試應用程式。由於預覽環境不支援即時資料或連線到外部來源，因此您可以在偵錯面板中檢視要傳輸的資料範例。

發佈應用程式

當您完成建立和設定應用程式後，下一步是發佈應用程式，以測試資料傳輸或與最終使用者共用資料傳輸。若要了解在 App Studio 中發佈應用程式，請務必了解可用的環境。App Studio 提供三個不同的環境，如下列清單所述：

1. 開發：您建置和預覽應用程式的位置。您不需要發佈到開發環境，因為應用程式的最新版本會自動託管在那裡。此環境中沒有即時資料或第三方服務或資源。
2. 測試：您可以在其中執行應用程式的完整測試。在測試環境中，您可以連線、傳送資料至其他服務，以及從其他服務接收資料。
3. 生產：最終使用者耗用的即時操作環境。

您的所有應用程式建置都會在開發環境中進行。然後，將存取 URL 提供給最終使用者，發佈至測試環境以測試其他服務之間的資料傳輸，以及使用者接受度測試 (UAT)。之後，將您的應用程式發佈至生產環境，在與使用者共用之前執行最終測試。如需應用程式環境的詳細資訊，請參閱 [應用程式環境](#)。

當您發佈應用程式時，在共用之前，使用者無法使用該應用程式。這可讓您在測試和生產環境中使用和測試應用程式，然後使用者才能存取應用程式。當您將應用程式發佈至先前已發佈和共用的生產環境時，使用者可用的版本會更新。


發佈應用程式

使用下列程序將 App Studio 應用程式發佈至測試或生產環境。

將應用程式發佈至測試或生產環境

1. 在導覽窗格中，選擇建置區段中的我的應用程式。系統會將您導向至顯示您有權存取之應用程式清單的頁面。
2. 針對您要發佈的應用程式選擇編輯。
3. 選擇右上角的發佈。
4. 在發佈更新對話方塊中：
 - a. 檢閱發佈應用程式的相關資訊。

- b. (選用) 在版本描述中，包含此版本應用程式的描述。
 - c. 選擇方塊以確認環境的相關資訊。
 - d. 選擇 開始使用。在即時環境中更新應用程式最多可能需要 15 分鐘。
5. 如需有關在測試或生產環境中檢視應用程式的資訊，請參閱 [檢視已發佈的應用程式](#)。

 Note

在測試或生產環境中使用應用程式會導致即時資料傳輸，例如在已與連接器連線的資料來源資料表中建立記錄。


從未共用的已發佈應用程式將無法供使用者或其他建置器使用。若要讓應用程式可供使用者使用，您必須在發佈後共用應用程式。如需詳細資訊，請參閱 [共用已發佈的應用程式](#)。

檢視已發佈的應用程式

您可以檢視發佈至測試和生產環境的應用程式，在與最終使用者或其他建置器共用應用程式之前進行測試。

在測試或生產環境中檢視已發佈的應用程式

1. 如有必要，請導覽至您要預覽之應用程式的應用程式 Studio：
 - a. 在導覽窗格中，選擇建置區段中的我的應用程式。
 - b. 針對應用程式選擇編輯。
2. 選擇右上角發佈旁邊的下拉箭頭，然後選擇發佈中心。
3. 從發佈中心，您可以檢視應用程式發佈到的環境。如果您的應用程式發佈至測試或生產環境，您可以使用每個環境的 URL 連結來檢視應用程式。

 Note

在測試或生產環境中使用應用程式會導致即時資料傳輸，例如在已與連接器連線的資料來源資料表中建立記錄。

應用程式環境

AWS App Studio 提供應用程式生命週期管理 (ALM) 功能與三個不同的環境 - 開發、測試和生產。這可協助您更輕鬆地進行最佳實務，例如在整個應用程式生命週期中維護不同的環境、版本控制、共用和監控。

開發環境

開發環境是一個隔離的沙盒，您可以在其中建置應用程式，而無需使用應用程式工作室和範例資料連線到任何即時資料來源或服務。在開發環境中，您可以預覽應用程式以檢視和測試應用程式，而不會影響生產資料。

雖然您的應用程式不會連線到開發環境中的其他服務，但您可以在應用程式中設定不同的資源來模擬即時資料連接器和自動化。

開發環境中的應用程式工作室底部有一個可摺疊的偵錯面板，其中包含錯誤和警告，可協助您在建置時檢查和偵錯應用程式。如需疑難排解和偵錯應用程式的詳細資訊，請參閱 [疑難排解和偵錯 App Studio](#)。

測試環境

初始應用程式開發完成後，下一步是發佈至測試環境。在測試環境中，您的應用程式可以連線、傳送資料至其他服務，以及從其他服務接收資料。因此，您可以使用此環境，透過向最終使用者提供存取 URL 來執行包括使用者接受度測試 (UAT) 在內的全面測試。

Note

您的初始發佈到測試環境最多可能需要 15 分鐘。

發佈至測試環境的應用程式版本會在最終使用者閒置 3 小時後移除。不過，所有版本都會保留，並且可以從版本歷史記錄索引標籤還原。

測試環境的主要功能如下：

- 與即時資料來源和 APIs 整合測試
- 透過受控存取促進的使用者接受度測試 (UAT)
- 收集意見回饋和解決問題的環境
- 能夠使用瀏覽器主控台和開發人員工具來檢查用戶端和伺服器端活動並進行偵錯。

如需疑難排解和偵錯應用程式的詳細資訊，請參閱 [疑難排解和偵錯 App Studio](#)。

生產環境

測試並修正任何問題後，您可以將應用程式版本從測試環境提升為生產環境，以供即時操作使用。雖然生產環境是最終使用者取用的即時操作環境，但您可以在與使用者共用發佈版本之前進行測試。

您在生產環境中發佈的版本會在最終使用者閒置 14 天後移除。不過，所有版本都會保留，並且可以從版本歷史記錄索引標籤還原。

生產環境的主要功能如下：

- 最終使用者耗用的即時操作環境
- 精細的角色型存取控制
- 版本控制和轉返功能
- 只能檢查和偵錯用戶端活動
- 使用即時連接器、資料、自動化和 APIs

版本控制和發行管理

App Studio 透過發佈中心的版本控制系統提供版本控制和版本管理功能。

金鑰版本控制功能：

- 發佈至測試環境會產生新的版本編號 (1.0、2.0、3.0...)。
- 從測試提升至生產環境時，版本編號不會變更。
- 您可以從版本歷史記錄轉返至任何先前的版本。
- 發佈至測試環境的應用程式會在閒置 3 小時後暫停。版本會保留，並且可以從版本歷史記錄還原。
- 發佈至生產環境的應用程式會在閒置 14 天後移除。版本會保留，並且可以從版本歷史記錄還原。

此版本控制模型允許快速迭代，同時在整個應用程式開發和測試週期中維持可追蹤性、回復功能和最佳效能。

維護和操作

App Studio 可能需要自動重新發佈您的應用程式，以解決某些維護任務、操作活動，並納入新的軟體程式庫。您、建置器不需要採取任何動作，但最終使用者可能需要重新登入應用程式。在某些情況下，

我們可能需要您重新發佈應用程式，以納入我們無法自動新增的新功能和程式庫。重新發佈之前，您可能需要解決任何錯誤並檢閱警告。

共用已發佈的應用程式

當您發佈尚未發佈的應用程式時，在共用之前，使用者無法使用該應用程式。一旦已發佈的應用程式被共用，它將可供使用者使用，如果發佈了另一個版本，則不需要再次共用。

Note

本節旨在與最終使用者或測試人員共用已發佈的應用程式。如需邀請其他使用者建置應用程式的相關資訊，請參閱 [使用多個使用者建置應用程式](#)。

共用已發佈的應用程式

1. 使用下列指示，從應用程式清單或您應用程式的應用程式工作室存取共用對話方塊：
 - 若要從應用程式清單存取共用對話方塊：在導覽窗格中，選擇建置區段中的我的應用程式。在您要共用的應用程式的動作欄中選擇下拉式清單，然後選擇共用。
 - 若要從應用程式工作室存取共用對話方塊：從應用程式的應用程式工作室中，選擇頂端標頭中的共用。
2. 在共用對話方塊中，選擇您要共用之環境的標籤。如果您沒有看到測試或生產索引標籤，您的應用程式可能不會發佈到對應的環境。如需發佈的詳細資訊，請參閱 [發佈應用程式](#)。
3. 在適當的索引標籤中，從下拉式功能表中選取群組，與他們共用環境。
4. (選用) 將應用程式層級角色指派給群組，以測試或設定條件式頁面可見性。如需詳細資訊，請參閱 [設定頁面的角色型可見性](#)。
5. 選擇共用。
6. (選用) 複製並與使用者共用連結。只有與應用程式和環境共用的使用者才能存取對應環境中的應用程式。

回復至先前發佈的版本

使用下列程序，將 App Studio 應用程式的生產環境復原至先前發佈的版本。您的應用程式最終使用者將受到影響，並在部署後查看應用程式的復原版本。當您轉返應用程式時，也會將元件程式碼轉返至先前發佈時間的版本，並影響整個應用程式部署堆疊（使用者程式碼、元件組態狀態）。這表示 App

Studio 對元件程式碼所做的任何更新，例如欄位或其他組態變更，都會復原，以確保復原的應用程式版本如最初發佈時一樣運作。

當您轉返發佈的版本時，開發環境中應用程式的進行中版本不會受到影響。

如果您偵測到已發佈應用程式的問題，且需要立即發佈先前運作的版本，或想要發佈先前的版本，並在開發環境中保留應用程式的最新更新，則轉返已發佈的應用程式版本會很有幫助。

Note

如果您想要將應用程式的開發環境還原為先前發佈的版本，您應該還原應用程式。如需詳細資訊，請參閱[編輯先前發佈的應用程式版本](#)。

將生產環境版本復原至先前發佈的應用程式版本

1. 如有必要，請透過編輯導覽至應用程式的開發環境。如需詳細資訊，請參閱[編輯或建置應用程式](#)。
2. 選擇生產環境圖磚頂端的版本下拉箭頭，以查看可供轉返的可用版本。下拉式清單包含過去 30 天內發佈的版本。如果停用此下拉式清單，可能是因為應用程式發佈已在進行中，而且只能同時進行一個發佈。
3. 選擇您要轉返的版本。
4. 輸入轉返的原因，然後選擇轉返。轉返發佈將開始，一旦完成，應用程式的生產環境將更新為所選的版本。

Note

您也可以復原之後，轉傳至先前發佈的應用程式版本。

匯出應用程式

您可以匯出應用程式的快照，以與其他 App Studio 執行個體共用。匯出應用程式時，會從應用程式的開發環境建立快照，並產生匯入碼。匯入程式碼接著可用來將應用程式匯入其他 App Studio 執行個體，以供檢視和建置。

匯出的應用程式可以匯入 App Studio AWS 區域 支援的任何 執行個體。

匯出應用程式

1. 在導覽窗格中，選擇建置區段中的我的應用程式，以導覽至您的應用程式清單。
2. 在您要匯出的應用程式的動作欄中，選擇下拉式清單。
3. 選擇 Export (匯出)。
4. 產生和共用匯入碼的程序會根據是否已為應用程式建立匯入碼而有所不同。
 - 如果尚未建立匯入碼：
 - a. 在應用程式匯入許可中，指定哪些執行個體可以匯入匯出的應用程式。您可以為所有執行個體提供匯入許可，或輸入其執行個體 IDs 來新增特定的 App Studio 執行個體。使用逗號分隔多個執行個體 IDs。

若要尋找執行個體 ID，請在 App Studio 主控台中選擇帳戶設定，以導覽至執行個體的帳戶設定。
 - b. 選擇產生匯入碼。
 - c. 複製並共用產生的匯入碼。
 - 如果已建立匯入碼：
 - 若要共用目前匯出的應用程式，請複製並共用現有的匯入碼。若要使用應用程式的最新變更建立新的匯出應用程式，請選擇產生新程式碼。您也可以視需要更新匯入許可。

頁面和元件：建置應用程式的使用者介面

主題

- [管理頁面](#)
- [管理元件](#)
- [設定頁面的角色型可見性](#)
- [在應用程式導覽中排序和組織頁面](#)
- [使用應用程式主題變更應用程式中的顏色](#)
- [元件參考](#)

管理頁面

使用下列程序從 AWS App Studio 應用程式建立、編輯或刪除頁面。

頁面是[元件](#)的容器，構成 App Studio 中應用程式的 UI。每個頁面都代表應用程式使用者介面 (UI) 的畫面，您的使用者將與其互動。頁面會在應用程式工作室的頁面索引標籤中建立和編輯。

建立頁面

使用下列程序在 App Studio 的應用程式中建立頁面。如需複製現有頁面的資訊，請參閱 [複製頁面](#)。

建立頁面

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側頁面選單中，選擇 + 新增。

複製頁面

使用下列程序在 App Studio 的應用程式中複製頁面。

複製頁面

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側頁面選單中，選擇您要複製之頁面名稱旁的省略符號選單，然後選擇複製。複製的頁面會直接新增到原始頁面之後。

檢視和編輯頁面屬性

使用下列程序在 App Studio 的應用程式中編輯頁面。您可以編輯頁面名稱、參數及其配置等屬性。

檢視或編輯頁面屬性

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側分頁選單中，選擇您要編輯的分頁名稱旁的省略符號選單，然後選擇分頁屬性。這會開啟右側屬性功能表。
4. 若要編輯頁面名稱：

Note

有效的頁面名稱字元：A-Z、a-z、0-9、_、\$

- a. 選擇屬性功能表頂端附近名稱旁的鉛筆圖示。
 - b. 輸入您頁面的新名稱，然後按 Enter 鍵。
5. 若要建立、編輯或刪除頁面參數：
- a. 若要建立頁面參數，請在頁面參數區段中選擇 + 新增。
 - b. 若要編輯頁面參數的金鑰或描述值，請選擇您要變更之屬性的輸入欄位，然後輸入新值。您的變更會在您編輯時儲存。
 - c. 若要刪除頁面參數，請選擇您要刪除之頁面參數的垃圾桶圖示。
6. 若要新增、編輯或移除頁面的標誌或橫幅：
- a. 若要新增頁面標誌或橫幅，請在樣式區段中啟用個別選項。設定影像的來源，並選擇性地提供替代文字。
 - b. 若要編輯頁面標誌或橫幅，請更新樣式區段中的欄位。
 - c. 若要移除頁面標誌或橫幅，請在樣式區段中停用個別選項。
7. 若要編輯頁面的配置：
- 更新配置區段中的欄位。

刪除頁面

使用下列程序從 App Studio 中的應用程式刪除頁面。

刪除頁面

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側頁面選單中，選擇您要刪除之頁面名稱旁的省略符號選單，然後選擇刪除。

管理元件

使用下列程序，在 App Studio 應用程式工作室的頁面中新增、編輯和刪除元件，為您的應用程式打造所需的使用者介面。

將元件新增至頁面

使用下列程序將元件新增至 App Studio 中的頁面。如需複製現有元件的詳細資訊，請參閱 [複製元件](#)。

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 元件面板位於右側選單中，其中包含可用的元件。
4. 將所需的元件從面板拖放到畫布上。或者，您可以按兩下面板中的元件，將其自動新增至目前頁面的中心。
5. 現在您已新增元件，請使用右側屬性面板來調整其設定，例如資料來源、配置和行為。如需設定每個元件類型的詳細資訊，請參閱 [元件參考](#)。

複製元件

使用下列程序在 App Studio 應用程式中複製元件。重複的元件包含來自原始元件的任何連結自動化或實體。

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 有兩種方式可以複製元件：
 - 在左側頁面選單中，展開包含您要複製之元件的頁面。選擇您要複製之元件名稱旁邊的省略符號功能表，然後選擇複製。
 - 選擇您要複製的元件，然後選擇複製圖示。

複製的元件會直接在原始元件之後新增。

Tip

您可以使用 CTRL+Z 或 CMD+Z 鍵盤快速鍵，復原元件複製，以及開發環境中的許多其他動作。

檢視和編輯元件屬性

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側頁面選單中，展開包含元件的頁面，然後選擇要檢視或編輯的元件。或者，您可以選擇頁面，然後從畫布中選擇元件。
4. 右側屬性面板會顯示所選元件的可設定設定。
5. 探索各種可用的屬性和選項，並視需要更新它們，以設定元件的外觀和行為。例如，您可能想要變更資料來源、設定配置或啟用其他功能。

如需設定每個元件類型的詳細資訊，請參閱 [元件參考](#)。

刪除元件

1. 如有必要，請透過編輯導覽至應用程式的開發環境。
2. 導覽至頁面索引標籤。
3. 在左側頁面選單中，選擇要刪除的元件以進行選取。
4. 在右側屬性功能表中，選擇垃圾桶圖示。
5. 在確認對話方塊中，選擇 Delete (刪除)。

設定頁面的角色型可見性


您可以在 App Studio 應用程式中建立角色，並根據這些角色設定頁面的可見性。例如，您可以根據使用者需求或存取層級建立角色，例如提供專案核准或宣告處理等功能的應用程式的管理員、管理員或使用者，並讓特定角色可以看到特定頁面。在此範例中，管理員可能具有完整存取權、管理員可能有權檢視報告儀表板，而使用者可能有權使用輸入表單存取任務頁面。

使用下列程序來設定 App Studio 應用程式中頁面的角色型可見性。

1. 如有必要，請導覽至您應用程式的應用程式工作室。從左側導覽功能表中，選擇我的應用程式，尋找您的應用程式，然後選擇編輯。
2. 在應用程式工作室中建立應用程式層級角色。
 - a. 選擇應用程式工作室頂端的應用程式設定索引標籤。
 - b. 選擇 + 新增角色

- c. 在角色名稱中，提供名稱以識別您的角色。我們建議您使用描述群組存取層級或職責的名稱，因為您會使用名稱來設定頁面可見性。
 - d. 或者，在描述中新增角色的描述。
 - e. 重複這些步驟，視需要建立任意數量的角色。
3. 設定頁面的可見度
- a. 選擇應用程式工作室頂端的分頁。
 - b. 從左側頁面選單中，選擇要設定角色型可見性的頁面。
 - c. 在右側功能表中，選擇屬性索引標籤。
 - d. 在可見性中，停用開放給所有最終使用者。
 - e. 保持選取角色，從您在上一個步驟中建立的角色清單中進行選擇。選擇自訂以撰寫 JavaScript 表達式，以取得更複雜的可見性組態。
 1. 選取角色後，勾選顯示頁面的應用程式角色方塊。
 2. 選取自訂後，輸入解析為 true 或 false 的 JavaScript 表達式。使用下列範例來檢查目前使用者是否具有 manager 的角色：

```
{{currentUser.roles.includes('manager')}}}
```
4. 現在您的可見性已設定完成，您可以透過預覽您的應用程式來測試頁面可見度。
- a. 選擇預覽以開啟應用程式的預覽。
 - b. 在預覽的右上角，選擇預覽為功能表，並勾選您要測試的角色方塊。可見頁面應反映選取的角色。
5. 現在，將群組指派給已發佈應用程式的應用程式角色。群組和角色指派必須針對每個環境分別設定。如需應用程式環境的詳細資訊，請參閱 [應用程式環境](#)。

 Note

您的應用程式必須發佈至測試或生產環境，才能將 App Studio 群組指派給您已建立和設定的角色。如有必要，請發佈您的應用程式，將群組指派給角色。如需發佈的詳細資訊，請參閱 [發佈應用程式](#)。

- a. 在應用程式工作室的右上角，選擇共用。
- b. 選擇您要設定頁面可見性之環境的標籤。

- c. 選擇搜尋群組輸入方塊，然後選擇要與其共用應用程式版本的群組。您可以輸入文字來搜尋群組。
- d. 在下拉式選單中，選擇要指派給群組的角色。您可以選擇無角色來共用應用程式版本，而不是將角色指派給群組。只有沒有角色的群組可以看到所有使用者都可看見的頁面。
- e. 選擇共用。重複這些步驟，視需要新增任意數量的群組。

在應用程式導覽中排序和組織頁面

本主題包含有關在 App Studio 應用程式中重新排序和組織頁面的資訊。產品有兩個區域顯示應用程式頁面：在應用程式工作室編輯應用程式時，在左側頁面選單中，以及已發佈應用程式預覽的左側導覽。

編輯應用程式時，在左側頁面選單中排序頁面

在應用程式工作室中編輯應用程式時，頁面會依左側頁面選單中的建立時間排序。您無法重新排序此選單中的頁面。

在預覽版或已發佈應用程式的導覽中排序、顯示或隱藏頁面

您可以編輯預覽或已發佈應用程式的左側導覽的下列設定：

- 整個導覽的可見性
- 導覽中特定頁面的可見性
- 導覽中的頁面順序

編輯預覽或已發佈應用程式的左側導覽

1. 如有必要，請導覽至應用程式的應用程式工作室進行編輯。
2. 在左側頁面選單中，選擇標頭和導覽。
3. 在右側標頭和導覽選單中，檢視或編輯下列項目：
 - a. 若要隱藏或顯示應用程式中的導覽，請使用應用程式導覽切換。
 - b. 若要隱藏應用程式導覽中的頁面，請將頁面拖曳至取消連結頁面區段。
 - c. 若要重新排序應用程式導覽中的頁面，請在連結頁面區段中將其拖曳至所需的順序。

使用應用程式主題變更應用程式中的顏色

使用下列程序，透過設定應用程式佈景主題來更新應用程式中的顏色。

1. 如有必要，請導覽至您應用程式的應用程式工作室進行編輯。
2. 在應用程式工作室中，導覽至分頁索引標籤。
3. 在左側導覽中，選擇應用程式佈景主題以開啟右側應用程式佈景主題設定。
4. 在基本佈景主題中，選擇淺色模式或深色模式。
5. 若要將自訂顏色新增至應用程式，請啟用自訂切換並更新下列設定：
 - a. 在主要顏色中，選擇套用至特定元件和應用程式導覽的顏色。您可以使用顏色挑選器、RGB、HSL 或 HEX 代碼選擇顏色。

Note

App Studio 會自動確保您的顏色可供存取。例如，如果您在光線模式下選擇淺色，則會將其更新為更易於存取。

- b. 在標頭顏色中，選擇套用至應用程式標頭的顏色。您可以使用顏色挑選器、RGB、HSL 或 HEX 代碼選擇顏色。
 - c. 選擇預設佈景主題以檢視並從預先定義的佈景主題中進行選擇，或選擇隨機化以產生隨機的主要和標頭顏色。
6. 選擇儲存變更以更新您的應用程式主題。

元件參考

本主題詳細說明每個 App Studio 的元件、其屬性，並包含組態範例。

常見元件屬性

本節概述應用程式工作室中多個元件共用的一般屬性和功能。每個屬性類型的特定實作詳細資訊和使用案例可能會因元件而有所不同，但這些屬性的一般概念在 App Studio 中保持不變。

名稱

每個元件都會產生預設名稱；不過，您可以編輯，將變更為每個元件的唯一名稱。您將使用此名稱來參考元件及其來自相同頁面中其他元件或表達式的資料。限制：請勿在元件名稱中包含空格；它只能有字母、數字、底線和貨幣符號。範例：usernameInput、ordersTable、metricCard1。

主要值、次要值和值

應用程式工作室中的許多元件會提供欄位來指定值或表達式，以決定元件內顯示的內容或資料。這些欄位通常標示為 Primary value、Secondary value 或 Value，取決於元件類型和用途。

Primary value 欄位通常用於定義應在元件內顯著顯示的主要值、資料點或內容。

可用時，Secondary value 欄位會用來顯示額外或支援值，或與主要值一起顯示的資訊。

Value 欄位可讓您指定應該顯示在元件中的值或表達式。

這些欄位同時支援靜態文字輸入和動態表達式。透過使用表達式，您可以參考應用程式中其他元件、資料來源或變數的資料，以啟用動態和資料驅動的內容顯示。

表達式的語法

在這些欄位中輸入表達式的語法遵循一致的模式：

```
{{expression}}
```

其中###是評估為您想要顯示之所需值或資料的有效運算式。

範例：靜態文字

- 主要值：您可以直接輸入靜態數字或值，例如 "123" 或 "\$1,999.99"。
- 次要值：您可以輸入靜態文字標籤，例如 "Goal" 或 "Projected Revenue"。
- 值：您可以輸入靜態字串，例如 "since last month" 或 "Total Quantity"。

範例：運算式

- Hello, {{currentUser.firstName}}：顯示問候語，其中包含目前登入使用者的名字。
- {{currentUser.role === 'Admin' ? 'Admin Dashboard' : 'User Dashboard'}}：根據使用者的角色，有條件地顯示不同的儀表板標題。
- {{ui.componentName.data?.[0]?.fieldName}}：從 ID 為 的元件資料的第一個項目擷取 fieldName 欄位的值 componentName。
- {{ui.componentName.value * 100}}：對 ID 為 的元件值執行計算 componentName。
- {{ui.componentName.value + ' items'}}：將元件的值與 ID componentName 和字串 串連 ' items'。

- `{{ui.ordersTable.data?.[0]?.orderNumber}}` : 從ordersTable元件中的第一列資料擷取訂單號碼。
- `{{ui.salesMetrics.data?.[0]?.totalRevenue * 1.15}}` : 透過將salesMetrics元件中第一列資料的總收入增加 15% 來計算預計收入。
- `{{ui.customerProfile.data?.[0]?.firstName + ' ' + ui.customerProfile.data?.lastName}}` : 從customerProfile元件中的資料串連名字和姓氏。
- `{{new Date(ui.orderDetails.data?.orderDate).toLocaleDateString()}}` : 將orderDetails元件的訂單日期格式化為更易讀的日期字串。
- `{{ui.productList.data?.length}}` : 顯示連線至productList元件之資料中的產品總數。
- `{{ui.discountPercentage.value * ui.orderTotal.value}}` : 根據折扣百分比和訂單總計計算折扣金額。
- `{{ui.cartItemCount.value + ' items in cart'}}` : 顯示購物車中的項目數量，以及標籤 items in cart。

透過使用這些表達式欄位，您可以在應用程式中建立動態和資料驅動的內容，讓您可以顯示根據使用者的內容或應用程式狀態量身打造的資訊。這可實現更個人化的互動式使用者體驗。

標籤

Label 屬性可讓您指定元件的字幕或標題。此標籤通常會在元件旁邊或上方顯示，協助使用者了解其用途。

您可以使用靜態文字和表達式來定義標籤。

範例：靜態文字

如果您在標籤欄位中輸入文字 "First Name"，元件會顯示 "First Name" 做為其標籤。

範例：運算式

範例：零售商店

下列範例會為每個使用者個人化標籤，讓界面感覺更適合個別使用者：

```
{{currentUser.firstName}} {{currentUser.lastName}}'s Account
```

範例：SaaS 專案管理

下列範例會從選取的專案提取資料，以提供內容特定的標籤，協助使用者在應用程式中保持方向：

```
Project {{ui.projectsTable.selectedRow.id}} - {{ui.projectsTable.selectedRow.name}}
```

範例：醫療保健診所

下列範例參考目前使用者的描述檔和醫生的資訊，為患者提供更個人化的體驗。

```
Dr. {{ui.doctorProfileTable.data.firstName}}  
    {{ui.doctorProfileTable.data.lastName}}
```

預留位置

預留位置屬性可讓您指定在元件為空時顯示在元件中的提示或指引文字。這可協助使用者了解預期的輸入格式，或提供額外的內容。

您可以使用靜態文字和表達式來定義預留位置。

範例：靜態文字

如果您在Enter your name預留位置欄位中輸入文字，元件會顯示Enter your name為預留位置文字。

範例：運算式

範例：金融服務

Enter the amount you'd like to deposit into your
{{ui.accountsTable.selectedRow.balance}} account 這些範例會從選取的帳戶提取資料，以顯示相關提示，讓銀行客戶的界面直覺化。

範例：電子商務

Enter the coupon code for {{ui.cartTable.data.currency}} total 這裡的預留位置會根據使用者的購物車內容動態更新，提供順暢的結帳體驗。

範例：醫療保健診所

Enter your {{ui.patientProfile.data.age}}-year-old patient's symptoms 透過使用參考病患年齡的表達式，應用程式可以建立更個人化且實用的預留位置。

來源

Source 屬性可讓您選取元件的資料來源。選擇後，您可以選擇下列資料來源類型：entity、expression或 automation。

實體

選取實體做為資料來源可讓您將元件連接到應用程式中現有的資料實體或模型。當您有定義良好的資料結構或結構描述想要在整個應用程式中利用時，這會很有用。

何時使用實體資料來源：

- 當您的資料模型或實體包含要在元件中顯示的資訊時（例如，具有 "Name"、"Description"、"Price" 等欄位的 "Products" 實體）。
- 當您需要動態從資料庫、API 或其他外部資料來源擷取資料，並將其呈現在元件中時。
- 當您想要利用應用程式資料模型中定義的關係和關聯時。

選取實體上的查詢

有時，您可能想要將元件連接到從實體而非整個實體擷取資料的特定查詢。在實體資料來源中，您可以選擇從現有查詢中選擇或建立新的查詢。

透過選取查詢，您可以：

- 根據特定條件篩選元件中顯示的資料。
- 將參數傳遞至查詢，以動態篩選或排序資料。
- 利用查詢中定義的複雜聯結、彙總或其他資料處理技術。

例如，如果您的應用程式中有具有 Name、Email 和 等欄位的 Customers 實體 PhoneNumber。您可以將資料表元件連接到此實體，然後選擇根據客戶狀態篩選的預先定義 Active Customers 資料動作。這可讓您在資料表中僅顯示作用中的客戶，而不是整個客戶資料庫。

將參數新增至實體資料來源

使用實體做為資料來源時，您也可以將參數新增至元件。這些參數可用來篩選、排序或轉換元件中顯示的資料。

例如，如果您的 Products 實體具有 Name、Price、Description 和 等欄位 Category。您可以將名為 的參數新增至顯示產品清單的 category 資料表元件。當使用者從下拉式清單中選取類別時，

資料表會自動更新，以使用資料動作中的 `{{params.category}}` 表達式，僅顯示屬於所選類別的產品。

表達式

選取表達式做為資料來源，以輸入自訂表達式或計算，動態產生元件的資料。當您需要執行轉換、合併來自多個來源的資料，或根據特定商業邏輯產生資料時，這會很有用。

何時使用表達式資料來源：

- 當您需要計算或衍生資料模型中未直接提供的資料時（例如，根據數量和價格計算總訂單值）。
- 當您想要結合來自多個實體或資料來源的資料來建立複合檢視時（例如，顯示客戶的訂單歷史記錄及其聯絡資訊）。
- 當您需要根據特定規則或條件產生資料時（例如，根據使用者的瀏覽歷史記錄顯示「建議產品」清單）。

例如，如果您的 `##` 元件需要顯示當月的總收入，您可以使用如下的表達式來計算和顯示每月收入：

```
{{ui.table1.orders.concat(ui.table1.orderDetails).filter(o => o.orderDate.getMonth() === new Date().getMonth()).reduce((a, b) => a + (b.quantity * b.unitPrice), 0)}}
```

自動化

選取自動化做為資料來源，將元件連線至應用程式中現有的自動化或工作流程。當在特定程序或工作流程中產生或更新元件的資料或功能時，這很有用。

何時使用自動化資料來源：

- 當元件中顯示的資料是特定自動化或工作流程的結果時（例如，在核准程序中更新的「待核准」資料表）。
- 當您想要根據自動化中的事件或條件來觸發元件的動作或更新時（例如，使用 SKU 的最新銷售數字更新指標）。
- 當您需要透過自動化將元件與應用程式中的其他服務或系統整合時（例如，從第三方 API 擷取資料並將其顯示在資料表中）。

例如，如果您有引導使用者完成任務應用程式程序的步驟流程元件。步驟流程元件可以連接到處理任務應用程式提交、背景檢查和產生優惠的自動化。隨著自動化進行這些步驟，步驟流程元件可以動態更新，以反映應用程式的目前狀態。

透過仔細為每個元件選取適當的資料來源，您可以確保應用程式的使用者介面是由正確的資料和邏輯提供支援，為您的使用者提供無縫且引人入勝的體驗。

如果 可見

使用 Visible if 屬性，根據特定條件或資料值顯示或隱藏元件或元素。當您想要動態控制應用程式使用者介面特定部分的可見性時，此功能非常有用。

屬性使用以下語法時顯示：

```
{{expression ? true : false}}
```

或

```
{{expression}}
```

其中###是評估為 true或 的布林運算式false。

如果表達式評估為 true，則會顯示元件。如果表達式評估為 false，元件將會隱藏。表達式可以參考應用程式中其他元件、資料來源或變數的值。

如果表達式範例可見

範例：根據電子郵件輸入顯示或隱藏密碼輸入欄位

假設您有一個包含電子郵件輸入欄位和密碼輸入欄位的登入表單。只有在使用者已輸入電子郵件地址時，才想要顯示密碼輸入欄位。如果表達式為 `{{ui.emailInput.value != ""}}`，您可以使用下列可見：

```
{{ui.emailInput.value != ""}}
```

此表達式會檢查emailInput元件的值是否不是空字串。如果使用者已輸入電子郵件地址，表達式會評估為 true，且密碼輸入欄位會顯示。如果電子郵件欄位為空白，表達式會評估為 false，且密碼輸入欄位會隱藏。

範例：根據下拉式清單選擇顯示其他表單欄位

假設您有一個表單，使用者可以從下拉式清單中選取類別。根據選取的類別，您想要顯示或隱藏其他表單欄位，以收集更具體的資訊。

例如，如果使用者選取##類別，您可以使用下列表達式來顯示額外的#####欄位：

```
{{ui.categoryDropdown.value === "Products"}}
```

如果使用者選取####類別，您可以使用此表達式來顯示一組不同的其他欄位：

```
{{ui.categoryDropdown.value === "Services" || ui.categoryDropdown.value === "Consulting"}}
```

範例：其他

如果元件的值不是空字串，則讓textInput1元件可見：

```
{{ui.textInput1.value !== "" ? false : true}}
```

若要讓元件始終可見：

```
{{true}}
```

如果元件的值不是空字串，則讓emailInput元件可見：

```
{{ui.emailInput.value !== ""}}
```

停用，如果

停用功能可讓您根據特定條件或資料值，有條件啟用或停用元件。這是透過使用 Disabled if 屬性來實現的，該屬性接受布林表達式來決定應該啟用或停用元件。

如果 屬性使用以下語法，則為已停用：

```
{{expression ? true : false}}
```

或

```
{{expression}}
```

如果表達式範例已停用

範例：根據表單驗證停用提交按鈕

如果您有包含多個輸入欄位的表單，而且您想要停用提交按鈕，直到正確填寫所有必要欄位，您可以使用下列已停用 如果表達式：

```
{{ui.nameInput.value === "" || ui.emailInput.value === "" || ui.passwordInput.value === ""}}
```

此表達式會檢查是否有任何必要的輸入欄位 (nameInput、emailInput、passwordInput) 是空的。如果任何欄位為空白，表達式會評估為 true，且提交按鈕將會停用。填寫所有必要欄位後，表達式會評估為 false，並啟用提交按鈕。

透過在可見 和已停用 ff 屬性中使用這些類型的條件式表達式，您可以建立動態和回應式使用者介面，以適應使用者輸入，為應用程式的使用者提供更簡化和相關的體驗。

其中###是評估為 true 或 false 的布林運算式。

範例：

```
{{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.
```

容器配置

配置屬性會決定元件中的內容或元素如何配置和定位。有多種配置選項可用，每個選項都以圖示表示：

- 資料欄配置：此配置會在單一資料欄中垂直排列內容或元素。
- 兩欄配置：此配置會將元件分成兩個等寬欄，讓您並排放置內容或元素。
- 資料列配置：此配置會在單一資料列中水平排列內容或元素。

方向

- 水平：此配置會在單一資料列中水平排列內容或元素。
- 垂直：此配置會在單一欄中垂直排列內容或元素。
- 內嵌包裝：此配置會水平排列內容或元素，但如果元素超過可用寬度，則會包裝至下一行。

Alignment

- 左側：將內容或元素對齊元件的左側。
- 中心：水平置中元件中的內容或元素。
- 右側：將內容或元素對齊元件的右側。

Width

Width 屬性指定元件的水平大小。您可以輸入介於 0% 到 100% 之間的百分比值，代表元件相對於其父容器或可用空間的寬度。

Height

高度屬性指定元件的垂直大小。「自動」值會根據元件的內容或可用空間自動調整元件的高度。

之間的空間

屬性之間的空間決定元件內內容或元素之間間距或間隙。您可以選擇從 0px（無間距）到 64px 的值，增量為 4px（例如 4px、8px、12px 等）。

填補

填補屬性控制內容或元素與元件邊緣之間的空間。您可以選擇從 0px（無填補）到 64px 的值，增量為 4px（例如 4px、8px、12px 等）。

背景介紹

背景啟用或停用元件的背景顏色或樣式。

這些配置屬性提供在元件內配置和定位內容的彈性，以及控制元件本身的大小、間距和視覺化外觀。

資料元件

本節涵蓋應用程式工作室中可用的各種資料元件，包括資料表、詳細資訊、指標、表單和延伸器元件。這些元件用於顯示、收集和操作應用程式中的資料。

資料表

資料表元件會以表格格式顯示資料，其中包含資料列和資料欄。它用於以有條不紊且 easy-to-read 的方式呈現結構化資料，例如資料庫中的項目或記錄清單。

資料表屬性

資料表元件與其他元件共用數個常見屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

除了常見的屬性之外，資料表元件還具有特定的屬性和組態選項，包括 Columns、Search and export 和 Expressions。

資料欄

在本節中，您可以定義要在資料表中顯示的欄。每個資料欄都可以設定下列屬性：

- 格式：欄位的資料類型，例如：文字、數字、日期。
- 資料欄標籤：資料欄的標頭文字。
- 值：來自資料來源的欄位，應顯示於此欄中。

此欄位可讓您指定應該顯示在資料欄儲存格中的值或表達式。您可以使用表達式來參考來自連線來源或其他元件的資料。

範例：`{{currentRow.title}}`- 此表達式會顯示資料欄儲存格中目前資料列的##欄位值。

- 啟用排序：此切換可讓您啟用或停用特定資料欄的排序功能。啟用時，使用者可以根據此欄中的值來排序資料表資料。

搜尋和匯出

資料表元件提供下列切換，以啟用或停用搜尋和匯出功能：

- 顯示搜尋 啟用時，此切換會將搜尋輸入欄位新增至資料表，允許使用者搜尋和篩選顯示的資料。
- 顯示匯出 啟用時，此切換會將匯出選項新增至資料表，允許使用者以各種格式下載資料表資料，例如：CSV。

Note

根據預設，搜尋功能僅限於已載入資料表的資料。若要完整使用搜尋，您需要載入所有資料頁面。

每頁的資料列數

您可以指定表格中每個頁面要顯示的列數。然後，使用者可以在頁面之間導覽以檢視完整的資料集。

預先擷取限制

指定每個查詢請求中要預先擷取的記錄數目上限。上限為 3000。

動作

在動作區段中，設定下列屬性：

- 動作位置：啟用向右固定時，任何新增的動作一律會顯示在資料表右側，無論使用者捲動為何。
- 動作：將動作按鈕新增至資料表。您可以設定這些按鈕在使用者點按時執行指定的動作，例如：
 - 執行元件動作
 - 導覽至不同的頁面
 - 叫用資料動作
 - 執行自訂 JavaScript
 - 叫用自動化

表達式

資料表元件提供數個使用表達式和資料列層級動作功能的區域，可讓您自訂和增強資料表的功能和互動性。它們可讓您動態參考和顯示資料表中的資料。透過利用這些表達式欄位，您可以建立動態資料欄、將資料傳遞至資料列層級動作，以及參考應用程式中其他元件或表達式的資料表資料。

範例：參考資料列值

`{{currentRow.columnName}}` 或 `{{currentRow["Column Name"]}}` 這些表達式可讓您參考正在轉譯的目前資料列之特定資料欄的值。將 `columnName` 或 `Column Name` 取代為您要參考的資料欄的實際名稱。

範例：

- `{{currentRow.productName}}` 顯示目前資料列的產品名稱。
- `{{currentRow["Supplier Name"]}}` 顯示目前資料列的供應商名稱，其中資料欄標頭是#####。
- `{{currentRow.orderDate}}` 顯示目前資料列的訂單日期。

範例：參考選取的資料列

`{{ui.table1.selectedRow["columnName"]}}` 此表達式可讓您使用 ID 資料表 1 來參考資料表中目前所選資料列的特定資料欄值。將 `table1` 取代為資料表元件的實際 ID，並將 `columnName` 取代為您要參考的欄名稱。

範例：

- `{{ui.ordersTable.selectedRow["totalAmount"]}}` 使用 ID `ordersTable` 顯示資料表中目前選取列的總金額。

- `{{ui.customersTable.selectedRow["email"]}}` 使用 ID `customersTable` 顯示資料表中目前選取資料列的電子郵件地址。
- `{{ui.employeesTable.selectedRow["department"]}}` 使用 ID `employeesTable` 顯示資料表中目前所選資料列的部門。

範例：建立自訂資料欄

您可以根據基礎資料動作、自動化或表達式傳回的資料，將自訂資料欄新增至資料表。您可以使用現有的資料欄值和 JavaScript 表達式來建立新的資料欄。

範例：

- `{{currentRow.quantity * currentRow.unitPrice}}` 透過將數量和單價資料欄相乘，建立顯示總價的新資料欄。
- `{{new Date(currentRow.orderDate).toLocaleDateString()}}` 建立新的資料欄，以更易讀的格式顯示訂單日期。
- `{{currentRow.firstName + ' ' + currentRow.lastName + ' (' + currentRow.email + ')'}}` 建立新的資料欄，顯示每一列的完整名稱和電子郵件地址。

範例：自訂資料欄顯示值：

您可以透過設定資料欄映射的欄位，自訂資料表資料欄內 Value 欄位的顯示值。這可讓您將自訂格式或轉換套用至顯示的資料。

範例：

- `{{ currentRow.rating >= 4 ? '##'.repeat(currentRow.rating) : currentRow.rating }}` 根據每一列的評分值顯示星星表情符號。
- `{{ currentRow.category.toLowerCase().replace(/\b\w/g, c => c.toUpperCase()) }}` 顯示類別值，其中每個單字為每列大寫。
- `{{ currentRow.status === 'Active' ? '# Active' : '# Inactive' }}`：根據每一列的狀態值顯示彩色圓圈表情符號和文字。

資料列層級按鈕動作

`{{currentRow.columnName}}` 或 `{{currentRow["Column Name"]}}` 您可以使用這些表達式，在資料列層級動作內傳遞參考的資料列內容，例如使用所選資料列的資料導覽至另一個頁面，或使用資料列的資料觸發自動化。

範例：

- 如果您在資料列動作欄中有編輯按鈕，您可以將 `{{currentRow.orderId}}` 做為參數傳遞，以導覽至具有所選訂單 ID 的訂單編輯頁面。
- 如果您在資料列動作欄中有刪除按鈕，您可以傳遞 `{{currentRow.customerName}}` 至自動化，在刪除訂單之前傳送確認電子郵件給客戶。
- 如果您在資料列動作欄中有檢視詳細資訊按鈕，您可以傳遞 `{{currentRow.employeeId}}` 至自動化，以記錄檢視訂單詳細資訊的員工。

透過利用這些表達式欄位和資料列層級動作功能，您可以建立高度自訂的互動式資料表，根據您的特定需求顯示和操作資料。此外，您可以連接資料列層級動作與應用程式中的其他元件或自動化，實現無縫的資料流程和功能。

詳細資訊

詳細資訊元件旨在顯示特定記錄或項目的詳細資訊。它提供專用空間來呈現與單一實體或資料列相關的完整資料，因此非常適合顯示深入詳細資訊或協助資料輸入和編輯任務。

詳細資訊屬性

詳細資訊元件與其他元件共用數個常見屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

詳細資訊元件也有特定的屬性和組態選項，包括 Fields、Layout 和 Expressions。

版面配置

配置區段可讓您自訂詳細資訊元件中欄位的配置和呈現。您可以設定選項，例如：

- 資料欄數：指定要顯示欄位的資料欄數。
- 欄位排序：拖放欄位以重新排序其外觀。
- 間距和對齊：調整元件內欄位的間距和對齊。

表達式和範例

詳細資訊元件提供各種表達式欄位，可讓您動態參考和顯示元件內的資料。這些表達式可讓您建立自訂和互動式 Detail 元件，以與您應用程式的資料和邏輯無縫連線。

範例：參考資料

`{{ui.details.data[0]?."colName"}}`：此表達式可讓您參考資料陣列中第一個項目（索引 0）名為 "colName" 的資料欄值，該資料陣列連接到 ID 為 "details" 的詳細資訊元件。將 "colName" 取代之為您要參考的資料欄的實際名稱。例如，下列表達式會顯示連線至「詳細資訊」元件的資料陣列中第一個項目的「customerName」資料欄的值：

```
{{ui.details.data[0]?."customerName"}}
```

Note

當 Detail 元件與所參考的資料表位於相同頁面上，而且您想要在 Detail 元件中顯示資料表第一列的資料時，此表達式很有用。

範例：條件式轉譯

`{{ui.table1.selectedRow["colName"]}}`：如果 ID 資料表 1 資料表中的所選資料列具有名為 *colName* 的資料欄資料，則此表達式會傳回 true。它可以用來根據資料表選取的資料列是否為空，有條件地顯示或隱藏 Detail 元件。

範例：

您可以在 Detail 元件的 Visible if 屬性中使用此表達式，根據資料表中選取的資料列來有條件地顯示或隱藏它。

```
{{ui.table1.selectedRow["customerName"]}}
```

如果此表達式評估為 true (### 1 元件中選取的資料列具有 *customerName* 資料欄的值)，則會顯示 Detail 元件。如果表達式評估為 false（即選取的資料列空白或沒有 "customerName" 的值），則詳細資訊元件將會隱藏。

範例：條件式顯示

`{{(ui.Component.value === "green" ? "#" : ui.Component.value === "yellow" ? "#" : ui.detail1.data?.[0]?.CustomerStatus)}}`：此表達式會根據元件或資料欄位的值，有條件地顯示表情符號。

明細：

- `ui.Component.value`：參考具有 ID 元件之##的值。
- `=== "green"`：檢查元件的值是否等於字串 "green"。
- `? "#"`：如果條件為 true，會顯示綠色圓圈表情符號。
- `: ui.Component.value === "yellow" ? "#"`：如果第一個條件為 false，會檢查元件的值是否等於字串 "yellow"。
- `? "#"`：如果第二個條件為 true，會顯示黃色方形表情符號。
- `: ui.detail1.data?.[0]?.CustomerStatus`：如果兩個條件都是 false，則會參考資料陣列中第一個項目的 "CustomerStatus" 值，該資料陣列連接到 ID 為 "detail1" 的詳細資訊元件。

此表達式可用來根據 Detail 元件中的元件或資料欄位的值顯示表情符號或特定值。

指標

指標元件是一種視覺化元素，以類似卡片的格式顯示關鍵指標或資料點。它旨在提供簡潔且視覺上吸引人的方式，以呈現重要資訊或效能指標。

指標屬性

指標元件與其他元件共用數個共同屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

趨勢

指標的趨勢功能可讓您針對顯示的指標，顯示效能或隨時間變化的視覺化指標。

趨勢值

此欄位可讓您指定應該用來判斷趨勢方向和大小的值或表達式。一般而言，這是代表特定期間內變更或效能的值。

範例：

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue}}
```

此表達式會從連線至「salesMetrics」指標的資料中的第一個項目擷取month-over-month營收值。

正面趨勢

此欄位可讓您輸入運算式，以定義正面趨勢的條件。表達式應評估為 true 或 false。

範例：

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue > 0}}
```

此表達式會檢查month-over-month營收值是否大於 0，表示正趨勢。

負面趨勢

此欄位可讓您輸入定義負趨勢條件的表達式。表達式應評估為 true 或 false。

範例：

```
{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue < 0}}
```

此表達式會檢查month-over-month營收值是否小於 0，表示負趨勢。

顏色列

此切換可讓您啟用或停用彩色長條的顯示，以視覺化方式指出趨勢狀態。

顏色列範例：

範例：銷售指標趨勢

- 趨勢值：{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue}}
- 正面趨勢：{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue > 0}}
- 負面趨勢：{{ui.salesMetrics.data?.[0]?.monthOverMonthRevenue < 0}}
- 顏色列：已啟用

範例：庫存指標趨勢

- 趨勢值：{{ui.inventoryMetrics.data?.[0]?.currentInventory - ui.inventoryMetrics.data?.[1]?.currentInventory}}

- 正面趨勢：`{{ui.inventoryMetrics.data?.[0]?.currentInventory > ui.inventoryMetrics.data?.[1]?.currentInventory}}`
- 負面趨勢：`{{ui.inventoryMetrics.data?.[0]?.currentInventory < ui.inventoryMetrics.data?.[1]?.currentInventory}}`
- Color Bbar：已啟用

範例：客戶滿意度趨勢

- 趨勢值：`{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore}}`
- 正面趨勢：`{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore >= 8}}`
- 負面趨勢：`{{ui.customerSatisfactionMetrics.data?.[0]?.npsScore < 7}}`
- 顏色列：已啟用

透過設定這些趨勢相關屬性，您可以建立指標元件，為顯示的指標提供效能或隨時間變化的視覺化呈現。

透過利用這些表達式，您可以建立高度自訂的互動式指標元件，以動態方式參考和顯示資料，讓您展示應用程式中的關鍵指標、效能指標和資料驅動視覺化。

指標表達式範例

在屬性面板中，您可以輸入運算式以顯示標題、主要值、次要值和值字幕，以動態顯示值。

範例：參考主要值

`{{ui.metric1.primaryValue}}`：此表達式可讓您參考指標元件的主要值，以及相同頁面中其他元件或表達式的 ID `## 1`。

範例：`{{ui.salesMetrics.primaryValue}}`會顯示 *salesMetrics* 指標元件的主要值。

範例：參考次要值

`{{ui.metric1.secondaryValue}}`：此表達式可讓您使用相同頁面中其他元件或表達式的 ID `## 1` 來參考指標元件的次要值。

範例：`{{ui.revenueMetrics.secondaryValue}}`會顯示 *revenueMetrics* 指標元件的次要值。

範例：參考資料

`{{ui.metric1.data}}`：此表達式可讓您使用相同頁面中其他元件或表達式的 ID **## 1** 來參考指標元件的資料。

範例：`{{ui.kpiMetrics.data}}` 會參考連線至 *kpiMetrics* 指標元件的資料。

範例：顯示特定資料值：

`{{ui.metric1.data?.[0]?.id}}`：此表達式是如何使用 ID **## 1** 顯示資料中特定資訊片段的範例。當您想要顯示資料中第一個項目的特定屬性時，此功能非常有用。

明細：

- `ui.metric1`：參考 ID **## 1** 的指標元件。
- `data`：參考連接到該元件的資訊或資料集。
- `?.[0]`：表示該資料集的第一個項目或項目。
- `?.id`：顯示第一個項目或項目的 *ID* 值或識別符。

範例：`{{ui.orderMetrics.data?.[0]?.orderId}}` 會顯示連線至 *orderMetrics* 指標元件的資料中第一個項目的 *orderId* 值。 *orderMetrics*

範例：顯示資料長度

`{{ui.metric1.data?.length}}`：此表達式示範如何使用 ID **## 1** 在連線至指標元件的資料中顯示長度（項目數量）。當您想要顯示資料中的項目數量時，此功能非常有用。

明細：

- `ui.metric1.data`：參考連接到元件的資料集。
- `?.length`：存取該資料集中的項目或項目總數或數量。

範例：`{{ui.productMetrics.data?.length}}` 會顯示連線至 *productMetrics* 指標元件的資料中的項目數量。

延伸器

延伸器元件是一種動態元件，可讓您根據提供的資料來源產生和顯示元素集合。它旨在促進在應用程式的使用者介面中建立清單、網格或重複模式。幾個範例使用案例包括：

- 顯示帳戶中每個使用者的卡片
- 顯示包含影像的產品清單，以及將影像新增至購物車的按鈕
- 顯示使用者可存取的檔案清單

延伸器元件會區分自己與具有豐富內容的資料表元件。資料表元件具有嚴格的資料列和資料欄格式。延伸器可以更靈活地顯示您的資料。

延伸器屬性

延伸器元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

除了常見的屬性之外，延伸器元件還具有下列其他屬性和組態選項。

項目範本

項目範本是一個容器，您可以在其中定義要針對資料來源中每個項目重複的結構和元件。您可以將其他元件拖放到此容器中，例如文字、影像、按鈕，或任何其他您需要代表每個項目的元件。

在項目範本中，您可以使用 格式的表達式來參考目前項目的屬性或值 `{{currentItem.propertyName}}`。

例如，如果您的資料來源包含 `itemName` 屬性，您可以使用 `{{currentItem.itemName}}` 來顯示目前項目的項目名稱 (s)。

版面配置

配置區段可讓您設定延伸器元件中重複元素的配置。

方向

- 清單：在單一欄中垂直排列重複的元素。
- 網格：使用多個資料欄在網格配置中配置重複的元素。

每頁的資料列數

指定清單配置中每個頁面要顯示的列數。分頁適用於溢位指定資料列數的項目。

每頁資料欄和資料列數（網格）

- 欄：指定網格配置中的欄數。

- 每頁列數：指定網格配置中每頁顯示的列數。分頁適用於溢位指定網格維度的項目。

表達式和範例

Repeater 元件提供各種表達式欄位，可讓您動態參考和顯示元件內的資料。這些表達式可讓您建立自訂和互動式的 Repeater 元件，無縫連線至應用程式的資料和邏輯。

範例：參考項目

- `{{currentItem.propertyName}}`：參考項目範本中目前項目的屬性或值。
- `{{ui.repeaterID[index]}}`：依索引參考延伸器元件中的特定項目。

範例：轉譯產品清單

- 來源：選取##實體做為資料來源。
- 項目範本：新增容器元件，其中包含文字元件以顯示產品名稱 (`{{currentItem.productName}}`) 和影像元件以顯示產品影像 (`{{currentItem.productImageUrl}}`)。
- 配置：將設定為 Rows per Page, OrientationList 並視需要調整。

範例：產生使用者頭像的網格

- 來源：使用表達式來產生使用者資料的陣列（例如 `[{name: 'John', avatarUrl: '...'}, {...}, {...}]`）。
- 項目範本：新增映像元件，並將其 Source 屬性設定為 `{{currentItem.avatarUrl}}`。
- 配置：將 Orientation 設定為 Grid，指定 Columns 和 的數量 Rows per Page，並視需要調整 Space Between 和 Padding。

透過使用 Repeater 元件，您可以建立動態和資料驅動的使用者介面，簡化渲染元素集合的程序，並減少手動重複或硬式編碼的需求。

表格

Form 元件旨在擷取使用者輸入，並促進應用程式中的資料輸入任務。它提供結構化配置來顯示輸入欄位、下拉式清單、核取方塊和其他表單控制項，允許使用者無縫輸入或修改資料。您可以在表單元件內部巢狀其他元件，例如資料表。

表單屬性

表單元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

產生表單

產生表單功能可根據選取的資料來源自動填入表單欄位，讓您輕鬆快速建立表單欄位。這可以在建置需要顯示大量欄位的表單時節省時間和精力。

若要使用產生表單功能：

1. 在表單元件的屬性中，找到產生表單區段。
2. 選取您要用來產生表單欄位的資料來源。這可以是實體、工作流程或應用程式中可用的任何其他資料來源。
3. 表單欄位會根據選取的資料來源自動產生，包括欄位標籤、類型和資料映射。
4. 檢閱產生的欄位並進行任何必要的自訂，例如新增驗證規則或變更欄位順序。
5. 一旦您對表單組態感到滿意，請選擇提交，將產生的欄位套用至表單元件。

當您的應用程式中有明確定義的資料模型或一組實體，需要擷取使用者輸入時，產生表單功能特別有用。透過自動產生表單欄位，您可以節省時間，並確保應用程式表單之間的一致性。

使用產生表單功能後，您可以進一步自訂表單元件的配置、動作和表達式，以符合您的特定需求。

表達式和範例

如同其他元件，您可以使用表達式來參考和顯示表單元件中的資料。例如：

- `{{ui.userForm.data.email}}`：參考從資料來源連線至 ID 為 `userForm` 的 Form 元件的 email 欄位值。

Note

[常見元件屬性](#) 如需常見屬性的更多表達式範例，請參閱。

透過設定這些屬性並利用表達式，您可以建立自訂且互動式的表單元件，無縫整合應用程式的資料來源和邏輯。這些元件可用來擷取使用者輸入、顯示預先填入的資料，以及根據表單提交或使用者互動觸發動作。

Stepflow

Stepflow 元件旨在引導使用者完成應用程式中的多步驟程序或工作流程。它提供結構化且直覺式的界面來呈現一系列的步驟，每個步驟都有自己的一組輸入、驗證和動作。

Stepflow 元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Actions。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

Stepflow 元件具有其他屬性和組態選項，例如 Step Navigation、Validation 和 Expressions。

AI 元件

Gen AI

Gen AI 元件是一種分組容器，用於將元件及其隨附的邏輯分組，以便使用應用程式工作室內的聊天透過 AI 輕鬆編輯它們。當您使用聊天建立元件時，它們會分組到 Gen AI 容器中。如需編輯或使用此元件的詳細資訊，請參閱 [建置或編輯您的應用程式](#)。

文字和數字元件

文字輸入

文字輸入元件可讓使用者在應用程式中輸入和提交文字資料。它提供簡單且直覺的方式擷取使用者輸入，例如名稱、地址或任何其他文字資訊。

- `{{ui.inputTextID.value}}`：傳回輸入欄位中提供的值。
- `{{ui.inputTextID.isValid}}`：傳回輸入欄位中所提供值的有效性。

文字

文字元件用於顯示應用程式中的文字資訊。它可用來顯示靜態文字、動態值或從表達式產生的內容。

文字區域

文字區域元件旨在擷取使用者的多行文字輸入。它提供較大的輸入欄位區域，讓使用者輸入較長的文字項目，例如描述、備註或註解。

- `{{ui.textAreaID.value}}`：傳回文字區域中提供的值。
- `{{ui.textAreaID.isValid}}`：傳回文字區域中所提供值的有效性。

Email

電子郵件元件是專門的輸入欄位，旨在從使用者擷取電子郵件地址。它可以強制執行特定的驗證規則，以確保輸入的值符合正確的電子郵件格式。

- `{{ui.emailID.value}}`：傳回電子郵件輸入欄位中提供的值。
- `{{ui.emailID.isValid}}`：傳回電子郵件輸入欄位中所提供值的有效性。

密碼

密碼元件是專門為使用者設計的輸入欄位，用於輸入敏感資訊，例如密碼或 PIN 碼。它會遮罩輸入的字元，以維護隱私權和安全性。

- `{{ui.passwordID.value}}`：傳回密碼輸入欄位中提供的值。
- `{{ui.passwordID.isValid}}`：傳回密碼輸入欄位中所提供值的有效性。

搜尋

搜尋元件為使用者提供專用輸入欄位，以執行搜尋查詢或在應用程式中填入的資料內輸入搜尋詞彙。

- `{{ui.searchID.value}}`：傳回搜尋欄位中提供的值。

Phone

電話元件是一種輸入欄位，專為從使用者擷取電話號碼或其他聯絡資訊而量身打造。它可以包含特定的驗證規則和格式選項，以確保輸入的值符合正確的電話號碼格式。

- `{{ui.phoneID.value}}`：傳回電話輸入欄位中提供的值。
- `{{ui.phoneID.isValid}}`：傳回電話輸入欄位中所提供值的有效性。

Number

Number 元件是專門為使用者輸入數值而設計的輸入欄位。它可以強制執行驗證規則，以確保輸入的值是指定範圍或格式內的有效數字。

- `{{ui.numberID.value}}`：傳回數字輸入欄位中提供的值。
- `{{ui.numberID.isValid}}`：傳回數字輸入欄位中所提供值的有效性。

Currency

貨幣元件是專門的輸入欄位，用於擷取貨幣值或金額。它可以包含格式選項，以顯示貨幣符號、小數分隔符號，以及強制執行貨幣輸入特定的驗證規則。

- `{{ui.currencyID.value}}`：傳回貨幣輸入欄位中提供的值。
- `{{ui.currencyID.isValid}}`：傳回貨幣輸入欄位中所提供值的有效性。

詳細資訊對

詳細資訊對元件用於以結構化且可讀取的格式顯示鍵值對或相關資訊對。它通常用於呈現與特定項目或實體相關聯的詳細資訊或中繼資料。

選擇元件

交換器

切換元件是一種使用者介面控制項，可讓使用者在兩種狀態或選項之間切換，例如開/關、true/false 或 enabled/disabled。它提供目前狀態的視覺化呈現，並允許使用者按一下或點選即可進行變更。

切換群組

切換群組元件是個別切換控制項的集合，可讓使用者從預先定義的集合中選取一或多個選項。它提供已選取和未選取選項的視覺化呈現，讓使用者更容易了解可用選項並與之互動。

切換群組表達式欄位

- `{{ui.switchGroupID.value}}`：傳回字串陣列，其中包含應用程式使用者啟用的每個交換器的值。

核取方塊群組

核取方塊群組元件會向使用者顯示一組核取方塊，讓他們可以同時選取多個選項。當您想要讓使用者能夠從選項清單中選擇一或多個項目時，此功能非常有用。

核取方塊群組表達式欄位

- `{{ui.checkboxGroupID.value}}`：傳回字串陣列，其中包含應用程式使用者所選取之每個核取方塊的值。

無線電群組

無線電群組元件是一組選項按鈕，可讓使用者從多個互斥選項中選取單一選項。它可確保一次只能選擇一個選項，為使用者提供清晰且明確的方式進行選擇。

無線電群組表達式欄位

下列欄位可用於表達式。

- `{{ui.radioGroupID.value}}`：傳回應用程式使用者所選取選項按鈕的值。

單一選取

單一選取元件為使用者提供選項清單，他們可以從中選取單一項目。它通常用於使用者需要從預先定義的一組選項中進行選擇的情況，例如選取類別、位置或偏好設定。

單一選取表達式欄位

- `{{ui.singleSelectID.value}}`：傳回應用程式使用者所選取清單項目的值。

多選

多選取元件類似於單一選取元件，但允許使用者從選項清單中同時選取多個選項。當使用者需要從預先定義的選項集中進行多個選擇時，例如選取多個標籤、興趣或偏好設定時，此功能非常有用。

多選取表達式欄位

- `{{ui.multiSelectID.value}}`：傳回字串陣列，其中包含應用程式使用者選取之每個清單項目的值。

按鈕與導覽元件

應用程式工作室提供各種按鈕和導覽元件，可讓使用者觸發動作並在應用程式中導覽。

按鈕元件

可用的按鈕元件包括：

- 按鈕
- 加框按鈕

- 圖示按鈕
- 文字按鈕

這些按鈕元件共用下列常見屬性：

內容

- 按鈕標籤：要在按鈕上顯示的文字。

Type

- 按鈕：標準按鈕。
- 概述：概述樣式的按鈕。
- 圖示：具有圖示的按鈕。
- 文字：純文字按鈕。

大小

按鈕的大小。可能值為 Small、Medium 及 Large。

圖示

您可以從要顯示在按鈕上的各種圖示中選取，包括：

- 信封已關閉
- 鐘
- 個人
- 漢堡選單
- 搜尋
- Info Circled
- 齒輪
- 左年紋
- Chevron 右側
- 水平點
- 垃圾桶

- 編輯
- Check
- Close (關閉)
- 首頁
- Plus

觸發

按一下按鈕時，您可以設定要觸發的一或多個動作。可用的動作類型為：

- 基本
 - 執行元件動作：在元件中執行特定動作。
 - 導覽：導覽至另一個頁面或檢視。
 - 叫用資料動作：觸發與資料相關的動作，例如建立、更新或刪除記錄。
- Advanced (進階)
 - JavaScript：執行自訂 JavaScript 程式碼。
 - 調用自動化：啟動現有的自動化或工作流程。

JavaScript 動作按鈕屬性

選取JavaScript動作類型，以在按一下按鈕時執行自訂 JavaScript 程式碼。

來源碼

在 Source code欄位中，您可以輸入您的 JavaScript 表達式或函數。例如：

```
return "Hello World";
```

這只會在按一下按鈕Hello World時傳回字串。

條件：如果

您也可以提供布林表達式，以判斷是否應執行 JavaScript 動作。此標籤使用以下語法：

```
{{ui.textinput1.value !== ""}}
```

在此範例中，只有在textinput1元件的值不是空字串時，才會執行 JavaScript 動作。

透過使用這些進階觸發選項，您可以建立高度自訂的按鈕行為，直接與應用程式的邏輯和資料整合。這可讓您擴展按鈕的內建功能，並根據您的特定需求量身打造使用者體驗。

Note

請務必徹底測試您的 JavaScript 動作，以確保它們如預期般運作。

超連結

Hyperlink 元件提供可點選的連結，用於導覽至外部 URLs 或內部應用程式路由。

超連結屬性

內容

- 超連結標籤：顯示為超連結標籤的文字。

URL

超連結的目的地 URL，可以是外部網站或內部應用程式路由。

觸發

按一下超連結時，您可以設定要觸發的一或多個動作。可用的動作類型與按鈕元件的動作類型相同。

日期和時間元件

Date

日期元件允許使用者選取和輸入日期。

Date 元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Validation。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

除了常見的屬性之外，日期元件還具有下列特定屬性：

日期屬性

格式

- YYYY/MM/DD、DD/MM/YYYY、YYYY/MM/DD、YYYY/DD/MM、MM/DD、DD/MM：應顯示日期的格式。

Value

- YYYY-MM-DD：日期值儲存在內部的格式。

最短日期

- YYYY-MM-DD：可選取的最小日期。

Note

此值必須符合的格式YYYY-MM-DD。

日期上限

- YYYY-MM-DD：可選取的最大日期。

Note

此值必須符合的格式YYYY-MM-DD。

行事曆類型

- 1 個月、2 個月：要顯示的行事曆 UI 類型。

停用日期

- 來源：應停用之日期的資料來源。例如：無、表達式。
- 停用日期：決定應停用哪些日期的表達式，例如：
 - `{{currentRow.column}}`：停用符合此表達式評估的目標日期。
 - `{{new Date(currentRow.dateColumn) < new Date("2023-01-01")}}`：停用 2023 年 1 月 1 日之前的日期
 - `{{new Date(currentRow.dateColumn).getDay() === 0 || new Date(currentRow.dateColumn).getDay() === 6}}`：停用週末。

Behavior (行為)

- 如果顯示：決定日期元件可見性的表達式。
- 停用，如果：決定是否應停用日期元件的表達式。

驗證

驗證區段可讓您定義日期輸入的其他規則和限制條件。透過設定這些驗證規則，您可以確保使用者輸入的日期值符合您應用程式的特定需求。您可以新增下列類型的驗證：

- 必要：此切換可確保使用者在提交表單之前必須輸入日期值。
- 自訂：您可以使用 JavaScript 表達式建立自訂驗證規則。例如：

```
{{new Date(ui.dateInput.value) < new Date("2023-01-01")}}
```

此表達式會檢查輸入的日期是否早於 2023 年 1 月 1 日。如果條件為 true，驗證將會失敗。

您也可以提供自訂驗證訊息，以便在不符合驗證時顯示：

```
"Validation not met. The date must be on or after January 1, 2023."
```

透過設定這些驗證規則，您可以確保使用者輸入的日期值符合您應用程式的特定需求。

表達式和範例

Date 元件提供下列表達式欄位：

- `{{ui.dateID.value}}`：傳回使用者輸入的日期值，格式為 YYYY-MM-DD。

時間

時間元件允許使用者選取和輸入時間值。透過設定時間元件的各種屬性，您可以建立符合應用程式特定需求的時間輸入欄位，例如限制可選取的時間範圍、停用特定時間，以及控制元件的可見性和互動性。

時間屬性

時間元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Validation。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

除了常見的屬性之外，時間元件還具有下列特定屬性：

時間間隔

- 5 分鐘、10 分鐘、15 分鐘、20 分鐘、25 分鐘、30 分鐘、60 分鐘：可用於選取時間的間隔。

Value

- HH : MM AA：時間值儲存在內部的格式。

Note

此值必須符合 的格式HH:MM AA。

預留位置

- 行事曆設定：當時間欄位為空時顯示的預留位置文字。

最短時間

- HH : MM AA：可選取的最短時間。

Note

此值必須符合 的格式HH:MM AA。

最長時間

- HH : MM AA：可選取的最長時間。

Note

此值必須符合 的格式HH:MM AA。

停用時間

- 來源：應停用時間的資料來源（例如 None、Expression）。
- 停用時間：決定應停用哪些時間的表達式，例如 `{{currentRow.column}}`。

停用時間組態

您可以使用已停用時間區段來指定哪些時間值應該無法選取。

來源

- 無：不會停用任何時間。
- 表達式：您可以使用 JavaScript 表達式來判斷應該停用哪些時間，例如 `{{currentRow.column}}`。

範例表達式：

```
{{currentRow.column === "Lunch Break"}}
```

此表達式會停用目前資料列的「午餐休息」資料欄為 true 的任何時間。

透過設定這些驗證規則和停用的時間表達式，您可以確保使用者輸入的時間值符合您應用程式的特定需求。

Behavior (行為)

- 如果顯示：決定時間元件可見性的表達式。
- 停用，如果：決定是否應停用時間元件的表達式。

驗證

- 必要：確保使用者在提交表單之前必須輸入時間值的切換。
- 自訂：可讓您使用 JavaScript 表達式建立自訂驗證規則。

自訂驗證訊息：不符合自訂驗證時要顯示的訊息。

例如：

```
{{ui.timeInput.value === "09:00 AM" || ui.timeInput.value === "09:30 AM"}}
```

此表達式會檢查輸入的時間是上午 9 : 00 還是上午 9 : 30。如果條件為 true，驗證將會失敗。

您也可以提供自訂驗證訊息，以便在不符合驗證時顯示：

```
Validation not met. The time must be 9:00 AM or 9:30 AM.
```

表達式和範例

Time 元件提供下列表達式欄位：

- `{{ui.timeID.value}}`：傳回使用者以 HH : MM AA 格式輸入的時間值。

範例：時間值

- `{{ui.timeID.value}}`：傳回使用者輸入的時間值，格式為 HH:MM AA。

範例：時間比較

- `{{ui.timeInput.value > "10:00 AM"}}`：檢查時間值是否大於上午 10 : 00。
- `{{ui.timeInput.value < "05:00 pM"}}`：檢查時間值是否小於下午 05 : 00。

日期範圍

日期範圍元件允許使用者選取和輸入日期範圍。透過設定日期範圍元件的各種屬性，您可以建立符合應用程式特定需求的日期範圍輸入欄位，例如限制可選取的日期範圍、停用特定日期，以及控制元件的可見性和互動性。

日期範圍屬性

日期範圍元件與其他元件共用數個常見的屬性，例如 Name、Source 和 Validation。如需這些屬性的詳細資訊，請參閱 [常見元件屬性](#)。

除了常見屬性之外，日期範圍元件還具有下列特定屬性：

格式

- MM/DD/YYYY：應顯示日期範圍的格式。

開始日期

- YYYY-MM-DD：可選取做為範圍開始的最小日期。

Note

此值必須符合 的格式YYYY-MM-DD。

結束日期

- YYYY-MM-DD：可選取為範圍結尾的最大日期。

Note

此值必須符合 的格式YYYY-MM-DD。

預留位置

- 行事曆設定：當日期範圍欄位空白時顯示的預留位置文字。

最短日期

- YYYY-MM-DD：可選取的最低日期。

Note

此值必須符合 的格式YYYY-MM-DD。

日期上限

- YYYY-MM-DD：可選取的最大日期。

Note

此值必須符合 的格式YYYY-MM-DD。

行事曆類型

- 1 個月：要顯示的行事曆 UI 類型。例如，單月。
- 2 個月：要顯示的行事曆 UI 類型。例如，兩個月。

選取強制性天數

- 0：必須在日期範圍內選取的必要天數。

停用日期

- 來源：應停用之日期的資料來源（例如，無、表達式、實體或自動化）。
- 停用日期：決定應停用哪些日期的表達式，例如 `{{currentRow.column}}`。

驗證

驗證區段可讓您定義日期範圍輸入的其他規則和限制條件。

表達式和範例

日期範圍元件提供下列表達式欄位：

- `{{ui.dateRangeID.startDate}}`：傳回所選範圍的開始日期，格式為 YYYY-MM-DD。
- `{{ui.dateRangeID.endDate}}`：傳回所選範圍的結束日期，格式為 YYYY-MM-DD。

範例：計算日期差異

- `{{(new Date(ui.dateRangeID.endDate) - new Date(ui.dateRangeID.startDate)) / (1000 * 60 * 60 * 24)}}` 計算開始和結束日期之間的天數。

範例：根據日期範圍的條件可見性

- `{{new Date(ui.dateRangeID.startDate) < new Date("2023-01-01") || new Date(ui.dateRangeID.endDate) > new Date("2023-12-31")}}` 檢查選取的日期範圍是否在 2023 年之外。

範例：根據目前資料列資料停用日期

- `{{currentRow.isHoliday}}` 停用目前資料列中「isHoliday」資料欄為 true 的日期。
- `{{new Date(currentRow.dateColumn) < new Date("2023-01-01")}}` 根據目前資料列中的 "dateColumn"，停用 2023 年 1 月 1 日之前的日期。
- `{{new Date(currentRow.dateColumn).getDay() === 0 || new Date(currentRow.dateColumn).getDay() === 6}}` 根據目前資料列中的 "dateColumn" 停用週末。

自訂驗證

- `{{new Date(ui.dateRangeID.startDate) > new Date(ui.dateRangeID.endDate)}}` 檢查開始日期是否晚於結束日期，這會使自訂驗證失敗。

媒體元件

應用程式工作室提供多種元件，用於內嵌和顯示應用程式中的各種媒體類型。

iFrame 內嵌

iFrame 內嵌元件可讓您使用 iFrame 在應用程式中內嵌外部 Web 內容或應用程式。

iFrame 內嵌屬性

URL

Note

此元件中顯示的媒體來源必須允許在您應用程式的內容安全設定中。如需詳細資訊，請參閱[檢視或更新應用程式的內容安全設定](#)。

您要嵌入的外部內容或應用程式的 URL。

版面配置

- 寬度：iFrame 的寬度，以百分比 (%) 或固定像素值（例如 300px）指定。
- 高度：iFrame 的高度，以百分比 (%) 或固定像素值指定。

S3 上傳

S3 上傳元件可讓使用者將檔案上傳至 Amazon S3 儲存貯體。透過設定 S3 上傳元件，您可以讓使用者輕鬆地將檔案上傳到應用程式的 Amazon S3 儲存貯體，然後利用應用程式邏輯和使用者介面中上傳的檔案資訊。

Note

請記得確保具備必要的許可和 Amazon S3 儲存貯體組態，以支援應用程式的檔案上傳和儲存需求。

S3 上傳屬性

S3 組態

- 連接器：選取要用於檔案上傳的預先設定 Amazon S3 連接器。
- 儲存貯體：上傳檔案的 Amazon S3 儲存貯體。
- 資料夾：Amazon S3 儲存貯體中存放檔案的資料夾。
- 檔案名稱：上傳檔案的命名慣例。

檔案上傳組態

- 標籤：顯示在檔案上傳區域上方的標籤或指示。
- 描述：檔案上傳的其他指示或資訊。
- 檔案類型：允許上傳的檔案類型。例如：影像、文件或影片。
- 大小：可上傳之個別檔案的大小上限。
- 按鈕標籤：檔案選擇按鈕上顯示的文字。
- 按鈕樣式：檔案選取按鈕的樣式。例如，概述或填充。
- 按鈕大小：檔案選擇按鈕的大小。

驗證

- 檔案數量上限：一次可上傳的檔案數量上限。
- 檔案大小上限：每個個別檔案允許的檔案大小上限。

觸發

- 成功時：檔案上傳成功時要觸發的動作。
- 失敗時：檔案上傳失敗時要觸發的動作。

S3 上傳表達式欄位

S3 上傳元件提供下列表達式欄位：

- `{{ui.s3uploadID.files}}`：傳回已上傳的檔案陣列。
- `{{ui.s3uploadID.files[0]?.size}}`：傳回指定索引的檔案大小。
- `{{ui.s3uploadID.files[0]?.type}}`：傳回指定索引的檔案類型。
- `{{ui.s3uploadID.files[0]?.nameOnly}}`：在指定的索引傳回沒有副檔名尾碼的檔案名稱。
- `{{ui.s3uploadID.files[0]?.nameWithExtension}}`：傳回指定索引中副檔名尾碼為 的檔案名稱。

表達式和範例

範例：存取上傳的檔案

- `{{ui.s3uploadID.files.length}}`：傳回已上傳的檔案數目。
- `{{ui.s3uploadID.files.map(f => f.name).join(', ')}}`：傳回已上傳檔案名稱的逗號分隔清單。
- `{{ui.s3uploadID.files.filter(f => f.type.startsWith('image/'))}}`：僅傳回已上傳的影像檔案陣列。

範例：驗證檔案上傳

- `{{ui.s3uploadID.files.some(f => f.size > 5 * 1024 * 1024)}}`：檢查任何上傳的檔案是否超過 5 MB。
- `{{ui.s3uploadID.files.every(f => f.type === 'image/png')}}`：檢查所有上傳的檔案是否為 PNG 映像。
- `{{ui.s3uploadID.files.length > 3}}`：檢查是否已上傳超過 3 個檔案。

範例：觸發動作

- `{{ui.s3uploadID.files.length > 0 ? 'Upload Successful' : 'No files uploaded'}}`：如果至少已上傳一個檔案，則會顯示成功訊息。
- `{{ui.s3uploadID.files.some(f => f.type.startsWith('video/')) ? triggerVideoProcessing() : null}}`：如果已上傳任何影片檔案，則觸發影片處理自動化。
- `{{ui.s3uploadID.files.map(f => f.url)}}`：擷取上傳檔案URLs，可用於顯示或進一步處理檔案。

這些表達式可讓您存取上傳的檔案、驗證檔案上傳，以及根據檔案上傳結果觸發動作。透過使用這些表達式，您可以在應用程式的檔案上傳功能中建立更動態且智慧的行為。

Note

使用 S33 上傳元件的 ID 取代 `s3uploadID`。

PDF 檢視器元件

PDF 檢視器元件可讓使用者檢視應用程式中的 PDF 文件並與之互動。App Studio 支援 PDF 來源的這些不同輸入類型，PDF 檢視器元件可讓您靈活地將 PDF 文件整合到您的應用程式，無論是來自靜態 URL、內嵌資料 URI 還是動態產生的內容。

PDF 檢視器屬性

來源

Note

此元件中顯示的媒體來源必須允許在您應用程式的內容安全設定中。如需詳細資訊，請參閱[檢視或更新應用程式的內容安全設定](#)。

PDF 文件的來源，可以是表達式、實體、URL 或自動化。

表達式

使用 表達式動態產生 PDF 來源。

實體

將 PDF 檢視器元件連接至包含 PDF 文件的資料實體。

URL

指定 PDF 文件的 URL。

URL

您可以輸入指向您要顯示之 PDF 文件的 URL。這可以是公有 Web URL 或您自己的應用程式中的 URL。

範例：`https://example.com/document.pdf`

資料 URI

資料 URI 是一種在應用程式中內嵌小型資料檔案（例如影像或 PDFs 精簡方式。PDF 文件編碼為 base64 字串，並直接包含在元件的組態中。

Blob 或 ArrayBuffer

您也可以提供 PDF 文件做為 Blob 或 ArrayBuffer 物件，可讓您動態產生或擷取應用程式中各種來源的 PDF 資料。

自動化

將 PDF 檢視器元件連接至提供 PDF 文件的自動化。

動作

- 下載：新增允許使用者下載 PDF 文件的按鈕或連結。

版面配置

- 寬度：PDF 檢視器的寬度，以百分比 (%) 或固定像素值（例如 600px）指定。
- 高度：指定為固定像素值的 PDF 檢視器高度。

影像檢視器

影像檢視器元件可讓使用者檢視應用程式內的影像檔案，並與之互動。

影像檢視器屬性

來源

Note

此元件中顯示的媒體來源必須允許在您的應用程式的內容安全設定中。如需詳細資訊，請參閱[檢視或更新應用程式的內容安全設定](#)。

- **實體**：將映像檢視器元件連接至包含映像檔案的資料實體。
- **URL**：指定映像檔案的 URL。
- **表達式**：使用表達式動態產生影像來源。
- **自動化**：將映像檢視器元件連接到提供映像檔案的自動化。

Alt 文字

影像的替代文字描述，用於可存取性目的。

版面配置

- **影像擬合**：決定影像應如何調整大小並在元件中顯示。例如，Contain、Cover 或 Fill。
- **寬度**：影像檢視器元件的寬度，以百分比 (%) 或固定像素值（例如 300px）指定。
- **高度**：影像檢視器元件的高度，指定為固定像素值。
- **背景**：可讓您設定影像檢視器元件的背景顏色或影像。

自動化和動作：定義應用程式的商業邏輯

自動化是您定義應用程式商業邏輯的方式。自動化的主要元件包括：啟動自動化的觸發、一或多個動作的序列、用來將資料傳遞至自動化的輸入參數，以及輸出。

主題

- [自動化概念](#)
- [建立、編輯和刪除自動化](#)
- [新增、編輯和刪除自動化動作](#)

- [自動化動作參考](#)

自動化概念

以下是在 App Studio 中使用自動化定義和設定應用程式的商業邏輯時應注意的一些概念和術語。

自動化

自動化是您定義應用程式商業邏輯的方式。自動化的主要元件包括：啟動自動化的觸發、一或多個動作的序列、用於將資料傳遞至自動化的輸入參數，以及輸出。

動作

自動化動作通常稱為 動作，是構成自動化之邏輯的個別步驟。每個動作都會執行特定任務，無論是傳送電子郵件、建立資料記錄、叫用 Lambda 函數或呼叫 APIs。動作會從動作程式庫新增至自動化，並可分組為條件式陳述式或迴圈。

自動化輸入參數

自動化輸入參數是動態輸入值，您可以從元件傳入自動化，使它們靈活且可重複使用。將參數視為自動化的變數，而不是硬式編碼值到自動化中，您可以定義參數並視需要提供不同的值。參數可讓您在每次執行時搭配不同的輸入使用相同的自動化。

模擬輸出

有些動作會使用連接器與外部資源或服務互動。使用預覽環境時，應用程式不會與外部服務互動。若要測試在預覽環境中使用連接器的動作，您可以使用模擬輸出來模擬連接器的行為和輸出。模擬輸出是使用 JavaScript 設定，結果會存放在動作的結果中，就像連接器的回應存放在已發佈的應用程式中一樣。

透過使用模擬，您可以使用預覽環境透過自動化測試各種案例及其對其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不滿意的路徑，而無需透過連接器呼叫外部服務。

自動化輸出

自動化輸出用於將值從一個自動化傳遞到應用程式的其他資源，例如元件或其他自動化。自動化輸出會設定為表達式，而表達式可以傳回靜態值或從自動化參數和動作計算的動態值。根據預設，自動化不會傳回任何資料，包括自動化內動作的結果。

如何使用自動化輸出的幾個範例：

- 您可以設定自動化輸出以傳回陣列，並傳遞該陣列以填入資料元件。
- 您可以使用自動化來計算值，並將該值傳遞至多個其他自動化，做為集中和重複使用商業邏輯的方式。

觸發

觸發會決定自動化執行的時間和條件。有些觸發範例On click適用於按鈕和On select文字輸入。元件類型會決定該元件的可用觸發條件清單。觸發條件會新增至[元件](#)，並在應用程式工作室中設定。

建立、編輯和刪除自動化

內容

- [建立自動化](#)
- [檢視或編輯自動化屬性](#)
- [刪除自動化](#)

建立自動化

使用下列程序在 App Studio 應用程式中建立自動化。建立之後，必須透過編輯自動化屬性並將其新增動作來設定自動化。

建立自動化


1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇自動化索引標籤。
3. 如果您沒有自動化，請選擇 + 在畫布中新增自動化。否則，在左側自動化選單中，選擇 + 新增。
4. 將建立新的自動化，您可以開始編輯其屬性，或新增和設定動作來定義應用程式的商業邏輯。

檢視或編輯自動化屬性

使用下列程序來檢視或編輯自動化屬性。

檢視或編輯自動化屬性

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇自動化索引標籤。

3. 在左側自動化選單中，選擇要檢視或編輯屬性的自動化，以開啟右側屬性選單。
 4. 在屬性功能表中，您可以檢視下列屬性：
 - 自動化識別符：自動化的唯一名稱。若要編輯，請在文字欄位中輸入新的識別符。
 - 自動化參數：自動化參數用於將動態值從應用程式的 UI 傳遞至自動化和資料動作。若要新增參數，請選擇 + 新增。選擇鉛筆圖示以變更參數的名稱、描述或類型。若要移除參數，請選擇垃圾桶圖示。
-  Tip

您也可以直接從畫布新增自動化參數。
5. 您可以透過新增和設定動作來定義自動化的功能。如需動作的詳細資訊，請參閱 [新增、編輯和刪除自動化動作](#) 和 [自動化動作參考](#)。

刪除自動化

使用下列程序來刪除 App Studio 應用程式中的自動化。

刪除自動化

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇自動化索引標籤。
3. 在左側自動化選單中，選擇您要刪除之自動化的省略項目選單，然後選擇刪除。或者，您可以從自動化的右側屬性功能表中選擇垃圾桶圖示。
4. 在確認對話方塊中，選擇 Delete (刪除)。

新增、編輯和刪除自動化動作

自動化動作通常稱為動作，是構成自動化之邏輯的個別步驟。每個動作都會執行特定任務，無論是傳送電子郵件、建立資料記錄、叫用 Lambda 函數或呼叫 APIs。動作會從動作程式庫新增至自動化，並可分組為條件式陳述式或迴圈。

內容

- [新增自動化動作](#)
- [檢視和編輯自動化動作屬性](#)
- [刪除自動化動作](#)

新增自動化動作

使用下列程序將動作新增至 App Studio 應用程式中的自動化。

新增自動化動作

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇自動化索引標籤。
3. 在左側自動化選單中，選擇您要新增動作的自動化。
4. 在右側動作選單中，選擇您要新增的動作，或將動作拖放到畫布中。建立動作之後，您可以選擇動作來設定動作屬性以定義動作的功能。如需動作屬性和設定它們的詳細資訊，請參閱 [自動化動作參考](#)。

檢視和編輯自動化動作屬性

使用下列程序來檢視或編輯 App Studio 應用程式中自動化動作的屬性。

檢視或編輯自動化動作屬性

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇自動化索引標籤。
3. 在左側自動化選單中，選擇您要檢視或編輯屬性的動作。或者，您可以在檢視包含它的自動化時，選擇畫布中的動作。
4. 您可以在右側的屬性功能表中檢視或編輯動作屬性。每個動作類型的動作屬性都不同。如需動作屬性和設定它們的詳細資訊，請參閱 [自動化動作參考](#)。

刪除自動化動作

使用下列程序，從 App Studio 應用程式的自動化中刪除動作。

刪除自動化動作

1. 如有必要，請導覽至您應用程式的應用程式工作室。

2. 選擇自動化索引標籤。
3. 在左側自動化選單中，選擇包含您要刪除之動作的自動化。
4. 在畫布中，選擇您要刪除動作中的垃圾桶圖示，然後選擇刪除。

自動化動作參考

以下是 App Studio 中使用的自動化動作參考文件。

自動化動作通常稱為 動作，是構成自動化之邏輯的個別步驟。每個動作都會執行特定任務，無論是傳送電子郵件、建立資料記錄、叫用 Lambda 函數或呼叫 APIs。動作會從動作程式庫新增至自動化，並可分組為條件式陳述式或迴圈。

如需有關建立和設定自動化及其動作的資訊，請參閱 中的主題 [自動化和動作：定義應用程式的商業邏輯](#)。

叫用 API

叫用 HTTP REST API 請求。建置器可以使用此動作，透過 APIs 將請求從 App Studio 傳送至其他系統或服務。例如，您可以使用它來連線至第三方系統或原生應用程式，以存取業務關鍵資料，或叫用無法由專用 App Studio 動作呼叫的 API 端點。

如需 REST APIs 的詳細資訊，請參閱 [什麼是 RESTful API ?](#)。

Properties

連接器

用於此動作提出之 API 請求的 Connector。連接器下拉式清單僅包含下列類型的連接器：API Connector 和 OpenAPI Connector。根據連接器的設定方式，它可以包含重要資訊，例如登入資料和預設標頭或查詢參數。

如需 API 連接器的詳細資訊，包括使用 API Connector 和 之間的比較 OpenAPI Connector，請參閱 [連線至第三方服務](#)。

API 請求組態屬性

從屬性面板選擇設定 API 請求，以開啟請求組態對話方塊。如果選取 API 連接器，對話方塊將包含連接器資訊。

方法：API 呼叫的方法。可能的值如下：

- DELETE：刪除指定的資源。
- GET：擷取資訊或資料。
- HEAD：僅擷取不含內文的回應標頭。
- POST：提交要處理的資料。
- PUSH：提交要處理的資料。
- PATCH：部分更新指定的資源。

路徑：資源的相對路徑。

標頭：任何要與 API 請求一起傳送的鍵/值對形式的標頭。如果選取連接器，其設定的標頭會自動新增且無法移除。無法編輯設定的標頭，但您可以透過新增具有相同名稱的另一個標頭來覆寫它們。

查詢參數：任何要與 API 請求一起傳送的鍵/值對形式的查詢參數。如果選取連接器，其設定的查詢參數會自動新增，且無法編輯或移除。

內文：要與 JSON 格式 API 請求一起傳送的資訊。GET 請求沒有內文。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

叫用 AWS

從 AWS 服務叫用操作。這是呼叫 AWS 服務或操作的一般動作，如果所需的 AWS 服務或操作沒有專用動作，則應使用。

Properties

服務

包含要執行之操作 AWS 的服務。

作業

要執行的操作。

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

執行指定操作時的 JSON 輸入。如需設定 AWS 操作輸入的詳細資訊，請參閱 [適用於 JavaScript 的 AWS SDK](#)。

調用 Lambda

叫用現有的 Lambda 函數。

Properties

連接器

此動作執行的 Lambda 函數所使用的連接器。設定的連接器應使用適當的登入資料來設定，以存取 Lambda 函數和其他組態資訊，例如包含 Lambda 函數 AWS 的區域。如需設定 Lambda 連接器的詳細資訊，請參閱 [步驟 3：建立 Lambda 連接器](#)。

函數名稱

要執行的 Lambda 函數名稱。請注意，這是函數名稱，而不是函數 ARN (Amazon Resource Name)。

函數事件

要做為事件承載傳遞至 Lambda 函數的鍵值對。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 results 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

循環

重複執行巢狀動作，逐一查看項目清單，一次一個項目。例如，將 [建立記錄](#) 動作新增至迴圈動作以建立多個記錄。

迴圈動作可以巢狀在其他迴圈或條件動作中。迴圈動作會依序執行，而不是平行執行。迴圈中每個動作的結果只能存取相同迴圈反覆運算中的後續動作。它們無法在迴圈外部或在迴圈的不同反覆運算中存取。

Properties

來源

要反覆查看的項目清單，一次一個項目。來源可能是先前動作的結果，或是您可以使用 JavaScript 表達式提供的字串、數字或物件靜態清單。

範例

下列清單包含來源輸入的範例。

- 先前動作的結果：`{{results.actionName.data}}`
- 號碼清單：`{{[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}}`
- 字串清單：`{{["apple", "banana", "orange", "grape", "kiwi"]}}`
- 計算值：`{{params.actionName.split("\n")}}`

目前項目名稱

變數的名稱，可用於參考正在反覆運算的目前項目。目前的項目名稱是可設定的，因此您可以巢狀兩個或多個迴圈，並從每個迴圈存取變數。例如，如果您使用兩個迴圈來循環國家/地區和城市，您可以設定和參考 `currentCountry` 和 `currentCity`。

條件

根據執行自動化時評估的一或多個指定邏輯條件的結果來執行動作。條件動作由下列元件組成：

- 條件欄位，用於提供評估為 `true` 或的 JavaScript 表達式 `false`。
- `true` 分支，其中包含條件評估為 時執行的動作 `true`。
- `false` 分支，其中包含條件評估為 時執行的動作 `false`。

將動作拖曳至條件動作，以新增至 true 和 false 分支。

Properties

條件

執行動作時要評估的 JavaScript 表達式。

建立記錄

在現有的 App Studio 實體中建立一個記錄。

Properties

實體

要在其中建立記錄的實體。選取實體之後，必須將值新增至實體的欄位，才能建立記錄。欄位的類型，以及在實體中定義欄位為必要或選用。

更新記錄

更新 App Studio 實體中的現有記錄。

Properties

實體

包含要更新之記錄的實體。

條件

定義動作要更新哪些記錄的條件。您可以分組條件來建立一個邏輯陳述式。您可以將群組或條件與 AND 或 OR 陳述式結合。

欄位

條件指定的記錄中要更新的欄位。

值

指定欄位中要更新的值。

刪除記錄

從 App Studio 實體刪除記錄。

Properties

實體

包含要刪除之記錄的實體。

條件

定義動作要刪除哪些記錄的條件。您可以分組條件來建立一個邏輯陳述式。您可以將群組或條件與 AND 或 OR 陳述式結合。

叫用資料動作

使用選用參數執行資料動作。

Properties

資料動作

動作要執行的資料動作。

Parameters

資料動作要使用的資料動作參數。資料動作參數用於傳送用作資料動作輸入的值。您可以在設定自動化動作時新增資料動作參數，但必須在資料索引標籤中編輯。

進階設定

Invoke data action 動作包含下列進階設定：

- 頁面大小：每個查詢中要擷取的記錄數目上限。預設值為 500，最大值為 3000。
- 分頁字符：用於從查詢擷取其他記錄的字符。例如，如果 Page size 設定為 500，但有超過 500 筆記錄，則將分頁字符傳遞至後續查詢將擷取接下來的 500 筆記錄。如果沒有更多記錄或頁面，則不會定義字符。

Amazon S3：放置物件

使用 Amazon S3 PutObject 操作將金鑰（檔案路徑）識別的物件新增至指定的 Amazon S3 儲存貯體。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 PutObject 命令中使用的必要選項。選項如下：

Note

如需 Amazon S3 PutObject 操作的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 [PutObject](#)。

- 儲存貯體：要在其中放置物件的 Amazon S3 儲存貯體名稱。
- 金鑰：要放入 Amazon S3 儲存貯體之物件的唯一名稱。
- 內文：要放入 Amazon S3 儲存貯體的物件內容。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，以模擬預覽環境中連接器的行為，用於測試目的。此程式碼片段會存放在動作的 results 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon S3：刪除物件

使用 Amazon S3 DeleteObject 操作，從指定的 Amazon S3 儲存貯體刪除由金鑰（檔案路徑）識別的物件。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 DeleteObject 命令中使用的必要選項。選項如下：

Note

如需 Amazon S3 DeleteObject 操作的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 [DeleteObject](#)。

- 儲存貯體：從中刪除物件的 Amazon S3 儲存貯體名稱。
- 金鑰：要從 Amazon S3 儲存貯體刪除之物件的唯一名稱。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 results 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon S3：取得物件

使用 Amazon S3 GetObject 操作從指定的 Amazon S3 儲存貯體擷取由金鑰（檔案路徑）識別的物件。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 GetObject 命令中使用的必要選項。選項如下：

Note

如需 Amazon S3 `GetObject` 操作的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 [GetObject](#)。

- 儲存貯體：要從中擷取物件的 Amazon S3 儲存貯體名稱。
- 金鑰：要從 Amazon S3 儲存貯體擷取之物件的唯一名稱。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon S3：列出物件

使用 Amazon S3 `ListObjects` 操作列出指定 Amazon S3 儲存貯體中的物件。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 `ListObjects` 命令中使用的必要選項。選項如下：

Note

如需 Amazon S3 `ListObjects` 操作的詳細資訊，請參閱《Amazon Simple Storage Service API 參考》中的 [ListObjects](#)。

- 儲存貯體：要從中列出物件的 Amazon S3 儲存貯體名稱。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Textract：分析文件

使用 Amazon Textract `AnalyzeDocument` 操作來分析輸入文件，找出偵測到的項目之間的關係。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 `AnalyzeDocument` 命令中使用的請求內容。選項如下：

Note

如需 Amazon Textract `AnalyzeDocument` 操作的詳細資訊，請參閱《Amazon Textract 開發人員指南》中的 [AnalyzeDocument](#)。

- 文件/S3Object/儲存貯體：Amazon S3 儲存貯體的名稱。如果檔案透過 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/名稱：輸入文件的檔案名稱。如果檔案透過 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/版本：如果 Amazon S3 儲存貯體已啟用版本控制，您可以指定物件的版本。如果檔案透過 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- `FeatureTypes`：要執行的分析類型清單。有效值為：TABLES、FORMS、QUERIES、SIGNATURES 和 LAYOUT。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Textract：分析費用

使用 Amazon Textract `AnalyzeExpense` 操作來分析文字之間財務相關關係的輸入文件。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 `AnalyzeExpense` 命令中使用的請求內容。選項如下：

Note

如需 Amazon Textract `AnalyzeExpense` 操作的詳細資訊，請參閱《Amazon Textract 開發人員指南》中的 [AnalyzeExpense](#)。

- 文件/S3Object/儲存貯體：Amazon S3 儲存貯體的名稱。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/名稱：輸入文件的檔案名稱。如果檔案透過 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/版本：如果 Amazon S3 儲存貯體已啟用版本控制，您可以指定物件的版本。如果檔案透過 S3 上傳元件傳遞至動作，則此參數可以保留空白。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Textract : 分析 ID

使用 Amazon Textract AnalyzeID 操作來分析身分文件以取得相關資訊。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 AnalyzeID 命令中使用的請求內容。選項如下：

Note

如需 Amazon Textract AnalyzeID 操作的詳細資訊，請參閱《Amazon Textract 開發人員指南》中的 [AnalyzeID](#)。

- 文件/S3Object/儲存貯體：Amazon S3 儲存貯體的名稱。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/名稱：輸入文件的檔案名稱。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/版本：如果 Amazon S3 儲存貯體已啟用版本控制，您可以指定物件的版本。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Textract：偵測文件文字

使用 Amazon Textract `DetectDocumentText` 操作來偵測輸入文件中構成一行文字的文字行和單字。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 `DetectDocumentText` 命令中使用的請求內容。選項如下：

Note

如需 Amazon Textract `DetectDocumentText` 操作的詳細資訊，請參閱《Amazon Textract 開發人員指南》中的 [DetectDocumentText](#)。

- 文件/S3Object/儲存貯體：Amazon S3 儲存貯體的名稱。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/名稱：輸入文件的檔案名稱。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。
- 文件/S3Object/版本：如果 Amazon S3 儲存貯體已啟用版本控制，您可以指定物件的版本。如果檔案使用 S3 上傳元件傳遞至動作，則此參數可以保留空白。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Bedrock : GenAI 提示

使用 [Amazon Bedrock InvokeModel](#) 操作，使用動作屬性中提供的提示和推論參數來執行推論。動作可以產生文字、影像和內嵌。

Properties

連接器

用於此動作所執行操作的連接器。若要成功使用此動作，必須使用 Amazon Bedrock Runtime 做為服務來設定連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

模型

Amazon Bedrock 用來處理請求的基礎模型。如需 Amazon Bedrock 中模型的詳細資訊，請參閱 [《Amazon Bedrock 使用者指南》中的 Amazon Bedrock 基礎模型資訊](#)。

輸入類型

傳送至 Amazon Bedrock 模型的輸入類型。可能的值為文字、文件和影像。如果輸入類型無法選取，則設定的模型可能不支援。

使用者提示

要傳送至要處理之 Amazon Bedrock 模型的提示，以產生回應。您可以輸入靜態文字，或從應用程式的另一個部分傳入輸入，例如使用參數的元件、自動化中的先前動作或其他自動化。下列範例示範如何從元件或先前動作傳入值：

- 若要使用參數從元件傳遞值：`{{params.paramName}}`
- 若要從上一個動作傳遞值：`{{results.actionName}}`

系統提示 (Claude 模型)

Amazon Bedrock 模型在處理請求時要使用的系統提示。系統提示用於為 Claude 模型提供內容、指示或指導方針。

請求設定

設定各種請求設定和模型推論參數。您可以進行下列設定：

- 溫度：Amazon Bedrock 模型在處理請求時要使用的溫度。溫度決定 Bedrock 模型輸出的隨機性或創造力。溫度越高，回應的創意和分析就越少。可能的值為 [0-10]。
- 最大權杖：限制 Amazon Bedrock 模型輸出的長度。
- TopP：在核取樣中，模型會以遞減的機率順序計算每個後續字符所有選項的累積分佈，並在達到 TopP 指定的特定機率時將其切斷。您應該更改溫度或 TopP，但不能同時更改兩者
- 停止序列：導致模型停止處理請求和產生輸出的序列。

如需詳細資訊，請參閱 [《Amazon Bedrock 使用者指南》中的基礎模型的推論請求參數和回應欄位](#)。

停止序列

輸入 Amazon Bedrock Guardrail ID 和版本。護欄用於根據您的使用案例和負責任的 AI 政策實作保護措施。如需詳細資訊，請參閱 [《Amazon Bedrock 使用者指南》中的使用 Amazon Bedrock Guardrails 停止模型中的有害內容](#)。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

Amazon Bedrock：調用模型

使用 [Amazon Bedrock InvokeModel](#) 操作，使用請求內文中提供的提示字元和推論參數來執行推論。您可以使用模型推論來產生文字、影像和內嵌。

Properties

連接器

用於此動作所執行操作的連接器。若要成功使用此動作，必須使用 Amazon Bedrock Runtime 做為服務來設定連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 `InvokeModel` 命令中使用的請求內容。

Note

如需 Amazon Bedrock `InvokeModel` 操作的詳細資訊，包括範例命令，請參閱《Amazon Bedrock API 參考》中的 [InvokeModel](#)。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

JavaScript

執行自訂 JavaScript 函數以傳回指定的值。

Important

App Studio 不支援使用第三方或自訂 JavaScript 程式庫。

Properties

來源碼

動作要執行的 JavaScript 程式碼片段。

i Tip

您可以使用 AI，透過執行下列步驟來協助您產生 JavaScript：

1. 選擇展開圖示以開啟展開的 JavaScript 編輯器。
2. (選用)：啟用修改程式碼切換來修改任何現有的 JavaScript。否則，AI 將取代任何現有的 JavaScript。
3. 在產生 JavaScript 中，描述您想要使用 JavaScript 做什麼，例如：**Add two numbers**。
4. 選擇傳送圖示以產生您的 JavaScript。

叫用自動化

執行指定的自動化。

Properties

調用自動化

動作要執行的自動化。

傳送電子郵件

使用 Amazon SES SendEmail 操作來傳送電子郵件。

Properties

連接器

用於此動作所執行操作的連接器。設定的連接器應設定適當的登入資料來執行操作，以及其他組態資訊，例如包含操作中參考之任何資源 AWS 的區域。

Configuration

要在 SendEmail 命令中使用的請求內容。選項如下：

i Note

如需 Amazon SES SendEmail 操作的詳細資訊，請參閱《Amazon Simple Email Service API 參考》中的 [SendEmail](#)。

模擬輸出

動作不會與預覽環境中的外部服務或資源互動。模擬輸出欄位用於提供 JSON 表達式，模擬預覽環境中連接器的行為，以供測試之用。此程式碼片段會存放在動作的 `results` 映射中，就像在即時環境中發佈的應用程式的連接器回應一樣。

透過此欄位，您可以測試各種案例及其對自動化內其他動作的影響，例如模擬不同的結果值、錯誤案例、邊緣案例或不快樂的路徑，而無需透過連接器與外部服務通訊。

實體和資料動作：設定應用程式的資料模型

實體是 App Studio 中的資料表。實體會直接與資料來源中的資料表互動。實體包括描述其中資料的欄位、尋找和傳回資料的查詢，以及將實體欄位連接至資料來源資料欄的映射。

主題

- [設計資料模型時的最佳實務](#)
- [在 App Studio 應用程式中建立實體](#)
- [在 App Studio 應用程式中設定或編輯實體](#)
- [刪除實體](#)
- [AWS App Studio 中的受管資料實體](#)

設計資料模型時的最佳實務

使用下列最佳實務，在 App Studio 中建立強大、可擴展且安全的關聯式資料模型，AWS 用於符合您應用程式需求的 App Studio 應用程式，並確保資料基礎設施的長期可靠性和效能。

- 選擇正確的 AWS 資料服務：根據您的需求，選擇適當的 AWS 資料服務。例如，對於線上交易處理 (OLTP) 應用程式，您可以考慮資料庫 (DB)，例如 Amazon Aurora，這是支援 MySQL 和 PostgreSQL 等各種資料庫引擎的雲端原生、關聯式和全受管資料庫服務。如需 App Studio 支援的完整 Aurora 版本清單，請參閱 [連線至 Amazon Aurora](#)。另一方面，對於線上分析處理 (OLAP) 使用案例，請考慮使用 Amazon Redshift，這是雲端資料倉儲，可讓您針對非常大型的資料集執行複雜的查詢。這些查詢通常需要時間（許多秒）才能完成，因此 Amazon Redshift 較不適合需要低延遲資料存取的 OLTP 應用程式。
- 可擴展性設計：將未來成長和可擴展性納入考量，規劃您的資料模型。選擇適當的資料服務和資料庫執行個體類型和組態（例如佈建容量）時，請考慮預期資料量、存取模式和效能需求等因素。
 - 如需使用 Aurora Serverless 擴展的詳細資訊，請參閱 [Aurora Serverless V2 的效能和擴展](#)。

- 標準化您的資料：遵循資料庫標準化的原則，將資料備援降至最低並改善資料完整性。這包括建立適當的資料表、定義主索引鍵和外部索引鍵，以及建立實體之間的關係。在 App Studio 中，從一個實體查詢資料時，您可以透過在查詢上指定子 join 句，從另一個實體擷取相關資料。
- 實作適當的索引：識別最重要的查詢和存取模式，並建立適當的索引以最佳化效能。
- 利用 AWS 資料服務功能：利用您選擇的 AWS 資料服務所提供的功能，例如自動備份、多可用區域部署和自動軟體更新。
- 保護您的資料：實作強大的安全措施，例如 IAM (AWS Identity and Access Management) 政策、建立具有資料表和結構描述限制許可的資料庫使用者，以及強制執行靜態和傳輸中的加密。
- 監控和最佳化效能：持續監控資料庫的效能，並視需要進行調整，例如擴展資源、最佳化查詢或調校資料庫組態。
- 自動化資料庫管理：利用 Aurora Autoscaling、Aurora 的績效詳情和 AWS Database Migration Service 等 AWS 服務，自動化資料庫管理任務並降低營運開銷。
- 實作災難復原和備份策略：確保您擁有定義明確的備份和復原計劃，利用 Aurora 自動化備份、point-in-time 復原和跨區域複本組態等功能。
- 遵循 AWS 最佳實務和文件：隨時 up-to-date 掌握所選資料服務的最新 AWS 最佳實務、準則和文件，以確保您的資料模型和實作符合 AWS 建議。

如需每個 AWS 資料服務的詳細指引，請參閱下列主題：

- [Amazon Aurora 的最佳實務](#)
- [Amazon Aurora MySQL 的最佳實務](#)
- [Amazon Redshift 查詢效能調校](#)
- [在 Amazon DynamoDB 中查詢和掃描資料的最佳實務](#)

在 App Studio 應用程式中建立實體

在 App Studio 應用程式中建立實體有四種方法。下列清單包含每個方法、其優點，以及使用該方法建立和設定實體之說明的連結。

- [從現有資料來源建立實體](#)：從現有的資料來源資料表自動建立實體及其欄位，並將欄位映射至資料來源資料表資料欄。如果您有要在 App Studio 應用程式中使用的現有資料來源，則此選項為佳。
- [使用 App Studio 受管資料來源建立實體](#)：建立 App Studio 為您管理的實體和 DynamoDB 資料表。DynamoDB 資料表會在您更新實體時自動更新。使用此選項，您不需要手動建立、管理或連接第三方資料來源，或指定從實體欄位到資料表欄的映射。您應用程式的所有資料建模和組態都是在

App Studio 中完成。如果您不想管理自己的資料來源和 DynamoDB 資料表，且其功能足以用於您的應用程式，則此選項為佳。

- [建立空白實體](#)：從頭開始完全建立空實體。如果您沒有管理員建立的任何現有資料來源或連接器，而且您想要彈性設計應用程式的資料模型，而不受外部資料來源限制，則此選項是較佳的。您可以在建立之後將實體連線至資料來源。
- [使用 AI 建立實體](#)：根據指定的實體名稱產生實體、欄位、資料動作和範例資料。如果您知道應用程式的資料模型，但想要協助將其翻譯為實體，則此選項是較適合的。

從現有資料來源建立實體

使用資料來源中的資料表自動建立實體及其欄位，並將實體欄位映射至資料表的資料欄。如果您有要在 App Studio 應用程式中使用的現有資料來源，則此選項為佳。

1. 如有必要，請導覽至您的應用程式。
2. 選擇畫布頂端的資料索引標籤。
3. 如果您的應用程式中沒有實體，請選擇 + 建立實體。否則，在左側實體功能表中，選擇 + 新增。
4. 選取使用現有資料來源中的資料表。
5. 在連接器中，選取包含您要用來建立實體之資料表的連接器。
6. 在資料表中，選擇您要用來建立實體的資料表。
7. 選取建立資料動作核取方塊以建立資料動作。
8. 選擇 Create entity (建立實體)。您的實體現在已建立，您可以在左側實體面板中看到它。
9. 遵循中的程序來設定您的新實體在 [App Studio 應用程式中設定或編輯實體](#)。請注意，由於您的實體是使用現有的資料來源建立的，因此已建立一些屬性或資源，例如欄位、連線的資料來源和欄位映射。此外，如果您在建立期間選取建立資料動作核取方塊，您的實體將包含資料動作。

使用 App Studio 受管資料來源建立實體

建立由 App Studio 管理的受管實體和對應的 DynamoDB 資料表。雖然 DynamoDB 資料表存在於關聯 AWS 帳戶中，但在 App Studio 應用程式中對實體進行變更時，DynamoDB 資料表會自動更新。使用此選項，您不需要手動建立、管理或連接第三方資料來源，或指定從實體欄位到資料表欄的映射。如果您不想管理自己的資料來源和 DynamoDB 資料表，且其功能足以用於您的應用程式，則此選項為佳。如需受管實體的詳細資訊，請參閱 [AWS App Studio 中的受管資料實體](#)。

您可以在多個應用程式中使用相同的受管實體。如需說明，請參閱 [從現有資料來源建立實體](#)。

1. 如有必要，請導覽至您的應用程式。
2. 選擇畫布頂端的資料索引標籤。
3. 如果您的應用程式中沒有實體，請選擇 + 建立實體。否則，在左側實體功能表中，選擇 + 新增。
4. 選取建立 App Studio 受管實體。
5. 在實體名稱中，提供實體的名稱。
6. 在主索引鍵中，提供實體主索引鍵的名稱。主索引鍵是實體的唯一識別符，建立實體之後就無法變更。
7. 在主索引鍵資料類型中，選取實體主索引鍵的資料類型。建立實體之後，無法變更資料類型。
8. 選擇 Create entity (建立實體)。您的實體現在已建立，您可以在左側實體面板中看到它。
9. 遵循中的程序來設定您的新實體在 [App Studio 應用程式中設定或編輯實體](#)。請注意，由於您的實體是使用受管資料建立的，因此已建立一些屬性或資源，例如主索引鍵欄位和連線的資料來源。

建立空白實體

從頭開始建立完全空的實體。如果您沒有管理員建立的任何現有資料來源或連接器，則此選項為佳。建立空實體可提供彈性，因為您可以在 App Studio 應用程式中設計實體，而不受外部資料來源的限制。在您設計應用程式的資料模型並相應地設定實體之後，您仍然可以稍後將其連接到外部資料來源。

1. 如有必要，請導覽至您的應用程式。
2. 選擇畫布頂端的資料索引標籤。
3. 如果您的應用程式中沒有實體，請選擇 + 建立實體。否則，在左側實體功能表中，選擇 + 新增。
4. 選取建立實體。
5. 選擇 Create entity (建立實體)。您的實體現在已建立，您可以在左側實體面板中看到它。
6. 遵循中的程序來設定您的新實體在 [App Studio 應用程式中設定或編輯實體](#)。

使用 AI 建立實體

根據指定的實體名稱產生實體、欄位、資料動作和範例資料。如果您知道應用程式的資料模型，但想要協助將其翻譯為實體，則此選項是較適合的。

1. 如有必要，請導覽至您的應用程式。
2. 選擇畫布頂端的資料索引標籤。
3. 如果您的應用程式中沒有實體，請選擇 + 建立實體。否則，在左側實體功能表中，選擇 + 新增。

4. 選取使用 AI 建立實體。
5. 在實體名稱中，提供實體的名稱。此名稱用於產生您實體的欄位、資料動作和範例資料。
6. 選取建立資料動作核取方塊以建立資料動作。
7. 選擇產生實體。您的實體現在已建立，您可以在左側實體面板中看到它。
8. 遵循中的程序來設定您的新實體在 [App Studio 應用程式中設定或編輯實體](#)。請注意，由於您的實體是使用 AI 建立的，因此您的實體將已包含產生的欄位。此外，如果您在建立期間選取建立資料動作核取方塊，您的實體將包含資料動作。

在 App Studio 應用程式中設定或編輯實體

使用下列主題在 App Studio 應用程式中設定實體。

主題

- [編輯實體名稱](#)
- [新增、編輯或刪除實體欄位](#)
- [建立、編輯或刪除資料動作](#)
- [新增或刪除範例資料](#)
- [新增或編輯連線的資料來源和映射欄位](#)

編輯實體名稱

1. 如有必要，請導覽至您要編輯的實體。
2. 在組態索引標籤的實體名稱中，更新實體名稱，然後在文字方塊外選擇 **以儲存變更**。

新增、編輯或刪除實體欄位

Tip

您可以按 CTRL+Z 復原實體的最新變更。

1. 如有必要，請導覽至您要編輯的實體。
2. 在組態索引標籤的欄位中，您可以檢視實體欄位的資料表。實體欄位具有下列資料欄：

- **顯示名稱**：顯示名稱類似於資料表標頭或表單欄位，應用程式使用者可檢視。它可以包含空格和特殊字元，但實體內必須是唯一的。
 - **系統名稱**：系統名稱是程式碼中用來參考欄位的唯一識別符。映射至 Amazon Redshift 資料表中的資料欄時，必須符合 Amazon Redshift 資料表資料欄名稱。
 - **資料類型**：將儲存在此欄位內的資料類型，例如 Integer、Boolean 或 String。
3. 若要新增欄位：
 - a. 若要使用 AI 根據實體名稱和連線的資料來源產生欄位，請選擇產生更多欄位。
 - b. 若要新增單一欄位，請選擇 + 新增欄位。
 4. 若要編輯欄位：
 - a. 若要編輯顯示名稱，請在顯示名稱文字方塊中輸入所需的值。如果欄位的系統名稱尚未編輯，則會將其更新為顯示名稱的新值。
 - b. 若要編輯系統名稱，請在系統名稱文字方塊中輸入所需的值。
 - c. 若要編輯資料類型，請選擇資料類型下拉式功能表，然後從清單中選擇所需的類型。
 - d. 若要編輯欄位的屬性，請選擇欄位的齒輪圖示。下列清單詳細說明欄位屬性：
 - **必要**：如果您的資料來源需要欄位，請啟用此選項。
 - **主索引鍵**：如果欄位對應至資料來源中的主索引鍵，請啟用此選項。
 - **唯一**：如果此欄位的值必須是唯一的，請啟用此選項。
 - **使用資料來源預設值**：如果資料來源提供欄位的值，例如使用自動遞增或事件時間戳記，請啟用此選項。
 - **資料類型選項**：某些資料類型的欄位可以設定資料類型選項，例如最小值或最大值。
 5. 若要刪除欄位，請選擇您要刪除之欄位的垃圾桶圖示。

建立、編輯或刪除資料動作

資料動作用於應用程式，以對實體的資料執行動作，例如擷取所有記錄，或依 ID 擷取記錄。資料動作可用來尋找和傳回符合指定條件的資料，以便在資料表或詳細資訊檢視等元件中檢視。

內容

- [建立資料動作](#)
- [編輯或設定資料動作](#)
- [資料動作條件運算子和範例](#)

- [資料庫的條件運算子支援](#)
- [資料動作條件範例](#)
- [刪除資料動作](#)

建立資料動作

Tip

您可以按 CTRL+Z 復原實體的最新變更。

1. 如有必要，請導覽至您要為其建立資料動作的實體。
2. 選擇資料動作索引標籤。
3. 有兩種方法來建立資料動作：
 - （建議）若要使用 AI 來產生資料動作，根據您的實體名稱、欄位和連線的資料來源，請選擇產生資料動作。將產生下列動作：
 1. `getAll`：從實體擷取所有記錄。當您需要顯示記錄清單或一次對多個記錄執行操作時，此動作很有用。
 2. `getByID`：根據實體的唯一識別符 (ID 或主索引鍵) 擷取單一記錄。當您需要對特定記錄顯示或執行操作時，此動作非常有用。
 - 若要新增單一資料動作，請選擇 + 新增資料動作。
4. 若要檢視或設定新的資料動作，請參閱下一節：[編輯或設定資料動作](#)。

編輯或設定資料動作

1. 如有必要，請導覽至您要為其建立資料動作的實體。
2. 選擇資料動作索引標籤。
3. 在欄位中，設定要由查詢傳回的欄位。根據預設，會選取實體中的所有設定欄位。

您也可以執行下列步驟，將聯結新增至資料動作：

1. 選擇 + 新增加入以開啟對話方塊。
2. 在相關實體中，選取您要加入目前實體的實體。
3. 在別名中，選擇性地輸入相關實體的暫時別名名稱。

4. 在聯結類型中，選取所需的聯結類型。
5. 從每個實體中選取欄位，以定義聯結子句。
6. 選擇新增以建立聯結。

建立後，聯結會顯示在聯結區段中，讓要傳回的欄位下拉式清單中提供其他欄位。您可以新增多個聯結，包括跨實體的鏈結聯結。您也可以依加入的實體中的欄位進行篩選和排序。

若要刪除聯結，請選擇其旁邊的垃圾桶圖示。這將從該聯結中移除任何欄位，並使用這些欄位破壞任何相依聯結或限制條件。

4. 在條件中，新增、編輯或移除篩選查詢輸出的規則。您可以將規則組織成群組，並使用 AND 或 OR 陳述式將多個規則鏈結在一起。如需您可以使用之運算子的詳細資訊，請參閱 [資料動作條件運算子和範例](#)。
5. 在排序中，選擇屬性並選擇遞增或遞減順序，以設定查詢結果的排序方式。您可以選擇排序規則旁的垃圾桶圖示，以移除排序組態。
6. 在轉換結果中，您可以輸入自訂 JavaScript，在結果顯示或傳送至自動化之前修改或格式化結果。
7. 在輸出預覽中，根據設定的欄位、篩選條件、排序和 JavaScript 檢視查詢輸出的預覽資料表。

資料動作條件運算子和範例

您可以使用條件運算子來比較設定的表達式值與實體資料欄，以傳回資料庫物件的子集。您可以使用的運算子取決於資料欄的資料類型，以及實體連線的資料庫類型，例如 Amazon Redshift、Amazon Aurora 或 Amazon DynamoDB。

下列條件運算子可與所有資料庫服務搭配使用：

- = 和 !=：適用於所有資料類型（主索引鍵資料欄除外）。
- <=、<、>= 和 >：僅適用於數值欄。
- IS NULL 和 IS NOT NULL：用於比對具有 null 或空白值的資料欄。Null 值在每個資料庫中的解譯方式通常不同，但在 App Studio 中，NULL 運算子會比對並傳回連線資料庫資料表中具有 Null 值的記錄。

下列條件運算子只能在連線至支援它們的資料庫服務的實體中使用：

- LIKE 和 NOT LIKE (Redshift、Aurora)：用於在連線的資料庫中執行模式型查詢。LIKE 運算子提供搜尋功能的彈性，因為它會尋找並傳回符合指定模式的記錄。您可以使用符合模式內任何字元或字元

序列的萬用字元來定義模式。每個資料庫管理系統都有一組唯一的萬用字元，但最熱門的兩個是%代表任意數量的字元（包括 0）_，以及代表單一字元。

- Contains 和 Not Contains(DynamoDB)：用於執行區分大小寫的搜尋，以判斷欄值內是否找到指定的文字。
- Starts With 和 Not Starts With(DynamoDB)：用於執行區分大小寫的搜尋，以判斷指定的文字是否在資料欄值的開頭找到。

資料庫的條件運算子支援

下表顯示每個可連線至 App Studio 的資料庫支援哪些資料動作條件運算子。

	=, !=, <, >, <=, >=	LIKE、NIKE	包含、不包含	從開始，從開始	IS NULL、IS NOT NULL
DynamoDB	是	否	是	是	是
Aurora	是	是	否	否	是
Redshift	是	是	否	否	是

資料動作條件範例

請考慮下列資料庫資料表，其中包含具有 name、city 和 hireDate 欄位的多個項目。

name	城市	hireDate
亞當	西雅圖	2025-03-01
Adrienne	波士頓	2025-03-05
Bob	阿布奎基	2025-03-06
Carlos	芝加哥	2025-03-10
Caroline	NULL	2025-03-12
Rita	邁阿密	2025-03-15

現在，請考慮在 App Studio 中建立資料動作，以針對符合指定條件的項目傳回 name 欄位。下列清單包含條件範例，以及資料表針對每個項目傳回的值。

Note

這些範例已格式化為 SQL 範例，可能不會如在 App Studio 中所示，而是用來說明運算子的行為。

- WHERE name LIKE 'Adam' : 傳回 Adam。
- WHERE name LIKE 'A%' : 傳回 Adam和 Adrienne。
- WHERE name NOT LIKE 'B_B' : 傳回 Adam、Adrienne、Caroline、Carlos和 Rita。
- WHERE contains(name, 'ita') : 傳回 Rita。
- WHERE begins_with(name, 'Car') : 傳回 Carlos和 Caroline。
- WHERE city IS NULL : 傳回 Caroline。
- WHERE hireDate < "2025-03-06" : 傳回 Adam和 Adrienne。
- WHERE hireDate >= DateTime.now().toISODate() : 請注意，DateTime.now().toISODate()會傳回目前日期。在目前日期為 2025-03-10 的情況下，表達式會傳回 Carlos、Caroline和 Rita。

Tip

如需在表達式中比較日期和時間的詳細資訊，請參閱 [日期和時間](#)。

刪除資料動作

使用下列程序從 App Studio 實體刪除資料動作。

1. 如有必要，請導覽至您要刪除資料動作的實體。
2. 選擇資料動作索引標籤。
3. 針對您要刪除的每個資料動作，選擇編輯旁的下拉式選單，然後選擇刪除。
4. 在對話方塊中選擇確認。

新增或刪除範例資料

您可以將範例資料新增至 App Studio 應用程式中的實體。由於應用程式在發佈之前不會與外部服務通訊，因此範例資料可用於在預覽環境中測試您的應用程式和實體。

1. 如有必要，請導覽至您要編輯的實體。
2. 選擇範例資料索引標籤。
3. 若要產生範例資料，請選擇產生更多範例資料。
4. 若要刪除範例資料，請選取您要刪除之資料的核取方塊，然後按 Delete 或 Backspace 鍵。選擇儲存以儲存變更。

新增或編輯連線的資料來源和映射欄位

Tip

您可以按 CTRL+Z 復原實體的最新變更。

1. 如有必要，請導覽至您要編輯的實體。
2. 選擇連線索引標籤，以檢視或管理發佈應用程式時存放資料的實體與資料來源資料表之間的連線。連接資料來源資料表後，您可以將實體欄位映射至資料表的資料欄。
3. 在連接器中，選擇包含所需資料來源資料表連線的連接器。如需連接器的詳細資訊，請參閱 [使用連接器將 App Studio 連接至其他服務](#)。
4. 在資料表中，選擇您要用作實體資料來源的資料表。
5. 資料表顯示實體的欄位，以及其對應的資料來源資料欄。選擇自動映射以自動映射您的實體欄位與資料來源資料欄。您也可以選擇每個實體欄位下拉式清單中的資料來源欄，在資料表中手動對應欄位。

刪除實體

使用下列程序從 App Studio 應用程式刪除實體。

Note

從 App Studio 應用程式刪除實體並不會刪除連接的資料來源資料表，包括受管實體對應的 DynamoDB 資料表。資料來源資料表將保留在相關聯的 AWS 帳戶中，並且需要視需要從對應的服務中刪除。

刪除實體

1. 如有必要，請導覽至您的應用程式。
2. 選擇資料索引標籤。
3. 在左側實體選單中，選擇您要刪除之實體旁的省略符號選單，然後選擇刪除。
4. 檢閱對話方塊中的資訊，輸入 **confirm** 並選擇刪除以刪除實體。

AWS App Studio 中的受管資料實體

一般而言，您會在 App Studio 中設定連線至外部資料庫資料表的實體，而且您必須使用連線資料庫資料表中的資料欄來建立和對應每個實體欄位。當您變更資料模型時，必須更新外部資料庫資料表和實體，並且必須重新映射變更的欄位。雖然此方法具有彈性並可使用不同類型的資料來源，但需要更預先的規劃和持續的維護。

受管實體是 App Studio 為您管理整個資料儲存和組態程序的一種實體。當您建立受管實體時，會在相關聯的 AWS 帳戶中建立對應的 DynamoDB 資料表。這可確保內部安全且透明的資料管理 AWS。使用受管實體時，您可以在 App Studio 中設定實體的結構描述，對應的 DynamoDB 資料表也會自動更新。

在多個應用程式中使用受管實體

在 App Studio 應用程式中建立受管實體後，該實體就可以在其他 App Studio 應用程式中使用。這有助於透過提供要維護的單一基礎資源，為具有相同資料模型和結構描述的應用程式設定資料儲存。

在多個應用程式中使用受管實體時，必須使用建立受管實體的原始應用程式，對對應的 DynamoDB 資料表進行所有結構描述更新。對其他應用程式中的實體所做的任何結構描述變更都不會更新對應的 DynamoDB 資料表。

受管實體限制

主金鑰更新限制：您無法在實體建立後變更其主金鑰名稱或類型，因為這是 DynamoDB 中的破壞性變更，並會導致現有資料遺失。

重新命名資料欄：重新命名 DynamoDB 中的資料欄時，您實際上會建立新的資料欄，而原始資料欄會保留原始資料。原始資料不會自動複製到新資料欄或從原始資料欄刪除。您可以重新命名稱為系統名稱的受管實體欄位，但您將無法存取原始資料欄及其資料。重新命名顯示名稱沒有限制。

變更資料類型：雖然 DynamoDB 允許彈性在建立資料表後修改資料欄資料類型，但此類變更可能會嚴重影響現有資料，以及查詢邏輯和準確性。資料類型變更需要轉換所有現有資料以符合新格式，這對大型的作用中資料表來說非常複雜。此外，資料動作可能會傳回非預期的結果，直到資料遷移完成為止。您可以切換欄位的資料類型，但現有的資料不會遷移至新的資料類型。

排序資料欄：DynamoDB 可透過排序索引鍵啟用排序資料擷取。排序索引鍵必須與分割區索引鍵一起定義為複合主索引鍵的一部分。限制包括強制性排序索引鍵、限制在一個分割區內的排序，以及沒有跨分割區的全域排序。需要謹慎建立排序索引鍵的資料模型，才能避免熱分割區。我們不支援排序預覽里程碑。

聯結：DynamoDB 不支援聯結。資料表會依設計取消標準化，以避免昂貴的聯結操作。若要 one-to-many 關係的模型，子資料表包含參考父資料表主索引鍵的屬性。多資料表資料查詢涉及從父資料表查詢項目以擷取詳細資訊。在預覽里程碑中，我們不支援受管實體的原生加入。作為解決方法，我們將引進自動化步驟，以執行 2 個實體的資料合併。這與單一層級查詢非常類似。我們不支援排序預覽里程碑。

Env Stage：我們將允許發佈測試，但在兩個環境中使用相同的受管存放區

頁面和自動化參數

參數是 AWS App Studio 中的強大功能，用於在應用程式中的不同元件、頁面和自動化之間傳遞動態值。使用參數，您可以提供靈活且內容感知的體驗，讓您的應用程式更具回應性和個人化性。本文涵蓋兩種類型的參數：頁面參數和自動化參數。

主題

- [頁面參數](#)
- [自動化參數](#)

頁面參數

頁面參數是一種在頁面之間傳送資訊的方式，通常用於在 App Studio 應用程式中從一個頁面導覽至另一個頁面，以維護內容或傳遞資料。頁面參數通常由名稱和值組成。

頁面參數使用案例

頁面參數用於在 App Studio 應用程式的不同頁面和元件之間傳遞資料。對於下列使用案例特別有用：

1. 搜尋和篩選：當使用者在您的應用程式首頁上搜尋時，搜尋詞彙可以做為參數傳遞至結果頁面，使其僅顯示相關的篩選項目。例如，如果使用者搜尋####，則具有####值的參數可以傳遞至產品清單頁面。
2. 檢視項目詳細資訊：如果使用者按一下清單，例如產品，則該項目的唯一識別符可以做為參數傳遞至詳細資訊頁面。這可讓詳細資訊頁面顯示特定項目的所有資訊。例如，當使用者按一下耳機產品時，產品的唯一 ID 會做為參數傳遞至產品詳細資訊頁面。
3. 在頁面導覽中傳遞使用者內容：當使用者在頁面之間導覽時，參數可以傳遞重要的內容，例如使用者的位置、偏好的產品類別、購物車內容和其他設定。例如，當使用者瀏覽應用程式上的不同產品類別時，其位置和偏好的類別會保留為參數，以提供個人化且一致的體驗。
4. 深層連結：使用頁面參數來共用或將應用程式中特定頁面的連結加入書籤。
5. 資料動作：您可以建立接受參數值的資料動作，以根據傳遞的參數篩選和查詢資料來源。例如，在產品清單頁面上，您可以建立接受category參數的資料動作來擷取相關產品。

頁面參數安全性考量事項

雖然頁面參數提供了在頁面之間傳遞資料的強大方式，但您必須謹慎使用它們，因為如果未正確使用，它們可能會公開敏感資訊。以下是需要記住的重要安全考量：

1. 避免在 URLs 中公開敏感資料
 - a. 風險：URLs，包括資料動作參數，通常可見於伺服器日誌、瀏覽器歷史記錄和其他位置。因此，請務必避免在頁面參數值中公開敏感資料，例如使用者登入資料、個人身分識別資訊 (PII) 或任何其他機密資料。
 - b. 緩解：考慮使用可以安全地映射到敏感資料的識別符。例如，您可以傳遞隨機的唯一識別符，用來擷取使用者名稱或電子郵件，而不是傳遞使用者名稱或電子郵件地址做為參數。

自動化參數

自動化參數是 App Studio 中強大的功能，可透過傳遞各種來源的動態值，例如 UI、其他自動化或資料動作，來建立靈活且可重複使用的自動化。它們可做為預留位置，在執行自動化時以實際值取代，讓您每次使用相同的自動化搭配不同的輸入。

在自動化中，參數具有唯一的名稱，您可以使用參數變數來參考參數的值，後面接著參數的名稱，例如 `{{params.customerId}}`。

本文提供對自動化參數的深入了解，包括基本概念、用量和最佳實務。

自動化參數優點

自動化參數提供多種優點，包括下列清單：

1. 可重複使用性：透過使用參數，您可以建立可重複使用的自動化，這些自動化可以使用不同的輸入值進行自訂，讓您可以使用不同的輸入重複使用相同的自動化邏輯。
2. 彈性：您可以視需要定義參數並提供不同的值，讓您的自動化更具動態性和適應性，而不是硬式編碼的自動化值。
3. 問題分離：參數有助於將自動化邏輯與使用的特定值分開，進而提升程式碼組織和可維護性。
4. 驗證：每個參數都有資料類型，例如字串、數字或布林值，並在執行時間進行驗證。這可確保拒絕具有不正確資料類型的請求，而不需要自訂驗證碼。
5. 選用和必要參數：您可以將自動化參數指定為選用或必要。執行自動化時必須提供必要的參數，而選用參數可以有預設值或省略。此彈性可讓您建立更多樣化的自動化，以根據提供的參數處理不同的案例。

案例和使用案例

案例：擷取產品詳細資訊

假設您有一個自動化，可根據產品 ID 從資料庫擷取產品詳細資訊。此自動化可以有一個名為 `productId` 的參數。

`productId` 參數充當預留位置，您可以在執行自動化時填入實際的產品 ID 值。您可以定義 `productId` 參數，並在每次執行自動化時傳入不同的產品 ID 值，而不是將特定產品 ID 硬式編碼到自動化中。

您可以從元件的資料來源呼叫此自動化，並使用雙大括號語法傳遞所選產品的 ID 做為 `productId` 參數：`{{ui.productsTable.selectedRow.id}}`。如此一來，當使用者從資料表選取產品 (`ui.productsTable`) 時，自動化將透過傳遞所選列的 ID 做為 `productId` 參數來擷取所選產品的詳細資訊。

或者，您可以從另一個自動化叫用此自動化，循環瀏覽產品清單，並透過傳遞產品的 ID 做為 `productId` 參數來擷取每個產品的詳細資訊。在此案例中，`productId` 參數值會從迴圈的每個反覆運算中的 `{{product.id}}` 表達式動態提供。

透過使用 `productId` 參數和雙大括號語法，您可以讓此自動化更具彈性且可重複使用。您可以只提供適當的產品 ID 做為來自不同來源的參數值，例如 UI 元件或其他自動化，讓單一自動化能夠擷取任何產品的詳細資訊，而不是為每個產品建立個別的自動化。

案例：使用備用值處理選用參數

假設您有一個具有必要「擁有者」欄的「任務」實體，但您希望此欄位在自動化中是選用的，並在未選取擁有者時提供備用值。

1. 使用名為 `owner` 的參數建立自動化 `Owner`，該參數會映射到 `Task` 實體的 `Owner` 欄位。
2. 由於實體中需要 `Owner` 欄位，因此 `Owner` 參數會與所需的設定同步。
3. 若要在自動化中將 `Owner` 參數設為選用，請關閉此參數 `required` 的設定。
4. 在自動化邏輯中，您可以使用類似的表達式 `{{params.Owner || currentUser.userId}}`。此表達式會檢查是否提供 `Owner` 參數。如果未提供，則會回復到目前使用者的 ID 做為擁有者。
5. 如此一來，如果使用者未在表單或元件中選取擁有者，自動化會自動將目前的使用者指派為任務的擁有者。

透過切換 `Owner` 參數 `required` 的設定並使用備用表達式，您可以將其與實體欄位需求分離，使其在自動化中為選用，並在未提供參數時提供預設值。

定義自動化參數類型

透過使用參數類型來指定資料類型和設定需求，您可以控制自動化的輸入。這有助於確保您的自動化使用預期的輸入可靠地執行。

從實體同步類型

從實體欄位定義動態同步參數類型和需求，可簡化與實體資料互動的建置自動化，確保參數一律反映最新的實體欄位類型和需求。

下列程序詳細說明從實體同步參數類型的一般步驟：

1. 建立具有輸入欄位（例如布林值、數字等）的實體，並視需要標記欄位。

2. 建立新的自動化。
3. 將參數新增至自動化，並在選擇類型時，選擇您要同步的實體欄位。資料類型和必要設定會自動從映射的實體欄位同步
4. 如有需要，您可以透過切換每個參數的開啟/關閉來覆寫「必要」設定。這表示所需的狀態不會與實體欄位保持同步，否則會保持同步。

手動定義類型

您也可以手動定義參數類型，而無需從實體同步

透過定義自訂參數類型，您可以建立接受特定輸入類型的自動化，並視需要處理選用或必要的參數，而無需依賴實體欄位映射。

1. 建立具有輸入欄位（例如布林值、數字等）的實體，並視需要標記欄位。
2. 建立新的自動化。
3. 將參數新增至自動化，並在選擇類型時，選擇所需的類型。

設定要傳遞至自動化參數的動態值

定義自動化的參數之後，您可以在叫用自動化時將值傳遞給它們。您可以透過兩種方式傳遞參數值：

1. 元件觸發：如果您要從元件觸發叫用自動化，例如按一下按鈕，您可以使用 JavaScript 表達式從元件內容傳遞值。例如，如果您有名為 `emailInput` 的文字輸入欄位，您可以使用下列表達式將其值傳遞至電子郵件參數：`ui.emailInput.value`。
2. 其他自動化：如果您要從另一個自動化叫用自動化，您可以使用 JavaScript 表達式從自動化內容傳遞值。例如，您可以傳遞另一個參數的值或上一個動作步驟的結果。

類型安全

透過定義具有字串、數字或布林等特定資料類型的參數，您可以確保傳遞至自動化的值為預期類型。

Note

在 App Studio 中，`date(s)` 是 ISO 字串日期，也會驗證這些日期。

這種類型的安全有助於防止類型不相符，這可能會導致自動化邏輯中的錯誤或意外行為。例如，如果您將參數定義為 `Number`，您可以確信傳遞給該參數的任何值都會是數字，而且您不需要在自動化內執行額外的類型檢查或轉換。

驗證

您可以將驗證規則新增至參數，確保傳遞至自動化的值符合特定條件。

雖然 App Studio 不提供參數的內建驗證設定，但您可以將 JavaScript 動作新增至自動化，以便在違反特定限制條件時擲出錯誤，以實作自訂驗證。

對於實體欄位，支援驗證規則的子集，例如最小值/最大值。不過，執行 `Create/Update/Delete` 記錄動作時，這些項目不會在自動化層級進行驗證，只會在資料層進行。

自動化參數的最佳實務

為了確保您的自動化參數設計良好、可維護且易於使用，請遵循下列最佳實務：

1. 使用描述性參數名稱：選擇可清楚描述參數目的或內容的參數名稱。
2. 提供參數描述：定義參數以解釋其目的、限制條件和期望時，請利用描述欄位。在參考參數時，這些描述會出現在 JSDoc 註解中，以及在使用者調用自動化時需要為參數提供值的任何使用者介面中。
3. 使用適當的資料類型：根據預期的輸入值仔細考慮每個參數的資料類型，例如：字串、數字、布林值、物件。
4. 驗證參數值：在自動化中實作適當的驗證檢查，以確保參數值符合特定需求，然後再繼續進行進一步的動作。
5. 使用備用或預設值：雖然 App Studio 目前不支援設定參數的預設值，但您可以在使用自動化邏輯中的參數時實作備用或預設值。例如，如果未 `{{ params.param1 || "default value" }}` 提供 `param1` 參數或具有 `false` 值，您可以使用 `||` 之類的表達式來提供預設值。
6. 維持參數一致性：如果您有多個自動化需要類似的參數，請嘗試維持這些自動化中參數名稱和資料類型的一致性。
7. 文件參數用量：維護自動化的明確文件，包括每個參數的說明、其用途、預期值，以及任何相關的範例或邊緣案例。
8. 經常檢閱和重構：定期檢閱自動化及其參數，視需要重構或合併參數，以提高清晰度、可維護性和可重複使用性。
9. 限制參數數量：雖然參數提供彈性，但太多參數會使自動化複雜且難以使用。旨在透過將參數數量限制為僅必要項目，在靈活性和簡單性之間取得平衡。

- 10.考慮參數分組：如果您發現自己定義了多個相關參數，請考慮將它們分組為單一##參數。
- 11.個別考量：避免將單一參數用於多個用途，或將不相關的值得合併為單一參數。每個參數都應該代表不同的考量或資料片段。
- 12.使用參數別名：如果您有名稱較長或複雜的參數，請考慮在自動化邏輯中使用別名或速記版本，以提高可讀性和可維護性。

透過遵循這些最佳實務，您可以確保自動化參數設計良好、可維護且易於使用，最終改善自動化的整體品質和效率。

使用 JavaScript 在 App Studio 中撰寫表達式

在 AWS App Studio 中，您可以使用 JavaScript 表達式動態控制應用程式的行為和外觀。單行 JavaScript 表達式是以雙大括號和撰寫 `{{ }}`，可用於自動化、UI 元件和資料查詢等各種內容。這些表達式會在執行時間進行評估，可用於執行計算、操作資料和控制應用程式邏輯。

App Studio 提供三個 JavaScript 開放原始碼程式庫的原生支援：Luxon、UUID、Lodash 以及 SDK 整合，以偵測應用程式組態中的 JavaScript 語法和類型檢查錯誤。

Important

App Studio 不支援使用第三方或自訂 JavaScript 程式庫。

基本語法

JavaScript 表達式可以包含變數、常值、運算子和函數呼叫。運算式通常用於執行計算或評估條件。

請參閱以下範例：

- `{{ 2 + 3 }}` 將評估為 5。
- `{{ "Hello, " + "World!" }}` 將評估為「Hello, World!」。
- `{{ Math.max(5, 10) }}` 將評估為 10。
- `{{ Math.random() * 10 }}` 傳回介於 **【0-10) 之間的隨機數字 (含小數)**。

插補

您也可以使用 JavaScript 在靜態文字中插入動態值。這可透過將 JavaScript 表達式封裝在雙大括號內來實現，如下列範例所示：

```
Hello {{ currentUser.firstName }}, welcome to App Studio!
```

在此範例中，`currentUser.firstName` 是一種 JavaScript 表達式，會擷取目前使用者的名字，然後動態插入問候語訊息中。

串連

您可以使用 JavaScript 中的 + 運算子串連字串和變數，如下列範例所示。

```
{{ currentRow.FirstName + " " + currentRow.LastName }}
```

此表達式結合了 `currentRow.FirstName` 和 `currentRow.LastName` 的值，以及介於兩者之間的空格，進而產生目前資料列的完整名稱。例如，如果 `currentRow.FirstName` 是 John，而 `currentRow.LastName` 是 Doe，則表達式會解析為 John Doe。

日期和時間

JavaScript 提供各種函數和物件來使用日期和時間。例如：

- `{{ new Date().toLocaleDateString() }}`：以當地語系化格式傳回目前日期。
- `{{ DateTime.now().toISODate() }}`：以 YYYY-MM-DD 格式傳回目前日期，以供日期元件使用。

日期和時間比較

使用 `=`、`>=`、`>`、`<` 或 `<=` 等運算子來比較日期或時間值。例如：

- `{{ ui.timeInput.value > "10:00 AM" }}`：檢查時間是否在上午 10:00 之後。
- `{{ ui.timeInput.value <= "5:00 PM" }}`：檢查時間是否在下午 5:00 之前。
- `{{ ui.timeInput.value > DateTime.now().toISOTime() }}`：檢查時間是否晚於目前時間。

- `{{ui.dateInput.value > DateTime.now().toISODate()}}`：檢查日期是否早於目前日期。
- `{{ DateTime.fromISO(ui.dateInput.value).diff(DateTime.now(), "days").days >= 5 }}`：檢查日期是否距離目前日期至少 5 天。

程式碼區塊

除了表達式之外，您也可以撰寫多行 JavaScript 程式碼區塊。與表達式不同，程式碼區塊不需要大括號。反之，您可以直接在程式碼區塊編輯器中撰寫 JavaScript 程式碼。

Note

評估表達式並顯示其值時，會執行程式碼區塊，並顯示其輸出（如果有的話）。

全域變數和函數

App Studio 可讓您存取可在 JavaScript 表達式和程式碼區塊中使用的特定全域變數和函數。例如，`currentUser` 是代表目前登入使用者的全域變數，您可以存取之類的屬性 `currentUser.role` 來擷取使用者的角色。

參考或更新 UI 元件值

您可以在元件和自動化動作中使用表達式來參考和更新 UI 元件值。透過以程式設計方式參考和更新元件值，您可以建立動態和互動式使用者介面，以回應使用者輸入和資料變更。

參考 UI 元件值

您可以透過從 UI 元件存取值，實作動態行為來建立互動式和資料驅動型應用程式。

您可以在運算式中使用 `ui` 命名空間，存取相同頁面上 UI 元件的值和屬性。透過參考元件的名稱，您可以擷取其值或根據其狀態執行操作。

Note

`ui` 命名空間只會顯示目前頁面上的元件，因為元件的範圍是各自的頁面。

在 App Studio 應用程式中參考元件的基本語法為：`{{ui.componentName}}`。

下列清單包含使用 `ui` 命名空間存取 UI 元件值的範例：

- `{{ui.textInputName.value}}`：代表名為 *textInputName* 的文字輸入元件的值。
- `{{ui.formName.isValid}}`：根據您提供的驗證條件，檢查名為 *formName* 之表單中的所有欄位是否有效。
- `{{ui.tableName.currentRow.columnName}}`：代表名為 *tableName* 之資料表元件目前資料列中特定資料欄的值。
- `{{ui.tableName.selectedRowData.fieldName}}`：代表名為 *tableName* 之資料表元件中所選資料列中指定欄位的值。然後，您可以附加欄位名稱，例如 `ID({{ui.tableName.selectedRowData.ID}})`，以參考所選資料列中該欄位的值。

下列清單包含參考元件值的更具體範例：

- `{{ui.inputText1.value.trim().length > 0}}`：檢查 *inputText1* 元件的值，在修剪任何前後空格之後，是否具有非空白字串。這有助於根據輸入文字欄位的值驗證使用者輸入或啟用/停用其他元件。
- `{{ui.multiSelect1.value.join(", ")}}`：對於名為 *multiSelect1* 的多選元件，此表達式會將所選選項值的陣列轉換為逗號分隔字串。這有助於以易於使用的格式顯示選取的選項，或將選取項目傳遞給另一個元件或自動化。
- `{{ui.multiSelect1.value.includes("option1")}}`：此表達式會檢查值 `## 1` 是否包含在 *multiSelect1* 元件的所選選項陣列中。如果選取 `## 1`，則傳回 `true`，否則傳回 `false`。這對於條件式轉譯元件或根據特定選項選擇採取動作非常有用。
- `{{ui.s3Upload1.files.length > 0}}`：對於名為 *s3Upload1* 的 Amazon S3 檔案上傳元件，此表達式會檢查檔案陣列的長度，以檢查是否已上傳任何檔案。根據檔案是否已上傳，這對於啟用/停用其他元件或動作很有用。
- `{{ui.s3Upload1.files.filter(file => file.type === "image/png").length}}`：此表達式會篩選 *s3Upload1* 元件中上傳的檔案清單，只包含 PNG 影像檔案，並傳回這些檔案的計數。這有助於驗證或顯示有關上傳檔案類型的資訊。

更新 UI 元件值

若要更新或操作元件的值，請在 `自動化RunComponentAction` 中使用。以下是語法範例，您可以使用 `RunComponentAction` 動作來更新名為 *myInput* 之文字輸入元件的值：

```
RunComponentAction(ui.myInput, "setValue", "New Value")
```

在此範例中，RunComponentAction 步驟會呼叫 *myInput* 元件上的 setValue 動作，傳入新值 *New Value*。

使用資料表資料

您可以存取資料表資料和值來執行操作。您可以使用下列表達式來存取資料表資料：

- `currentRow`：用於從資料表中的目前資料列存取資料表資料。例如，設定資料表動作的名稱、將資料列的值傳送至從動作啟動的自動化，或使用資料表中現有資料欄的值來建立新的資料欄。
- `ui.tableName.selectedRow` 和 `ui.tableName.selectedRowData` 都用於從頁面上的其他元件存取資料表資料。例如，根據選取的資料列，在資料表外部設定按鈕的名稱。傳回的值相同，但 `selectedRow` 和 `selectedRowData` 之間的差異如下：
 - `selectedRow`：此命名空間包含每個欄位的資料欄標頭中顯示的名稱。參考資料表中可見資料欄的值 `selectedRow` 時，您應該使用。例如，如果您的資料表中有自訂或運算的資料欄，而該資料欄不存在於實體中的欄位。
 - `selectedRowData`：此命名空間包含做為資料表來源的實體中的欄位。您應該使用 `selectedRowData` 來參考資料表中不可見之實體的值，但對於應用程式中的其他元件或自動化很有用。

下列清單包含存取表達式中資料表資料的範例：

- `{{ui.tableName.selectedRow.columnNameWithNoSpace}}`：從資料表中選取的資料列傳回 *columnNameWithNoSpace* 資料欄的值。
- `{{ui.tableName.selectedRow['Column Name With Space']}}`：從資料表中選取的資料列傳回 ##### 資料欄的值。
- `{{ui.tableName.selectedRowData.fieldName}}`：從資料表中選取的資料列傳回 *fieldName* 實體欄位的值。
- `{{ui.tableName.selectedRows[0].columnMappingName}}`：從相同頁面上的其他元件或表達式參考所選資料列的資料欄名稱。
- `{{currentRow.firstName + ' ' + currentRow.lastNamecolumnMapping}}`：串連來自多個資料欄的值，以在資料表中建立新的資料欄。
- `{{ { "Blocked": "#", "Delayed": "#", "On track": "#" } [currentRow.statuscolumnMapping] + " " + currentRow.statuscolumnMapping}}`：根據儲存的狀態值自訂資料表內欄位的顯示值。

- `{{currentRow.colName}}`、`{{currentRow["First Name"]}}`、`{{currentRow}}`或 `{{ui.tableName.selectedRows[0]}}`：在資料列動作內傳遞參考的資料列內容。

存取自動化

您可以使用自動化在 App Studio 中執行伺服器端邏輯和操作。在自動化動作中，您可以使用表達式來處理資料、產生動態值，以及納入先前動作的結果。

存取自動化參數

您可以將動態值從 UI 元件和其他自動化傳遞至自動化，使其可重複使用且靈活。這是使用自動化參數搭配 `params` 命名空間來完成的，如下所示：

`{{params.parameterName}}`：參考從 UI 元件或其他來源傳遞至自動化的值。例如，`{{params.ID}}` 會參考名為 `ID` 的參數。

操縱自動化參數

您可以使用 JavaScript 來操作自動化參數。請參閱以下範例：

- `{{params.firstName}} {{params.lastName}}`：做為參數傳遞的串連值。
- `{{params.numberParam1 + params.numberParam2}}`：新增兩個數字參數。
- `{{params.valueProvided?.length > 0 ? params.valueProvided : 'Default'}}`：檢查參數不是 null 或未定義，而且長度不為零。如果為 true，請使用提供的值；否則，請設定預設值。
- `{{params.rootCause || "No root cause provided"}}`：如果 `params.rootCause` 參數為 false (null、未定義或空字串)，請使用提供的預設值。
- `{{Math.min(params.numberOfProducts, 100)}}`：將參數的值限制為最大值（在此情況下為 100）。
- `{{ DateTime.fromISO(params.startDate).plus({ days: 7 }).toISO() }}`：如果 `params.startDate` 參數為 "2023-06-15T10:30:00.000Z"，則此表達式將評估為 "2023-06-22T10:30:00.000Z"，即開始日期後一週的日期。

存取先前動作的自動化結果

自動化可讓應用程式執行伺服器端邏輯和操作，例如查詢資料庫、與 APIs 互動或執行資料轉換。`results` 命名空間可讓您存取相同自動化內先前動作傳回的輸出和資料。請注意下列有關存取自動化結果的要點：

1. 您只能存取相同自動化內先前自動化步驟的結果。
2. 如果您有動作名為 *action1* 和 *action2*，則 *action1* 無法參考任何結果，而 *action2* 只能存取 `results.action1`。
3. 這也適用於用戶端動作。例如，如果您有一個使用 `InvokeAutomation` 動作觸發自動化的按鈕。然後，如果自動化指出檔案是 PDF，則您可以使用具有類似 `Run If` 條件的導覽步驟 `results.myInvokeAutomation1.fileType === "pdf"`，以使用 PDF 檢視器導覽至頁面。

下列清單包含使用 `results` 命名空間存取先前動作自動化結果的語法。

- `{{results.stepName.data}}`：從名為 *stepName* 的自動化步驟擷取資料陣列。
- `{{results.stepName.output}}`：擷取名為 *stepName* 之自動化步驟的輸出。

您存取自動化步驟結果的方式取決於動作類型及其傳回的資料類型。不同的動作可能會傳回不同的屬性或資料結構。以下是一些常見的範例：

- 對於資料動作，您可以使用 存取傳回的資料陣列 `results.stepName.data`。
- 對於 API 呼叫動作，您可以使用 存取回應內文 `results.stepName.body`。
- 對於 Amazon S3 動作，您可以使用 存取檔案內容 `results.stepName.Body.transformToWebStream()`。

請參閱您使用的特定動作類型文件，以了解其傳回的資料形狀，以及如何在 `results` 命名空間中存取。下列清單包含一些範例

- `{{results.getDataStep.data.filter(row => row.status === "pending").length}}`：假設 *getDataStep* 是傳回資料列陣列的 `Invoke Data Action` 自動化動作，此表達式會篩選資料陣列，只包含狀態欄位等於 `pending` 的資料列，並傳回篩選陣列的長度（計數）。這對於根據特定條件查詢或處理資料非常有用。
- `{{params.email.split("@")[0]}}`：如果 `params.email` 參數包含電子郵件地址，此表達式會以 `@` 符號分割字串，並在 `@` 符號之前傳回 部分，有效地擷取電子郵件地址的使用者名稱部分。
- `{{new Date(params.timestamp * 1000)}}`：此表達式採用 Unix 時間戳記參數 (`params.timestamp`)，並將其轉換為 JavaScript `Date` 物件。其假設時間戳記以秒為單位，因此會將時間戳記乘以 1000，將其轉換為毫秒，這是 `Date` 建構函數預期的格式。這對於在自動化中使用日期和時間值很有用。

- `{{results.stepName.Body}}`：對於名為 *stepName* 的 Amazon S3 GetObject 自動化動作，此表達式會擷取檔案內容，例如影像或 PDF 檢視器等 UI 元件可用來顯示擷取的檔案。請注意，此表達式需要在自動化的自動化輸出中設定，才能在元件中使用。

資料相依性和時間考量

在 App Studio 中建置複雜的應用程式時，請務必了解和管理不同資料元件之間的資料相依性，例如表單、詳細資訊檢視和自動化支援的元件。資料元件和自動化可能不會同時完成其資料擷取或執行，這可能會導致時間問題、錯誤和意外行為。透過了解潛在的計時問題並遵循最佳實務，您可以在 App Studio 應用程式中建立更可靠且一致的使用者體驗。

有些潛在問題如下所示：

1. 轉譯時間衝突：資料元件的轉譯順序可能不符合其資料相依性，進而可能導致視覺化不一致或錯誤。
2. 自動化執行時間：自動化任務可能會在元件完全載入之前完成，導致執行時間執行錯誤。
3. 元件當機：由自動化提供支援的元件可能會在無效的回應或自動化尚未完成執行時當機。

範例：訂單詳細資訊和客戶資訊

此範例示範資料元件之間的相依性如何導致資料顯示中的計時問題和潛在錯誤。

考慮在同一頁面上具有下列兩個資料元件的應用程式：

- 擷取訂單資料的詳細資訊元件 (`orderDetails`)。
- 顯示訂單相關客戶詳細資訊的詳細資訊元件 (`customerDetails`)。

在此應用程式中，`orderDetails` 詳細資訊元件中有兩個欄位，以下列值設定：

```
// 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
```

在此範例中，`orderDetails`元件正在嘗試透過參考元件中的資料來顯示客戶名稱`customerDetails`。這是有問題的，因為`orderDetails`元件可能會在`customerDetails`元件擷取其資料之前轉譯。如果`customerDetails`元件資料擷取延遲或失敗，`orderDetails`元件會顯示不完整或不正確的資訊。

資料相依性和時間最佳實務

使用下列最佳實務來緩解 App Studio 應用程式中的資料相依性和計時問題：

1. 使用條件式轉譯：只有在確認可用時，才會轉譯元件或顯示資料。使用條件式陳述式在顯示資料之前檢查資料是否存在。下列程式碼片段顯示範例條件式陳述式：

```
{{ui.someComponent.data ? ui.someComponent.data.fieldName : "Loading..."}}
```

2. 管理子元件可見性：對於在載入資料之前轉譯子元件的 Stepflow、Form 或 Detail 等元件，請手動設定子元件的可見性。下列程式碼片段顯示根據父元件資料可用性設定可見性的範例：

```
{{ui.parentComponent.data ? true : false}}
```

3. 使用聯結查詢：盡可能使用聯結查詢在單一查詢中擷取相關資料。這可減少個別資料擷取的數量，並將資料元件之間的時間問題降至最低。
4. 在自動化中實作錯誤處理：在自動化中實作強大的錯誤處理，以正常管理預期資料不可用或收到無效回應的情況。
5. 使用選用鏈結：存取巢狀屬性時，如果父屬性未定義，請使用選用鏈結來防止錯誤。下列程式碼片段顯示選用鏈結的範例：

```
{{ui.component.data?.[0]?.fieldSystemName}}
```

使用多個使用者建置應用程式

多個使用者可以使用單一 App Studio 應用程式，但一次只能有一個使用者編輯應用程式。如需邀請其他使用者編輯應用程式的資訊，以及多個使用者同時嘗試編輯應用程式時的行為，請參閱下列各節。

邀請建置者編輯應用程式

使用下列指示來邀請其他建置器編輯 App Studio 應用程式。

邀請其他建置者編輯應用程式

1. 如有必要，請導覽至您應用程式的應用程式工作室。
2. 選擇共用。
3. 在開發索引標籤中，使用文字方塊來搜尋和選取您要邀請來編輯應用程式的群組或個別使用者。
4. 針對每個使用者或群組，選擇下拉式清單，然後選取要授予該使用者或群組的許可。
 - 共同擁有者：共同擁有者具有與應用程式擁有者相同的許可。
 - 僅限編輯：具有僅限編輯角色的使用者具有與擁有者和共同擁有者相同的許可，但下列項目除外：
 - 他們無法邀請其他使用者編輯應用程式。
 - 他們無法將應用程式發佈至測試或生產環境。
 - 他們無法將資料來源新增至應用程式。
 - 他們無法刪除或複製應用程式。

嘗試編輯另一個使用者正在編輯的應用程式

單一 App Studio 應用程式一次只能由一個使用者編輯。請參閱下列範例，以了解多個使用者同時嘗試編輯應用程式時會發生什麼情況。

在此範例中，User A 目前正在編輯應用程式，並已與共用 User B。User B 然後會嘗試編輯正在由編輯的應用程式 User A。

User B 當嘗試編輯應用程式時，將出現一個對話方塊，通知他們 User A 目前正在編輯應用程式，而繼續將 User A 退出應用程式工作室，並將儲存所有變更。User B 可以選擇取消並 User A 繼續，或繼續並輸入應用程式工作室來編輯應用程式。在此範例中，他們選擇編輯應用程式。

當 User B 選擇編輯應用程式時，User A 會收到 User B 已開始編輯應用程式的通知，且其工作階段已結束。請注意，如果 User A 在非作用中瀏覽器索引標籤中開啟應用程式，他們可能不會收到通知。在這種情況下，如果他們嘗試返回應用程式並嘗試進行編輯，他們將收到錯誤訊息並被引導重新整理頁面，該頁面將返回應用程式清單。

檢視或更新應用程式的內容安全設定

App Studio 中的每個應用程式都有內容安全設定，可用於限制外部媒體或資源，例如映像、iFrames 和 PDFs 載入，或僅允許來自指定的網域或 URLs（包括 Amazon S3 儲存貯體）。您也可以指定應用程式可將物件上傳至 Amazon S3 的網域。

所有應用程式的預設內容安全設定是封鎖從外部來源載入所有媒體，包括 Amazon S3 儲存貯體，以及封鎖將物件上傳至 Amazon S3。因此，若要載入映像、iFrames、PDFs 或類似媒體，您必須編輯設定以允許媒體來源。此外，若要允許將物件上傳到 Amazon S3，您必須編輯設定以允許可上傳到的網域。

Note

內容安全設定用於設定應用程式中的內容安全政策 (CSP) 標頭。CSP 是一項安全標準，可協助保護您的應用程式免受跨網站指令碼 (XSS)、點擊劫持和其他程式碼注入攻擊。如需 CSP 的詳細資訊，請參閱 MDN Web 文件中的 [內容安全政策 \(CSP\)](#)。

更新應用程式的內容安全設定

1. 如有必要，請從應用程式清單中選擇編輯，以導覽至應用程式的應用程式工作室。
2. 選擇應用程式設定。
3. 選擇內容安全設定索引標籤以檢視下列設定：
 - 影格來源：用於管理應用程式可以從中載入影格和 iframe（例如互動式內容或 PDFs 網域。此設定會影響下列元件或應用程式資源：
 - iFrame 內嵌元件
 - PDF 檢視器元件
 - 映像來源：用於管理應用程式可以從中載入映像的網域。此設定會影響下列元件或應用程式資源：
 - 應用程式標誌和橫幅
 - 影像檢視器元件
 - 連線來源：用於管理應用程式可上傳 Amazon S3 物件的網域。
4. 對於每個設定，從下拉式清單中選擇所需的設定：
 - 封鎖所有 frames/images/connections：不允許載入任何媒體（影像、影格、PDFs）或上傳到 Amazon S3 的任何物件。
 - 允許所有 frames/images/connections：允許所有網域的所有媒體（影像、影格、PDFs）載入，或允許將物件上傳至所有網域的 Amazon S3。
 - 允許特定網域：允許從指定的網域載入媒體或將媒體上傳至指定的網域。網域或 URLs 指定為以空格分隔的表達式清單，其中萬用字元 (*) 可用於子網域、主機地址或連接埠號碼，以指出每個的所有法律值都有效。指定 http 也符合 https。下列清單包含有效項目的範例：

- `blob:` : 比對所有 Blob，其中包含自動化動作傳回的檔案資料，例如從 Amazon S3 儲存貯體 `GetObject` 傳回項目，或 Amazon Bedrock 產生的映像。

 Important

您必須將 包含 `blob:` 到您提供的表達式中，以允許動作傳回的檔案資料，即使您的表達式是 `*`，也應該將其更新為 `* blob:`

- `http://*.example.com` : 比對從 的任何子網域載入的所有嘗試 `example.com`。也符合 `https` 資源。
 - `https://source1.example.com https://source2.example.com` : 符合從 `https://source1.example.com` 和 載入的所有嘗試 `https://source2.example.com`
 - `https://example.com/subdirectory/` : 比對在子目錄目錄下載入檔案的所有嘗試。例如 `https://example.com/subdirectory/path/to/file.jpeg`。它不符合 `https://example.com/path/to/file.jpeg`。
5. 選擇儲存，以儲存變更。

疑難排解和偵錯 App Studio

主題

- [疑難排解 App Studio 設定、許可和加入](#)
- [疑難排解和偵錯應用程式](#)
- [對發佈和共用應用程式進行故障診斷](#)

疑難排解 App Studio 設定、許可和加入

本主題包含設定或加入 App Studio 時常見問題的疑難排解，以及管理許可的相關資訊。

選擇為我建立帳戶執行個體選項時，App Studio 設定失敗

問題：如果您在任何 AWS 區域中有帳戶層級的 IAM Identity Center 執行個體，使用建立帳戶執行個體來設定 App Studio 將會失敗，因為 IAM Identity Center 僅支援一個執行個體。

解決方案：導覽至位於 <https://console.aws.amazon.com/singlesignon/> 的 IAM Identity Center 主控台，以檢查您是否有 IAM Identity Center 執行個體。檢查每個支援的 AWS 區域，直到您找到執行個體為止。您可以在設定 App Studio 時使用該執行個體，或刪除 IAM Identity Center 執行個體，並使用為我建立帳戶執行個體選項再試一次。

Warning

刪除 IAM Identity Center 執行個體會影響任何現有的使用案例。刪除之前，請確定未使用執行個體，或使用執行個體來設定 App Studio。

設定後無法存取 App Studio

問題：設定 App Studio 時，您可能已提供不屬於的 IAM Identity Center 群組。您必須是至少一個群組的成員，才能存取 App Studio。

解決方案：導覽至 IAM Identity Center 主控台，網址為 <https://console.aws.amazon.com/singlesignon/> : //。

不確定登入 App Studio 時要使用的使用者名稱或密碼

問題：您可能不確定如何登入 App Studio，因為您尚未設定 IAM Identity Center 登入資料，或忘記 IAM Identity Center 使用者名稱或密碼。

解決方案：在沒有 IAM Identity Center 執行個體的情況下設定 App Studio 時，會為每個使用者提供用於建立 IAM Identity Center 使用者的電子郵件和使用者名稱。提供的每個電子郵件地址都會收到一封電子郵件，其中包含加入 IAM Identity Center 的邀請。每個使用者都必須接受邀請，並為其 IAM Identity Center 使用者登入資料建立密碼。然後，每個使用者可以使用 IAM Identity Center 使用者名稱和密碼登入 App Studio。

如果您已經設定登入資料並忘記使用者名稱或密碼，您必須要求管理員使用 IAM Identity Center 主控台來檢視並提供使用者名稱，或重設密碼。

我在設定 App Studio 時收到系統錯誤

問題：您在設定 App Studio 時收到下列錯誤：

```
System error. We encountered a problem. Report the issue and the App Studio service team will get back to you.
```

當服務遇到未知錯誤時，會發生此錯誤。

解決方案：在左側導覽的學習區段或編輯應用程式時，選擇在 Slack 上加入我們，以加入社群 Slack 來聯絡支援團隊。

我找不到我的 App Studio 執行個體 URL

如果您找不到存取 App Studio 執行個體的 URL，請聯絡設定 App Studio 的管理員。管理員可以在的 App Studio 主控台中檢視 URL AWS 管理主控台。

我無法在 App Studio 中修改群組或角色

問題：您無法在左側導覽中看到角色連結。這是因為只有具有管理員角色的使用者才能修改 App Studio 中的群組和角色。

解決方案：聯絡具有管理員角色的使用者以變更群組或角色，或聯絡管理員以新增至管理員群組。

如何從 App Studio 離職

您目前無法從 App Studio 離職。建議移除所有資源，例如應用程式和連接器，並將群組的角色變更為 App User，以防止存取或使用。您也應該刪除專門用於 App Studio 的第三方資源，例如 IAM 角色或資料庫資料表。

疑難排解和偵錯應用程式

下列主題包含疑難排解和偵錯 App Studio 應用程式的資訊。

主題

- [疑難排解 AI 建置器助理和聊天](#)
- [在應用程式工作室進行故障診斷](#)
- [預覽應用程式的疑難排解](#)
- [在測試環境中進行故障診斷](#)
- [使用 Amazon CloudWatch Logs 中已發佈應用程式的日誌進行偵錯](#)
- [連接器故障診斷](#)

疑難排解 AI 建置器助理和聊天

本主題包含使用 AI 建置器助理時常見問題的疑難排解指引。

使用 AI 建立應用程式時發生錯誤

使用 AI 提示建立應用程式時，可能會發生下列錯誤：

```
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.
```

問題：請求因可能有害的內容而遭到封鎖。

解決方案：重述提示，然後再試一次。

使用 AI 產生的應用程式是空白應用程式或缺少的元件。

問題：這可能是因為非預期的服務錯誤所造成。

解決方案：重試使用 AI 建立應用程式，或在產生的應用程式中手動建立元件。

在應用程式工作室進行故障診斷

本主題包含建置應用程式時問題的疑難排解和偵錯指引。

使用偵錯面板

為了在建置應用程式時協助即時偵錯，App Studio 提供可摺疊的建置器偵錯面板，涵蓋應用程式工作室的頁面、自動化和資料索引標籤。此面板會顯示錯誤和警告。雖然警告可做為可行的建議，例如尚未設定的資源，但必須解決錯誤，才能成功預覽或發佈您的應用程式。每個錯誤或警告都包含檢視連結，可用於導覽至問題的位置。

偵錯面板會在新錯誤或警告發生時自動更新，錯誤或警告會在解決後自動消失。當您離開建置器時，這些警告和錯誤訊息的狀態會持續存在。

JavaScript 表達式語法和資料類型處理

App Studio 具有 JavaScript 錯誤偵測功能，可將程式碼以紅線加底線來強調錯誤。這些編譯錯誤會阻止應用程式成功建置，指出必要資料類型的錯別字、無效參考、無效操作和不正確輸出等問題。如需常見問題，請參閱下列清單：

1. 重新命名資源造成的錯誤：當 JavaScript 運算式在 App Studio 中參考資源名稱時，變更這些名稱會導致運算式不正確並產生錯誤。您可以在偵錯面板中檢視這些錯誤。
2. 資料類型問題：資料類型不相符會在您的應用程式中產生錯誤。例如，如果自動化設定為接受類型的參數String，但元件設定為傳送類型的值Integer，則會發生錯誤。檢查資料類型是否符合適當的資源，包括元件、自動化以及資料實體和動作。您可能需要變更 JavaScript 表達式中值的類型。

預覽應用程式的疑難排解

本主題包含嘗試預覽應用程式時疑難排解問題的相關資訊。

預覽載入失敗，並出現下列錯誤：**Your app failed to build and cannot be previewed**

問題：您的應用程式必須成功建置才能預覽。當編譯錯誤導致您的應用程式無法成功建置時，就會發生此錯誤。

解決方案：使用應用程式工作室中的偵錯面板來檢閱和解決錯誤。

預覽載入需要很長的時間

問題：某些類型的應用程式更新需要很長時間才能編譯和建置。

解決方案：保持標籤開啟，並等待建立更新。您應該會在應用程式的應用程式工作室右上角看到已儲存，預覽將會重新載入。

預覽不會反映最新的變更

問題：當另一個使用者接管您的應用程式編輯工作階段，但您未收到通知時，可能會發生這種情況。這可能會導致編輯的應用程式與預覽環境不相符。

解決方案：重新整理應用程式 Studio 瀏覽器索引標籤，並視需要接管編輯工作階段。

在測試環境中進行故障診斷

本主題包含有關對發佈至測試環境的應用程式進行故障診斷的資訊。

Note

自動化或資料動作的 HTTP 500 回應可能是由表達式中的執行時間當機、連接器故障，或從連接到應用程式的資料來源調節所造成。使用 [中的指示](#) [使用瀏覽器主控台除錯](#) 來檢視會顯示基礎錯誤詳細資訊的偵錯日誌。

使用偵錯面板

與建置應用程式時使用的建置偵錯面板類似，App Studio 會在測試環境中提供可摺疊的偵錯面板。此面板會顯示資訊性訊息，例如頁面載入時間、使用者導覽和應用程式事件。它還包含錯誤和警告。偵錯面板會在事件發生時自動更新新訊息。

使用瀏覽器主控台除錯

由於預覽您的應用程式時不會叫用動作，因此您的應用程式需要發佈至測試環境，以測試其呼叫和回應處理。如果在自動化執行期間發生錯誤，或者如果您想要了解應用程式為何以特定方式運作，您可以使用瀏覽器的主控台進行即時偵錯。

使用瀏覽器主控台在測試環境中偵錯應用程式

1. 附加?debug=true至 URL 的結尾，然後按 Enter 鍵。請注意，如果 URL 已有查詢字串（其中包含?），請改為附加&debug=true到 URL 的結尾。

2. 探索動作或 API 輸入和輸出，開啟瀏覽器主控台以開始偵錯。

- 在 Chrome 中：在瀏覽器中按一下滑鼠右鍵，然後選擇檢查。如需使用 Chrome DevTools 進行偵錯的詳細資訊，請參閱 [Chrome DevTools 文件](#)。
- 在 Firefox 中：按住或在網頁元素上按一下滑鼠右鍵，然後選擇檢查元素。如需使用 Firefox DevTools 進行偵錯的詳細資訊，請參閱 [Firefox DevTools 使用者文件](#)。

下列清單包含一些產生錯誤的常見問題：

- 執行時間錯誤
 - 問題：如果自動化或表達式設定不正確，執行自動化時可能會導致錯誤。常見的錯誤是重新命名資產，導致不正確的表達式、其他 JavaScript 編譯錯誤，或嘗試使用的資料或資產 undefined。
 - 解決方案：檢查每個自訂程式碼輸入的使用情況（運算式、JavaScript 和 JSON），並確保程式碼編輯器或偵錯面板中沒有編譯錯誤。
- 連接器問題
 - 問題：由於 App Studio 應用程式在發佈連接器之前不會與外部服務通訊，因此在測試環境中可能會發生在預覽期間未發生的錯誤。如果自動化中使用連接器的動作失敗，可能是來自將請求傳送至連接器的動作中的組態錯誤，或使用連接器組態本身。
 - 解決方案：您應該在預覽環境中提早使用模擬輸出來測試自動化，以防止這些錯誤。請確定您的連接器已正確設定，如需詳細資訊，請參閱 [連接器故障診斷](#)。最後，您可以使用 CloudWatch 來檢閱日誌。如需詳細資訊，請參閱 [使用 Amazon CloudWatch Logs 中已發佈應用程式的日誌進行偵錯](#)。在 ConnectorService 命名空間日誌中，應該會有源自連接器的錯誤訊息或中繼資料。

使用 Amazon CloudWatch Logs 中已發佈應用程式的日誌進行偵錯

Amazon CloudWatch Logs 會即時監控您的 AWS 資源和您在 上執行 AWS 的應用程式。您可以使用 CloudWatch Logs 來收集和追蹤指標，這些指標是您可以針對資源和應用程式測量的變數。

對於對 App Studio 應用程式進行偵錯，CloudWatch Logs 有助於追蹤應用程式執行期間發生的錯誤、稽核資訊，並提供使用者動作和專屬互動的內容。日誌提供歷史資料，您可以用來稽核應用程式用量和存取模式，以及檢閱使用者遇到的錯誤。

Note

CloudWatch Logs 不提供從應用程式 UI 傳遞的參數值的即時追蹤。

使用下列程序從 CloudWatch Logs 中的 App Studio 應用程式存取日誌。

1. 在應用程式的 App Studio 應用程式工作室中，查看 URL，尋找並記下您的應用程式 ID。應用程式 ID 看起來可能會類似這樣：802a3bd6-ed4d-424c-9f6b-405aa42a62c5。
2. 透過 <https://console.aws.amazon.com/cloudwatch/> 開啟 CloudWatch 主控台。
3. 在導覽窗格中，選擇 Log groups (日誌群組)。
4. 在這裡，您可以找到每個應用程式的五個日誌群組。根據您感興趣的資訊類型，選取群組並針對您要探索的資料撰寫查詢。

下列清單包含日誌群組，以及何時使用每個日誌群組的相關資訊：

1. `/aws/appstudio/teamId/appId/TEST/app`：使用 偵錯目前發佈至測試環境的應用程式版本相關的自動化回應、元件錯誤或 JavaScript 程式碼。
2. `/aws/appstudio/teamId/appId/TEST/audit`：使用 偵錯 JavaScript 程式碼錯誤，例如與目前發佈至測試環境的應用程式版本相關的條件式可見性或轉換、查詢失敗，以及登入或許可使用者錯誤。
3. `/aws/appstudio/teamId/setup`：使用 監控建置器或管理員動作。
4. `/aws/appstudio/teamId/appId/PRODUCTION/app`：使用 偵錯目前發佈至生產環境的應用程式版本相關的自動化回應、查詢失敗、元件錯誤或 JavaScript 程式碼。
5. `/aws/appstudio/teamId/appId/PRODUCTION/audit`：使用 偵錯 JavaScript 程式碼錯誤，例如條件式可見性或轉換，以及與目前發佈至生產環境的應用程式版本相關的登入或許可使用者錯誤。

Note

用於偵錯的大多數日誌都會在 `DebugLogClient` 命名空間下分類。

5. 進入日誌群組後，您可以選擇最近的日誌串流，或最後一個事件時間最接近感興趣的時間的日誌串流，也可以選擇搜尋所有日誌串流，以搜尋該日誌群組上的所有事件。如需在 CloudWatch Logs 中檢視日誌資料的詳細資訊，請參閱[檢視傳送至 CloudWatch Logs 的日誌資料](#)。

使用 CloudWatch Logs Insights 查詢來篩選和排序日誌

您可以使用 CloudWatch Logs Insights 一次查詢多個日誌群組。識別包含工作階段資訊的日誌群組清單後，請導覽至 CloudWatch Logs Insights 並選取日誌群組。然後，透過自訂查詢進一步縮小目標日誌項目。以下是一些範例查詢：

包含關鍵字的日誌清單：**##**

```
fields @timestamp, @message
| filter @message like 'error'
| sort @timestamp desc
```

從測試環境偵錯日誌：

```
fields @timestamp, @message
| filter namespace = "DebugLogClient"
| sort @timestamp desc
```

5 分鐘內的整體 504/404/500 錯誤計數：

```
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)
```

如需 CloudWatch Logs Insights 的詳細資訊，請參閱《Amazon CloudWatch [Logs 使用者指南](#)》中的 [使用 CloudWatch Logs Insights 分析日誌資料](#)。

連接器故障診斷

本主題包含常見連接器問題的疑難排解指引。您必須是管理員群組的成員，才能檢視或編輯連接器。

檢查您的 IAM 角色是否具有正確的自訂信任政策和標籤

為您的連接器設定 IAM 角色時，請確定自訂信任政策已正確設定為提供對 App Studio 的存取。如果 AWS 資源位於用來設定 App Studio 的相同 AWS 帳戶中，則仍然需要此自訂信任政策。

- 確保 Principal 區段中的 AWS 帳號是用於設定 App Studio 之帳戶的帳戶 AWS ID。此帳戶號碼不一定是資源所在的帳戶。
- 確保 "aws:PrincipalTag/IsAppStudioAccessRole": "true" 在 sts:AssumeRole 區段中正確新增。

JSON

```
{
```

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

此外，請確定具有下列索引鍵和值的標籤已新增至 IAM 角色，如需新增標籤的詳細資訊，請參閱[標籤 IAM 角色](#)：

Note

請注意，標籤的值為 `IsAppStudioDataAccessRole`，這與自訂信任政策 () 中的值略有不同 `IsAppStudioAccessRole`。

- 索引鍵：IsAppStudioDataAccessRole
- Value (值)：true

檢查連接器連線之產品或服務中資源的組態。有些資源，例如 Amazon Redshift 資料表，需要額外的組態才能與 App Studio 搭配使用。

檢查您的連接器組態。對於 AWS 服務，請前往 App Studio 中的連接器，並確保包含正確的 Amazon Resource Name (ARN)，且指定的 AWS 區域是包含您資源的區域。

檢查您的 IAM 角色是否具有正確的許可

若要提供 App Studio AWS 資源的存取權，您必須將適當的許可指派給連接器所使用的 IAM 角色。所需的許可對於要執行的服務、資源和動作是唯一的。例如，從 Amazon Redshift 資料表讀取資料需要

與將物件上傳至 Amazon S3 儲存貯體不同的許可。如需[連線至 AWS 服務](#)詳細資訊，請參閱 中的適當主題。

對 Amazon Redshift 連接器進行故障診斷

本節包含 Amazon Redshift 連接器常見問題的疑難排解指引。如需設定 Amazon Redshift 連接器和資源的資訊，請參閱 [連線至 Amazon Redshift](#)。

1. 確定 Isolated Session 切換在 Amazon Redshift 編輯器 OFF 上設定為 。需要此設定才能查看其他使用者所做的資料變更，例如 App Studio 應用程式。
2. 確保在 Amazon Redshift 資料表上授予適當的許可。
3. 在連接器組態中，確定已選取適當的運算類型 (Provisioned 或 Serverless)，以符合 Amazon Redshift 資料表類型。

故障診斷 Aurora 連接器

本節包含 Aurora 連接器常見問題的疑難排解指引。如需設定 Aurora 連接器和資源的資訊，請參閱 [連線至 Amazon Aurora](#)。

1. 建立資料表時，請確定已選擇適當且支援的 Aurora 版本。
2. 確認 Amazon RDS Data API 已啟用，因為這是允許 App Studio 在 Aurora 資料表上執行操作的必要條件。如需詳細資訊，請參閱 [啟用 Amazon RDS Data API](#)。
3. 確認已提供 AWS Secrets Manager 許可。

故障診斷 DynamoDB 連接器

本節包含 DynamoDB 連接器常見問題的疑難排解指引。如需設定 DynamoDB 連接器和資源的資訊，請參閱 [連線至 Amazon DynamoDB](#)。

如果您的 DynamoDB 資料表結構描述在建立連接器時未出現，可能是因為您的 DynamoDB 資料表使用客戶受管金鑰 (CMK) 加密，而且資料表資料無法在沒有描述金鑰和解密資料表的許可的情況下存取。若要使用以 CMK 加密的資料表建立 DynamoDB 連接器，您必須將 `kms:decrypt` 和 `kms:describeKey` 許可新增至 IAM 角色。

對 Amazon S3 連接器進行故障診斷

本節包含 Amazon S3 連接器常見問題的疑難排解指引。如需設定 Amazon S3 連接器和資源的資訊，請參閱 [連線至 Amazon Simple Storage Service \(Amazon S3\)](#)。

一般故障診斷指引包括檢查下列項目：

1. 確保使用 Amazon S3 資源所在的 AWS 區域設定 Amazon S3 連接器。
2. 確定已正確設定 IAM 角色。
3. 在 Amazon S3 儲存貯體中，確保 CORS 組態授予適當的許可。如需詳細資訊，請參閱[步驟 1：建立和設定 Amazon S3 資源](#)。

Amazon S3 檔案上傳錯誤：無法計算預先簽章的 URL

嘗試使用 Amazon S3 儲存貯體時，您可能會遇到下列錯誤：

```
Error while uploading file to S3: Failed to calculate presigned URL.
```

此錯誤通常是由不正確的 IAM 角色組態或 Amazon S3 儲存貯體上的不正確 CORS 組態所造成，可以透過使用 [中的資訊修正這些組態來解決連線至 Amazon Simple Storage Service \(Amazon S3\)](#)。

對發佈和共用應用程式進行故障診斷

本主題包含發佈或共用 App Studio 應用程式時常見問題的疑難排解指引。

我在共用對話方塊中看不到新建立的應用程式角色

重新發佈應用程式後，新建立的應用程式層級角色才會顯示在共用對話方塊中。在建立新角色之後發佈應用程式以使用它們。

我的應用程式發佈完成時，我未收到電子郵件

只有應用程式擁有者會在應用程式發佈時收到電子郵件。

我應用程式的最終使用者無法存取已發佈的應用程式

如果您的最終使用者無法存取您發佈的應用程式，並在嘗試存取它時收到Forbidden訊息，則發佈的應用程式可能不會與嘗試存取該應用程式的使用者共用。發佈的應用程式必須與群組共用，才能將存取權授予群組中的使用者。

若要深入了解共用應用程式，請參閱[共用已發佈的應用程式](#)。

AWS App Studio 的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全性是 AWS 與您之間共同責任。[共同責任模型](#) 將此描述為雲端的安全和雲端內的安全：

- 雲端的安全性 – AWS 負責保護在 AWS Cloud 中執行 AWS 服務的基礎設施。AWS 也為您提供可安全使用的服務。第三方稽核人員定期檢測及驗證安全的效率也是我們 [AWS 合規計劃](#) 的一部分。若要了解適用於 App Studio 的合規計劃，請參閱 [AWS 合規計劃範圍內的服務](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對資料敏感度、組織要求，以及適用法律和法規等其他因素負責。

本文件將協助您了解如何在使用 App Studio 時套用共同責任模型。下列主題說明如何設定 App Studio 以符合您的安全與合規目標。您也將了解如何使用其他 AWS 服務，協助您監控和保護 App Studio 資源。

主題

- [安全考量和緩解措施](#)
- [AWS App Studio 中的資料保護](#)
- [AWS App Studio 和 AWS Identity and Access Management \(IAM\)](#)
- [AWS App Studio 的合規驗證](#)
- [AWS App Studio 中的彈性](#)
- [AWS App Studio 中的基礎設施安全](#)
- [AWS App Studio 中的組態和漏洞分析](#)
- [預防跨服務混淆代理人](#)
- [AWS App Studio 中的跨區域資料傳輸](#)

安全考量和緩解措施

安全考量

處理資料連接器、資料模型和已發佈的應用程式時，會出現與資料暴露、存取控制和潛在漏洞相關的幾個安全問題。下列清單包含主要安全問題。

IAM 角色組態不正確

資料連接器的 IAM 角色組態不正確可能會導致未經授權的存取和資料外洩。授予對資料連接器 IAM 角色過度寬鬆的存取權，可以允許未經授權的使用者存取和修改敏感資料。

使用 IAM 角色來執行資料操作

由於 App Studio 應用程式的最終使用者會擔任連接器組態中提供的 IAM 角色來執行動作，因此這些最終使用者可能會存取他們通常無法存取的資料。

刪除已發佈應用程式的資料連接器

刪除資料連接器時，不會自動從已使用該連接器的已發佈應用程式中移除相關聯的秘密登入資料。在此案例中，如果應用程式已使用特定連接器發佈，且其中一個連接器已從 App Studio 中刪除，則已發佈的應用程式將繼續使用先前存放的連接器憑證。請務必注意，即使連接器刪除，已發佈的應用程式仍會保持不受影響和運作。

編輯已發佈應用程式上的資料連接器

編輯資料連接器時，變更不會自動反映在使用該連接器的已發佈應用程式中。如果應用程式已使用特定連接器發佈，且在 App Studio 中修改其中一個連接器，則發佈的應用程式將繼續使用先前存放的連接器組態和登入資料。若要納入更新的連接器變更，必須重新發佈應用程式。在應用程式重新發佈之前，它將保持不正確且不運作，或不受影響和運作，但不會反映最新的連接器修改。

安全風險緩解建議

本節列出緩解建議，以避免先前安全性考量一節中詳述的安全風險。

1. 適當的 IAM 角色組態：確保資料連接器的 IAM 角色已正確設定最低權限原則，以防止未經授權的存取和資料外洩。
2. 受限的應用程式存取：僅與有權檢視或對應用程式資料執行動作的使用者共用您的應用程式。
3. 應用程式發佈：確保每當連接器更新或刪除時，應用程式都會重新發佈。

AWS App Studio 中的資料保護

AWS [共同責任模型](#)適用於 AWS App Studio 中的資料保護。如此模型所述，AWS 負責保護執行所有的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權常見問答集](#)。如需

有關歐洲資料保護的相關資訊，請參閱AWS 安全性部落格上的[AWS 共同責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。如需有關使用 CloudTrail 追蹤擷取 AWS 活動的資訊，請參閱AWS CloudTrail 《使用者指南》中的[使用 CloudTrail 追蹤](#)。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-3](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 AWS App Studio 或使用主控台、API AWS CLI或其他 AWS 服務 AWS SDKs 時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

資料加密

App Studio 透過加密靜態和傳輸中的資料，安全地存放和傳輸資料。

靜態加密

靜態加密是指在存放時對資料進行加密，以保護您的資料免受未經授權的存取。App Studio 預設會使用 AWS KMS 金鑰提供靜態加密，您不需要為靜態資料加密進行任何額外的組態。

App Studio 會為您的應用程式安全地存放下列資料：原始碼、建置成品、中繼資料和許可資訊。

使用以 AWS KMS 客戶受管金鑰 (CMK) 加密的資料來源時，App Studio 資源會繼續使用 AWS 受管金鑰加密，而加密資料來源中的資料則會由 CMK 加密。如需在 App Studio 應用程式中使用加密資料來源的詳細資訊，請參閱 [搭配 CMKs 使用加密的資料來源](#)。

App Studio 使用 Amazon CloudFront 為您的使用者提供應用程式。CloudFront 使用了適用於節點連接點 (POP) 加密的 SSD 和區域邊緣快取 (REC) 加密的 EBS 磁碟區。CloudFront Functions 中的函數程

式碼和組態始終以加密格式儲存在節點 POP 上的加密 SSD 中，以及 CloudFront 使用的其他儲存位置中。

傳輸中加密

傳輸中的加密指的是保護您的資料免於在通訊端點間移動時遭到攔截。App Studio 預設為傳輸中的資料提供加密。客戶與 App Studio 之間，以及 App Studio 與其下游相依性之間的所有通訊，都會使用使用 Signature 第 4 版簽署程序簽署的 TLS 連線進行保護。所有 App Studio 端點都使用由 AWS Certificate Manager 私有憑證授權單位管理的 SHA-256 憑證。

金鑰管理

App Studio 不支援管理加密金鑰。

網際網路流量隱私權

當您在 App Studio 中建立執行個體時，您可以選擇存放該執行個體 AWS 資料和資源的區域。應用程式建置成品和中繼資料永遠不會離開該 AWS 區域。

不過，請注意下列資訊：

- 由於 App Studio 使用 Amazon CloudFront 為您的應用程式提供服務，並使用 Lambda@Edge 來管理應用程式的身分驗證，因此從 CloudFront 節點存取一組有限的身分驗證資料、授權資料、應用程式中繼資料，這些位置可能位於不同的區域。
- AWS App Studio 跨 AWS 區域傳輸資料，以在服務中啟用特定生成式 AI 功能。如需跨區域資料傳輸啟用的功能、跨區域移動的資料類型，以及如何選擇退出的詳細資訊，請參閱 [AWS App Studio 中的跨區域資料傳輸](#)。

AWS App Studio 和 AWS Identity and Access Management (IAM)

在 AWS App Studio 中，您可以透過將 IAM Identity Center 中的群組指派給 App Studio 中的適當角色，來管理服務的存取和許可。群組成員的許可取決於指派的角色，而不是直接在 AWS Identity and Access Management (IAM) 中設定使用者、角色或許可。如需在 App Studio 中管理存取和許可的詳細資訊，請參閱 [在 App Studio 中管理存取和角色](#)。

App Studio 在驗證執行個體進行計費時，以及連線到 AWS 帳戶以建立和使用該 AWS 帳戶中的資源時，會與 IAM 整合。如需將 App Studio 連線至其他 AWS 服務以在應用程式中使用的資訊，請參閱 [連線至 AWS 服務](#)。

當您在 App Studio 中建立執行個體時，您必須連接 AWS 帳戶做為執行個體的帳單和管理帳戶。為了啟用關鍵功能，App Studio 也會建立 [IAM 服務角色](#)，為服務提供必要的許可，以代表您執行任務。

AWS Identity and Access Management (IAM) 是一種 AWS 服務，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證（登入）和授權（具有許可）來使用 App Studio 資源。IAM 是您可以免費使用 AWS 服務的。

主題

- [App Studio 的身分型政策](#)
- [App Studio 中的資源型政策](#)
- [App Studio 的政策動作](#)
- [App Studio 的政策資源](#)
- [App Studio 的政策條件索引鍵](#)
- [App Studio ACLs](#)
- [ABAC 搭配 App Studio](#)
- [搭配 App Studio 使用臨時登入資料](#)
- [App Studio 的跨服務主體許可](#)
- [App Studio 的服務角色](#)
- [App Studio 的服務連結角色](#)
- [AWS App Studio 的受管政策](#)
- [App Studio 的服務連結角色](#)
- [AWS App Studio 的身分型政策範例](#)

在您使用 IAM 管理 App Studio 的存取權之前，請先了解哪些 IAM 功能可與 App Studio 搭配使用。

您可以搭配 AWS App Studio 使用的 IAM 功能

IAM 功能	App Studio 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是

IAM 功能	App Studio 支援
政策條件索引鍵	否
ACL	否
ABAC(政策中的標籤)	否
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	是

若要全面了解 App Studio 和其他 AWS 服務如何與大多數 IAM 功能搭配使用，請參閱 [《AWS IAM 使用者指南》](#) 中的 [與 IAM 搭配使用的服務](#)。

App Studio 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱 [《IAM 使用者指南》](#) 中的 [透過客戶管理政策定義自訂 IAM 許可](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱 [《IAM 使用者指南》](#) 中的 [IAM JSON 政策元素參考](#)。

App Studio 的身分型政策範例

若要檢視 App Studio 身分型政策的範例，請參閱 [AWS App Studio 的身分型政策範例](#)。

App Studio 中的資源型政策

支援資源型政策：否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源

的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中[指定主體](#)。委託人可以包含帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

如需啟用跨帳戶存取權，您可以在其他帳戶內指定所有帳戶或 IAM 實體作為資源型政策的主體。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 中的快帳戶資源存取](#)。

App Studio 的政策動作

支援政策動作：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。政策會使用動作來授予執行相關聯動作的許可。

若要查看 App Studio 動作的清單，請參閱服務授權參考中的[AWS App Studio 定義的動作](#)。

App Studio 中的政策動作在動作之前使用以下字首：

```
appstudio
```

若要在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
    "appstudio:action1",  
    "appstudio:action2"  
]
```

下列陳述式列出 App Studio 中的所有動作：

App Studio 的政策資源

支援政策資源：是

App Studio 許可僅支援政策 Resource 元素中的萬用字元 (*)。

App Studio 的政策條件索引鍵

支援服務特定政策條件金鑰：否

App Studio 不支援政策條件金鑰。

App Studio ACLs

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

ABAC 搭配 App Studio

支援 ABAC (政策中的標籤)：否

App Studio 不支援屬性型存取控制 (ABAC)。

搭配 App Studio 使用臨時登入資料

支援臨時憑證：是

臨時登入資料提供 AWS 資源的短期存取權，並在您使用聯合或切換角色時自動建立。AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 中的臨時安全憑證與可與 IAM 搭配運作的 AWS 服務](#)。

App Studio 的跨服務主體許可

支援轉寄存取工作階段 (FAS)：是

轉送存取工作階段 (FAS) 使用呼叫的委託人許可 AWS 服務，結合 AWS 服務請求向下游服務提出請求。如需提出 FAS 請求時的政策詳細資訊，請參閱 [轉發存取工作階段](#)。

App Studio 的服務角色

支援服務角色：是

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的 [建立角色以委派許可給 AWS 服務](#)。

AWS App Studio 對某些功能使用 [IAM 服務角色](#)，以授予 App Studio 代表您執行任務的許可。當您設定 App Studio 時，主控台會自動為支援的功能建立服務角色。

Warning

變更服務角色的許可可能會中斷 App Studio 功能。只有在 App Studio 提供指引時，才能編輯服務角色。

App Studio 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 `中` AWS 帳戶，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服务連結角色的詳細資訊，請參閱[可搭配 IAM 運作的AWS 服務](#)。在資料表中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

AWS AWS App Studio 的 受管政策

若要新增許可給使用者、群組和角色，使用 AWS 受管政策比自行撰寫政策更容易。建立 [IAM 客戶受管政策](#) 需要時間和專業知識，而受管政策可為您的團隊提供其所需的許可。若要快速開始使用，您可以使用我們的 AWS 受管政策。這些政策涵蓋常見的使用案例，並可在您的 AWS 帳戶中使用。如需 AWS 受管政策的詳細資訊，請參閱《IAM 使用者指南》中的 [AWS 受管政策](#)。

AWS 服務會維護和更新 AWS 受管政策。您無法變更 AWS 受管政策中的許可。服務偶爾會在 AWS 受管政策中新增其他許可以支援新功能。此類型的更新會影響已連接政策的所有身分識別 (使用者、群組和角色)。當新功能啟動或新操作可用時，服務很可能會更新 AWS 受管政策。服務不會從 AWS 受管政策中移除許可，因此政策更新不會破壞您現有的許可。

此外，AWS 支援跨多個服務之任務函數的受管政策。例如，ReadOnlyAccess AWS 受管政策提供所有 AWS 服務和資源的唯讀存取權。當服務啟動新功能時，會為新操作和資源 AWS 新增唯讀許可。如需任務職能政策的清單和說明，請參閱 IAM 使用者指南中 [有關任務職能的AWS 受管政策](#)。

AWS 受管政策：AppStudioServiceRolePolicy

您不得將 AppStudioServiceRolePolicy 連接到 IAM 實體。此政策會連接到服務連結角色，允許 App Studio 代表您執行動作。如需詳細資訊，請參閱[App Studio 的服務連結角色](#)。

此政策會授予許可，允許服務連結角色管理 AWS 資源。

許可詳細資訊

此政策包含執行以下動作的許可：

- logs - 建立 CloudWatch 日誌群組和日誌串流。也提供在這些日誌群組和串流中建立日誌事件的許可。
- secretsmanager - 建立、讀取、更新和刪除由 App Studio 管理的受管秘密。
- sso - 擷取應用程式執行個體。
- sso-directory - 擷取使用者的相關資訊，並擷取群組中的成員清單。

JSON

```
{
  "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}"
        }
      }
    }
  ]
}

```

AWS 受管政策的 App Studio 更新

檢視自此服務開始追蹤這些變更以來，App Studio AWS 受管政策更新的詳細資訊。

變更	描述	Date
AppStudioServiceRolePolicy – 更新至現有政策	App Studio 新增了允許管理 App Studio 受管秘密的許可 AWS Secrets Manager。	2025 年 3 月 14 日
App Studio 已開始追蹤變更	App Studio 開始追蹤其 AWS 受管政策的變更。	2024 年 6 月 28 日

App Studio 的服務連結角色

App Studio 使用 [AWS Identity and Access Management \(IAM\) 服務連結角色](#)。服務連結角色是直接連結至 App Studio 的唯一 IAM 角色類型。服務連結角色由 App Studio 預先定義，並包含該服務代表您呼叫其他 AWS 服務所需的所有許可。

服務連結角色可讓您更輕鬆地設定 App Studio，因為您不必手動新增必要的許可。App Studio 定義其服務連結角色的許可，除非另有定義，否則只有 App Studio 可以擔任其角色。定義的許可包括信任政策和許可政策，且該許可政策無法附加至其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。這可保護您的 App Studio 資源，因為您不會不小心移除存取資源的許可。

目錄

- [App Studio 的服務連結角色許可](#)
- [為 App Studio 建立服務連結角色](#)
- [編輯 App Studio 的服務連結角色](#)
- [刪除 App Studio 的服務連結角色](#)

App Studio 的服務連結角色許可

App Studio 使用名為的服務連結角色 `AWSServiceRoleForAppStudio`。這是 App Studio 持續管理 AWS 服務以維護應用程式建置體驗所需的服務連結角色。

`AWSServiceRoleForAppStudio` 服務連結角色使用以下僅信任 `appstudio-service.amazonaws.com` 服務的信任政策：

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appstudio-service.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

對於許可，`AWSServiceRoleForAppStudio` 服務連結角色會提供下列服務的許可：

- Amazon CloudWatch：傳送 App Studio 用量的日誌和指標。
- AWS Secrets Manager：管理 App Studio 中連接器的登入資料，用於將應用程式連線至其他服務。
- IAM Identity Center：唯讀存取以管理使用者存取。

具體而言，透過授予的許可 `AWSServiceRoleForAppStudio` 由連接的 `AppStudioServiceRolePolicy` 受管政策定義。如需受管政策的詳細資訊，包括其包含的許可，請參閱 [AWS 受管政策：AppStudioServiceRolePolicy](#)。

為 App Studio 建立服務連結角色

您不需要手動建立服務連結角色，當您建立 App Studio 執行個體時，App Studio 會為您建立服務連結角色。

如果您刪除此服務連結角色，建議您建立 App Studio 執行個體，以便為您自動建立另一個執行個體。

雖然不是不必要的，但您也可以使用 IAM 主控台 AWS CLI 或建立服務連結角色，方法是使用 `appstudio-service.amazonaws.com` 服務名稱建立服務連結角色，如先前顯示的信任政策程式碼片段所示。如需詳細資訊，請參閱《IAM 使用者指南》中的「[建立服務連結角色](#)」。

編輯 App Studio 的服務連結角色

App Studio 不允許您編輯 `AWSServiceRoleForAppStudio` 服務連結角色。因為可能有各種實體會參考服務連結角色，所以您無法在建立角色之後變更其名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱《IAM 使用者指南》中的 [編輯服務連結角色](#)。

刪除 App Studio 的服務連結角色

您不需要刪除 `AWSServiceRoleForAppStudio` 角色。當您刪除 App Studio 執行個體時，App Studio 會清除資源，並自動刪除服務連結角色。

雖然不建議，但您可以使用 IAM 主控台或 AWS CLI 來刪除服務連結角色。若要這樣做，您必須先清除服務連結角色的資源，然後才能將其刪除。

Note

如果 App Studio 在您嘗試刪除資源時使用角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。

使用 IAM 手動刪除服務連結角色

1. 從 App Studio 執行個體刪除應用程式和連接器。
2. 使用 IAM 主控台、IAM CLI 或 IAM API 刪除 `AWSServiceRoleForAppStudio` 服務連結角色。
如需詳細資訊，請參閱《IAM 使用者指南》中的[刪除服務連結角色](#)。

AWS App Studio 的身分型政策範例

根據預設，使用者和角色沒有建立或修改 App Studio 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策 \(主控台\)](#)。

如需 App Studio 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱服務授權參考中的[AWS App Studio 的動作、資源和條件金鑰](#)。

主題

- [政策最佳實務](#)
- [使用 App Studio 主控台](#)
- [允許使用者檢視他們自己的許可](#)
- [範例 1：允許使用者設定 App Studio 執行個體](#)
- [範例 2：拒絕使用者設定 App Studio 執行個體](#)

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 App Studio 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的[AWS 受管政策](#)或[任務職能的AWS 受管政策](#)。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的[IAM 中的政策和許可](#)。

- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 例如 使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [使用 IAM Access Analyzer 驗證政策](#)。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [透過 MFA 的安全 API 存取](#)。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 [IAM 安全最佳實務](#)。

使用 App Studio 主控台

若要存取 AWS App Studio 主控台，您必須擁有一組最低許可。這些許可必須允許您列出和檢視 中 App Studio 資源的詳細資訊 AWS 帳戶。如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

為了確保使用者和角色仍然可以使用 App Studio 主控台，請將 App Studio *ConsoleAccess* 或 *ReadOnly* AWS 受管政策連接到實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台或使用 或 AWS CLI AWS API 以程式設計方式完成此動作的許可。

```
{
  "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": "*"
}
]
}

```

範例 1：允許使用者設定 App Studio 執行個體

下列範例顯示以身分為基礎的政策，以允許角色設定 App Studio 執行個體。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "appstudio:GetAccountStatus",
      "appstudio:GetEnablementJobStatus",
      "appstudio:StartEnablementJob",
      "appstudio:StartRollbackEnablementJob",
      "appstudio:StartTeamDeployment"
    ]
  }],

```

```
    "Resource": "*"
  }]
}
```

範例 2：拒絕使用者設定 App Studio 執行個體

下列範例顯示身分型政策，以拒絕角色設定 App Studio 執行個體。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Deny",
    "Action": [
      "appstudio:*"
    ],
    "Resource": "*"
  }]
}
```

AWS App Studio 的合規驗證

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱[AWS 服務 合規計劃範圍內](#)然後選擇您感興趣的合規計劃。如需一般資訊，請參閱[AWS 合規計劃](#)。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱[下載報告 in AWS Artifact](#)

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 [AWS 安全文件](#)。

AWS App Studio 中的彈性

AWS 全球基礎設施是以 AWS 區域 和 可用區域 為基礎建置。AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 [AWS 全球基礎設施](#)。

除了 AWS 全球基礎設施之外，AWS App Studio 還提供數種功能，以協助支援您的資料彈性和備份需求。

AWS App Studio 中的基礎設施安全

作為受管服務，AWS App Studio 受到 [Amazon Web Services：安全程序概觀](#) 白皮書中所述的 AWS 全球網路安全程序的保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 App Studio。用戶端必須至少支援 Transport Layer Security (TLS) 1.2，但建議使用 TLS 1.3。用戶端也必須支援具備完美轉送私密 (PFS) 的密碼套件，例如臨時 Diffie-Hellman (DHE) 或橢圓曲線臨時 Diffie-Hellman (ECDHE)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

AWS App Studio 中的組態和漏洞分析

組態和 IT 控制是 AWS 與身為我們客戶的您共同的責任。如需詳細資訊，請參閱 AWS [共同的責任模型](#)。

預防跨服務混淆代理人

混淆代理人問題屬於安全性問題，其中沒有執行動作許可的實體可以強制具有更多許可的實體執行該動作。在中 AWS，跨服務模擬可能會導致混淆代理人問題。在某個服務 (呼叫服務) 呼叫另一個服務 (被呼叫服務) 時，可能會發生跨服務模擬。可以操縱呼叫服務來使用其許可，以其不應有存取許可的方式對其他客戶的資源採取動作。為了預防這種情況，AWS 提供的工具可協助您保護所有服務的資料，而這些服務主體已獲得您帳戶中資源的存取權。

我們建議在資源政策中使用 [aws:SourceArn](#) 和 [aws:SourceAccount](#) 全域條件內容索引鍵，以限制將另一個服務提供給資源的許可。如果您想要僅允許一個資源與跨服務存取相關聯，則請使用 `aws:SourceArn`。如果您想要允許該帳戶中的任何資源與跨服務使用相關聯，請使用 `aws:SourceAccount`。

防範混淆代理人問題的最有效方法是使用 `aws:SourceArn` 全域條件內容索引鍵，以及資源的完整 ARN。如果不知道資源的完整 ARN，或者如果您指定了多個資源，請使用

`aws:SourceArn` 全域內容條件索引鍵搭配萬用字元 (*) 來表示 ARN 的未知部分。例如 `arn:aws:servicename:*:123456789012:*`。

如果 `aws:SourceArn` 值不包含帳戶 ID (例如 Amazon S3 儲存貯體 ARN)，您必須使用這兩個全域條件內容索引鍵來限制許可。

`aws:SourceArn` 的值必須是 `ResourceDescription`。

下列範例示範如何在 中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全域條件內容索引鍵，以防止混淆代理人問題。

AWS App Studio 中的跨區域資料傳輸

AWS App Studio 跨 AWS 區域傳輸資料，以在服務中啟用特定生成式 AI 功能。本主題包含跨區域資料傳輸啟用的功能、跨區域移動的資料類型，以及如何選擇退出的相關資訊。

下列功能由跨區域資料傳輸啟用，如果您選擇退出，則無法在執行個體中存取：

1. 使用 AI 建立應用程式，透過使用自然語言描述應用程式並為您建立資源，來開始應用程式建置。
2. 應用程式工作室中的 AI 聊天，用於詢問有關應用程式建置、發佈和共用的問題。

下列資料會跨 區域傳輸：

1. 來自上述功能的提示或使用者輸入。

若要選擇退出跨區域資料傳輸及其啟用的功能，請使用下列程序從主控台填寫選擇退出申請表：

1. 在 <https://console.aws.amazon.com/appstudio/> 開啟 App Studio 主控台。
2. 選擇選擇退出資料傳輸。
3. 輸入 AWS 您的帳戶 ID，並提供您的電子郵件地址。
4. 選擇提交。
5. 提交後，系統會處理您選擇退出跨區域資料傳輸的請求，最多可能需要 60 天。

AWS App Studio 支援的瀏覽器

本主題包含有關 AWS App Studio 支援和建議瀏覽器的資訊，包括存取已發佈應用程式的最終使用者和應用程式建置器的瀏覽器支援。

建置應用程式的支援和建議的瀏覽器

為了獲得最佳的應用程式建置體驗，App Studio 支援並強烈建議使用 Google Chrome。

Note

雖然不建議，但您也可以使用其他熱門 Web 瀏覽器建置應用程式，例如 Mozilla Firefox、Microsoft Edge 或 Apple Safari for MacOS，但請注意，這些瀏覽器並未正式支援或驗證，您可能需要更新設定以存取某些建置器功能。如需詳細資訊，請參閱[更新瀏覽器設定以在 App Studio 上建置應用程式](#)。

App Studio 不支援從行動平台建置應用程式。

應用程式最終使用者的支援和建議的瀏覽器

對於存取已發佈應用程式的最終使用者，App Studio 強烈建議使用 Google Chrome 或 Mozilla Firefox。雖然這些是建議的瀏覽器，但最終使用者可以使用其他熱門 Web 瀏覽器存取發佈的應用程式，例如 Microsoft Edge 或 Apple Safari for MacOS。

最終使用者可以從行動平台存取已發佈的應用程式。

更新瀏覽器設定以在 App Studio 上建置應用程式

App Studio 正式支援和建議使用 Google Chrome 來建置應用程式。不過，如果您想要使用其他瀏覽器來建置應用程式，您可能需要更新與跨網站追蹤相關的特定設定或 Cookie，才能存取 App Studio 中的特定頁面。

對於 Mozilla Firefox：若要預覽應用程式，請將下列設定更新 Firefox Settings > Privacy & Security > Enhanced Tracking Protection 為 Custom > Cookies > Cross-site tracking cookies。

對於 Apple Safari for MacOS：若要建置或預覽應用程式，請停用下列設定：Settings > Privacy > Prevent cross-site tracking。

AWS App Studio 的配額

下表說明 AWS App Studio 的配額和限制。

App Studio 執行個體中的應用程式數量上限	20
App Studio 執行個體中發佈至測試或生產環境的應用程式數量上限。發佈至測試和生產的單一應用程式會視為兩個已發佈的應用程式。	6
每個應用程式的受管實體數量上限	20
每個查詢傳回的資料列數目上限	3000
每個實體的範例資料列數上限	500
自動化的最長執行時間	2 分鐘。執行超過 2 分鐘的自動化將會失敗。
自動化輸入和輸出大小上限	每個輸入或輸出 5GB。
自動化或資料動作使用的資料大小上限	每個自動化或資料動作執行 450MB。
頁面名稱和元件名稱	必須是非空白且唯一的。只能包含字母、數字底線 (_) 和貨幣符號 (\$)。不能包含空格。

AWS App Studio 使用者指南的文件歷史記錄

下表說明 AWS App Studio 的文件版本。

變更	描述	日期
新主題：轉返應用程式的生產環境版本	新增在偵測到問題時，將應用程式已發佈版本復原至先前發佈版本的相關資訊。如需詳細資訊，請參閱 回復至先前發佈的版本 。	2025 年 4 月 10 日
新主題：重複的應用程式、頁面和元件	新增了有關在 App Studio 中複製應用程式、頁面和元件的資訊的新主題。如需詳細資訊，請參閱 複製應用程式 、 複製頁面 和 複製元件 。	2025 年 4 月 7 日
新主題：在 App Studio 執行個體之間匯入和匯出應用程式	新增了有關在 App Studio 執行個體之間匯入和匯出應用程式的資訊主題，包括 App Studio 提供的可匯入應用程式清單，可用於學習應用程式建置概念。如需詳細資訊，請參閱 匯入應用程式 和 匯出應用程式 。	2025 年 3 月 30 日
更新主題：AWS 受管政策：AppStudioServiceRolePolicy	已更新每個服務的AppStudio ServiceRolePolicy 許可並新增政策描述資訊。如需詳細資訊，請參閱 AWS 受管政策：AppStudioServiceRolePolicy 。	2025 年 3 月 14 日
更新主題：編輯或設定資料動作	新增了資料動作條件中使用的現有運算子和新運算子的相關資訊，這些運算子用於從您的資料庫資料表擷取符合條件的	2025 年 2 月 27 日

	<p>資料子集。如需詳細資訊，請參閱編輯或設定資料動作。</p>	
更新主題：使用 JavaScript 撰寫表達式	<p>已重新組織和新增有關在應用程式中使用表達式參考或更新 UI 元件和資料表資料的資訊。如需詳細資訊，請參閱使用 JavaScript 在 App Studio 中寫入表達式。</p>	2025 年 2 月 18 日
更新主題：應用程式內容安全設定	<p>新增內容來源應用程式內容安全設定的相關資訊。您可以使用此設定來限制您的應用程式可將物件上傳至 Amazon S3 的網域。如需詳細資訊，請參閱檢視或更新應用程式的內容安全設定。</p>	2025 年 2 月 14 日
新主題：在 App Studio 應用程式中叫用 Lambda 函數	<p>新增了簡短的教學課程，詳細說明如何在 App Studio 應用程式中叫用 Lambda 函數。如需詳細資訊，請參閱調用 Lambda 函數。</p>	2025 年 1 月 24 日
新主題：連線至 Amazon SES	<p>新增建立 Amazon SES 連接器以在 App Studio 應用程式中使用服務的指示。如需詳細資訊，請參閱連線至 Amazon Simple Email Service。</p>	2025 年 1 月 16 日
更新主題：第一次建立和設定 App Studio 執行個體	<p>新增使用輕鬆建立方法來建立 App Studio 執行個體以更快速開始使用的指示。如需詳細資訊，請參閱第一次建立和設定 App Studio 執行個體。</p>	2024 年 12 月 13 日

新主題：管理資料相依性和時間問題的最佳實務	新增了有關在 App Studio 應用程式中正常管理資料相依性和計時問題的文件。如需詳細資訊，請參閱 資料相依性和時間考量 。	2024 年 11 月 20 日
更新主題：使用 AI 編輯您的應用程式	新增文件，其中包含使用應用程式工作室中的 AI 聊天編輯應用程式的相關資訊。如需詳細資訊，請參閱 使用生成式 AI 建置 App Studio 應用程式 。	2024 年 11 月 18 日
更新主題：使用 AI 為您產生 JavaScript	更新 JavaScript 自動化動作參考，以包含使用 AI 為您產生 JavaScript 的相關資訊。如需詳細資訊，請參閱 JavaScript 自動化動作 。	2024 年 11 月 18 日
更新主題：使用 Amazon Bedrock 建置 AI 文字摘要器應用程式	已更新 Amazon Bedrock 提示教學課程，以使用新發行的 GenAI 提示動作。如需詳細資訊，請參閱 使用 Amazon Bedrock 建置 AI 文字摘要器應用程式 。	2024 年 11 月 18 日
新主題：使用應用程式主題變更應用程式的顏色	新增主題，其中包含使用應用程式主題變更應用程式中顏色的相關資訊。如需詳細資訊，請參閱 使用應用程式主題變更應用程式中的顏色 。	2024 年 11 月 18 日
新主題：資料模型最佳實務	新增了具有最佳實務的主題，用於建立安全、強大且可擴展的資料模型，以用於 App Studio 應用程式。如需詳細資訊，請參閱 設計資料模型時的最佳實務 。	2024 年 11 月 15 日

更新主題：連線至 AWS 服務	更新信任政策以包含 <code>sts:ExternalId</code> ，這是用於建立 AWS 服務連接器之 IAM 角色的必要項目。如需詳細資訊，請參閱 連線至 AWS 服務 。	2024 年 11 月 13 日
新主題：復原或還原至先前發佈的應用程式版本	新增了主題，其中包含有關復原或還原應用程式至先前發佈版本的資訊。如需詳細資訊，請參閱 回復至先前發佈的版本 。	2024 年 11 月 13 日
新主題：刪除 App Studio 執行個體	新增主題，其中包含刪除 App Studio 執行個體的相關資訊，包括如何刪除執行個體的指示。如需詳細資訊，請參閱 刪除 App Studio 執行個體 。	2024 年 11 月 12 日
新主題：更新應用程式內容安全設定	新增主題，其中包含 App Studio 中應用程式內容安全設定的相關資訊，包括如何更新這些設定。如需詳細資訊，請參閱 檢視或更新應用程式的內容安全設定 。	2024 年 11 月 8 日
更新主題：AWS App Studio 中的安全性	擴充安全性文件，包括有關資料保護以及 App Studio 如何與 IAM 互動的資訊。如需詳細資訊，請參閱 AWS App Studio 中的安全性 。	2024 年 11 月 6 日
更新主題：AWS App Studio 中的配額	更新 App Studio 服務配額和限制文件，以修正不正確的值並移除一些配額。如需詳細資訊，請參閱 AWS App Studio 中的配額 。	2024 年 10 月 21 日

更新主題：將 App Studio 連線至其他 AWS 服務	更新了連線至 AWS 服務的文件，以提供指示和範例，以將服務或資源的必要許可降至最低，從而更好地遵守最佳安全實務。如需詳細資訊，請參閱 連線至 AWS 服務 。	2024 年 10 月 18 日
更新主題：新增 Aurora 連接器文件的版本支援	新增支援版本的清單至 Aurora 連接器文件。如需詳細資訊，請參閱 連線至 Amazon Aurora 。	2024 年 10 月 16 日
新主題：App Studio 支援的瀏覽器	新增主題，其中包含瀏覽器支援和使用 App Studio 的建議。如需詳細資訊，請參閱 支援的瀏覽器 。	2024 年 10 月 10 日
新主題：AWS App Studio 的運作方式	新增主題，逐步解說 App Studio 中應用程式開發的重要概念，包括圖表和螢幕擷取畫面。如需詳細資訊，請參閱 App Studio 的運作方式 。	2024 年 10 月 10 日
新主題：排序和組織頁面	新增了主題，其中包含有關重新排序和隱藏或在預覽或已發佈應用程式的導覽中顯示頁面的資訊。如需詳細資訊，請參閱 排序和組織頁面 。	2024 年 9 月 24 日
新主題：AWS App Studio 中的配額	新增了主題，其中包含與 App Studio 相關的服務配額和限制。如需詳細資訊，請參閱 AWS App Studio 中的配額 。	2024 年 9 月 11 日

[更新主題：連線至加密的
DynamoDB 資料表](#)

新增使用以 AWS KMS 客戶受管金鑰 (CMKs) 加密的 DynamoDB 資料表搭配 App Studio 的資訊，例如必要的許可。如需詳細資訊，請參閱[連線至 DynamoDB](#)。

2024 年 9 月 6 日

[更新主題：連線至
DynamoDB、Amazon Redshift
和 Aurora](#)

新增將新增至 IAM 角色的最低必要許可，以搭配 App Studio 應用程式使用 DynamoDB、Amazon Redshift 和 Aurora 資源。如需詳細資訊，請參閱[連線至 AWS 服務](#)。

2024 年 9 月 5 日

[更新主題：連線至 Amazon
Aurora](#)

更新建立和設定 Amazon Aurora 資料庫和資料表以搭配 App Studio 應用程式使用的文件。如需詳細資訊，請參閱[連線至 Amazon Aurora](#)。

2024 年 9 月 5 日

[新增和更新的主題：故障診斷
和偵錯](#)

擴充疑難排解和偵錯文件，以協助解決 App Studio 的常見問題，包括建置應用程式的偵錯資訊。如需詳細資訊，請參閱[對 App Studio 進行故障診斷和偵錯](#)。

2024 年 8 月 26 日

[新主題：教學課程：使用
Amazon Bedrock 建置 AI 文字
摘要器應用程式](#)

依照教學課程中的步驟來建置應用程式，以接收最終使用者的輸入提示、將其傳送至 Amazon Bedrock，並傳回和顯示摘要版本。如需詳細資訊，請參閱[使用 Amazon Bedrock 建置 AI 文字摘要器應用程式](#)。

2024 年 8 月 20 日

更新主題：預覽、發佈和共用 App Studio 應用程式	擴展預覽、發佈和共用文件以增加清晰度、符合服務體驗，並提供有關發佈環境和檢視其中應用程式的其他資訊。如需詳細資訊，請參閱 預覽、發佈和共用應用程式 。	2024 年 8 月 2 日
新主題：使用多個使用者建置應用程式	擴展預覽、發佈和共用文件以增加清晰度、符合服務體驗，並提供有關發佈環境和檢視其中應用程式的其他資訊。如需詳細資訊，請參閱 使用多個使用者建置應用程式 。	2024 年 8 月 2 日
更新主題：將 App Studio 連線至 AWS 服務	新增有關建立和提供 IAM 角色的資訊，以便在建立其他服務 AWS 連接器時提供 AWS 資源的存取權。如需詳細資訊，請參閱 AWS 使用其他服務連接器連線至 AWS 服務 。	2024 年 7 月 29 日
更新主題：新增建立 AWS 管理使用者的指示，做為設定的一部分	新增 設定 App Studio 文件中的指示，以建立管理使用者來管理 AWS 資源。在整個連接器文件中也進行了更新，以建議使用該使用者。	2024 年 7 月 24 日
新主題：連線至 Amazon Bedrock	新增了主題，其中包含為 Amazon Bedrock 建立連接器的說明。建置器可以使用連接器來建置使用 Amazon Bedrock 的應用程式。如需詳細資訊，請參閱 連線至 Amazon Bedrock 。	2024 年 7 月 24 日
初始版本	AWS App Studio 使用者指南的初始版本	2024 年 7 月 10 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。