



開發人員指南

AWS HealthLake



AWS HealthLake: 開發人員指南

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 HealthLake ?	1
重要通知	2
功能	2
相關服務	3
存取	3
HIPAA	4
定價	4
開始使用	5
概念	5
授權策略	5
整合式 NLP	6
整合式分析	6
設定	6
註冊 AWS 帳戶	7
設定 IAM 使用者或角色	7
新增 Data Lake 管理員使用者或角色	9
建立 S3 儲存貯體	10
建立資料存放區	10
設定匯入許可	11
設定匯出許可	13
安裝 AWS CLI	17
教學課程	17
管理資料存放區	19
建立資料存放區	19
取得資料存放區屬性	27
列出資料存放區	30
更新資料存放區	34
標記資料存放區	37
標記資料存放區	38
列出資料存放區的標籤	41
取消標記資料存放區	44
刪除資料存放區	47
管理 FHIR 訂閱	51
FHIR 訂閱的運作方式	51

關鍵元件	51
訂閱主題	51
Subscriptions	52
通知管道	52
通知承載	53
最佳實務	53
訂閱生命週期	54
建立訂閱	55
範例訂閱承載	58
通知承載範例	63
搜尋訂閱	67
篩選通知	70
匯入 FHIR 資料	73
啟動匯入任務	75
取得匯入任務屬性	79
列出匯入任務	83
管理 FHIR 資源	89
建立資源	90
讀取資源	94
讀取資源歷史記錄	95
讀取版本特定的歷史記錄	98
更新資源	100
條件式更新	102
設定資源更新的驗證層級	103
修改資源	104
支援的修補程式格式	104
Usage	104
JSON 修補程式格式	105
FHIRPath 修補程式格式	108
請求標頭	110
回應範例	110
Behavior (行為)	111
錯誤處理	111
功能摘要	111
限制	112
其他資源	112

綁定資源	112
做為獨立實體的套件	116
條件式 PUTs	120
做為單一實體的套件	124
設定套件的驗證層級	126
對 Bundle 類型 "message" 的有限支援	128
非同步交易	129
刪除資源	138
FHIR 的條件式刪除	139
冪等和並行	141
冪等性金鑰	141
AWS HealthLake 中的 ETag	143
搜尋 FHIR 資源	145
使用 GET 搜尋	145
GET 搜尋範例	147
使用 POST 搜尋	149
POST 搜尋範例	151
搜尋一致性層級	153
一致性層級	153
使用範例	153
最佳實務	154
匯出 FHIR 資料	155
啟動匯出任務	155
取得匯出任務屬性	160
列出匯出任務	164
程式碼範例	169
基本概念	169
動作	170
整合	216
自然語言處理	216
NLP 程式庫	217
使用 FHIR APIs	218
搜尋參數	219
請求範例	221
SQL 索引和查詢	237
開始使用	237

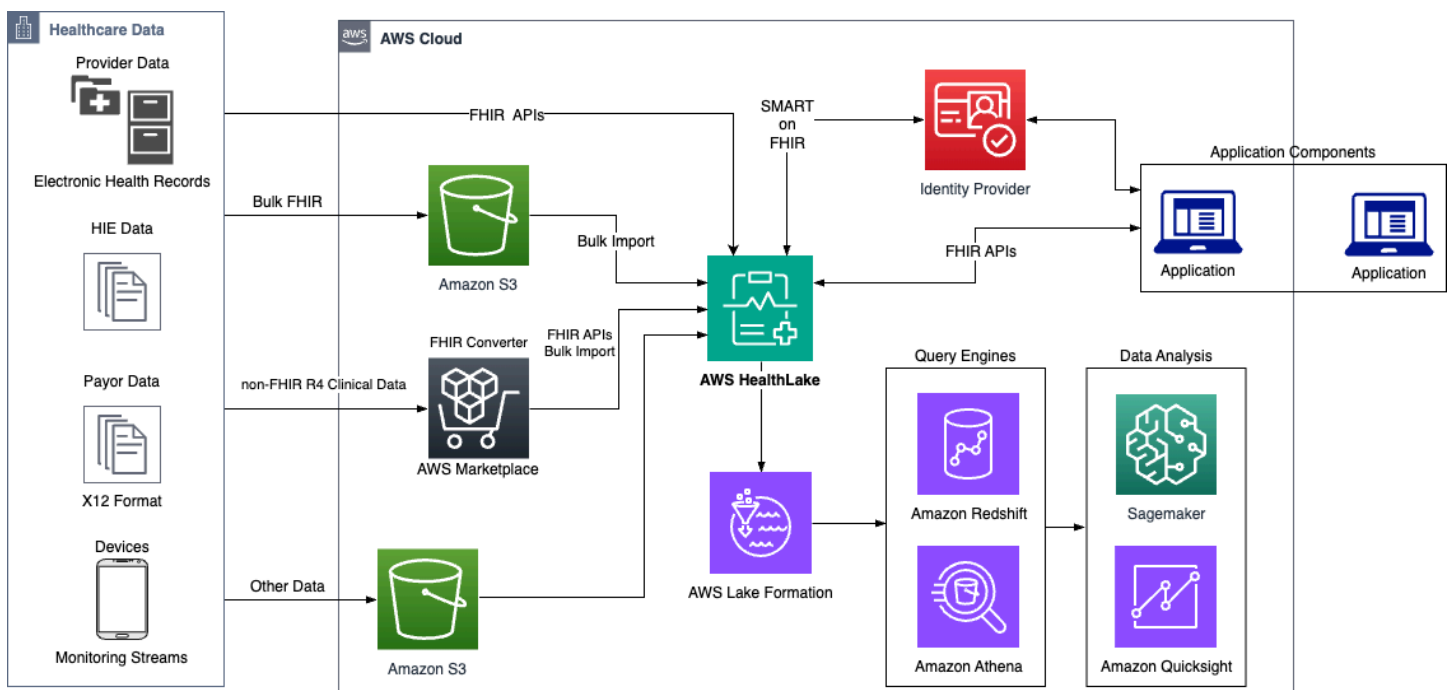
使用 SQL 查詢	240
查詢範例	247
監控	253
CloudTrail (API 呼叫)	253
AWS HealthLake CloudTrail 中的資訊	254
了解 AWS HealthLake 日誌檔案項目	255
CloudWatch (指標)	256
檢視 HealthLake 指標	259
建立 警示	259
EventBridge (事件)	260
傳送至 EventBridge 的 HealthLake 事件	260
HealthLake 事件結構	261
安全	275
資料保護	275
靜態加密	276
AWS 擁有的 KMS 金鑰	276
客戶受管的 KMS 金鑰	277
建立客戶自管金鑰	277
使用客戶受管 KMS 金鑰所需的 IAM 許可	278
傳輸中加密	285
身分與存取管理	285
目標對象	285
使用身分驗證	286
使用政策管理存取權	287
如何使用 IAM 搭配 AWS HealthLake	288
身分型政策範例	292
AWS 受管政策	295
疑難排解	299
法規遵循驗證	301
基礎設施安全性	301
基礎設施即程式碼	301
HealthLake 和 CloudFormation 範本	301
進一步了解 CloudFormation	302
VPC 端點	302
HealthLake VPC 端點的考量事項	302
為 HealthLake 建立介面 VPC 端點 ;	303

為 HealthLake 建立 VPC 端點政策	303
最佳實務	304
恢復能力	304
參考資料	305
FHIR 上的 SMART	305
開始使用	306
身分驗證	308
OAuth 2.0 範圍	309
字符驗證	317
精細授權	327
探索文件	328
請求範例	329
FHIR R4	330
功能陳述式	330
設定檔驗證	331
資源類型	338
搜尋參數	340
\$操作	351
合規	490
CMS	490
HealthLake	495
端點和配額	496
預先載入的資料類型	504
專案範例	505
疑難排解	506
使用 AWS SDKs	513
推出	515
.....	dxxxii

什麼是 AWS HealthLake ？

AWS HealthLake 是一項符合 HIPAA 資格的服務，可使用 Fast Healthcare 互通性資源 (FHIR) R4 規格，在雲端中儲存、分析和共用運作狀態資料。HealthLake 使用案例包括：

- 企業運作狀態資料 – 直接從 管理和共用 FHIR R4 運作狀態資料，AWS 雲端 同時保持高效能和可用性。
- 醫療保健互通性 – 透過全受管 FHIR 資料存放區，支援客戶符合 21st Century Cures Act 進行病患存取。
- 自然語言處理 (NLP) – 利用整合的 NLP 模型，從非結構化的運作狀態資料中擷取有意義的醫療資訊。
- 多模態分析 – 結合 HealthLake 資料與 AWS HealthImaging 資料 AWS HealthOmics，以提供精準醫學的洞見。



主題

- [重要通知](#)
- [的功能 AWS HealthLake](#)
- [相關 AWS 服務](#)
- [存取 AWS HealthLake](#)

- [HIPAA 資格和資料安全](#)
- [定價](#)

重要通知

AWS HealthLake 不能取代專業醫療建議、診斷或治療，也不能用於治愈、治療、緩解、預防或診斷任何疾病或健康狀況。您有責任將人工審核作為任何用途的一部分 AWS HealthLake，包括與旨在告知臨床決策的任何第三方產品相關。只有在經過 AWS HealthLake 培訓的醫療專業人員採用健全的醫療判斷進行審核後，才應該用於患者護理或臨床案例。

的功能 AWS HealthLake

AWS HealthLake 提供下列功能。

匯入 FHIR R4 運作狀態資料

透過 HealthLake 原生匯入動作，您可以輕鬆地將 FHIR 資料從 Amazon S3 儲存貯體遷移到 HealthLake 資料存放區，包括臨床備註、實驗室報告、保險索賠等。HealthLake 支援用於醫療保健資料交換的 FHIR R4 規格。如有需要，您可以與 [AWS HealthLake 合作夥伴](#) 合作，將運作狀態資料轉換為 FHIR R4 格式。

以安全、合規且可稽核的方式存放運作狀態資料

HealthLake 資料存放區有助於編製運作狀態資料的索引，以便查詢資料。資料存放區會依時間順序建立每位患者醫療歷史記錄的完整檢視，並使用 FHIR R4 規格促進資訊交換。而且，它始終執行，讓您的索引保持最新狀態，讓您能夠使用標準 FHIR R4 互動搭配持久的主要儲存體和索引擴展，隨時搜尋資訊。

利用交易 FHIR 伺服器

利用 FHIR APIs 進行標準資源驗證、SMART on FHIR 授權和大量資料 FHIR API 匯出功能，以支援統一和分析您的資料，以降低營運成本並改善決策。HealthLake 支援客戶符合最新的 ONC 和 CMS 法規標準，包括：HL7 FHIR R4 APIs、FHIR 大量資料存取、US Core IG STU、HL7 SMART App Launch Framework IG、OAuth 2.0 和 OpenID Connect。

使用 NLP 轉換非結構化醫療資料

整合醫療自然語言處理 (NLP) 會轉換 HealthLake 資料存放區中的所有原始醫療文字資料，以了解並從非結構化醫療資料中擷取有意義的資訊。透過整合式醫療 NLP，您可以自動從醫療文字中擷取

實體、實體關係、實體特徵和受保護醫療資訊 (PHI)。NLP 擷取的實體會以原生 FHIR R4 資源形式儲存在 HealthLake 資料存放區中，並且可以透過 FHIR R4 APIs 或 Amazon Athena (SQL) 存取。

相關 AWS 服務

AWS HealthLake 與其他 AWS 服務緊密整合。了解下列服務有助於充分利用 HealthLake。

- [AWS Identity and Access Management](#) – 使用 IAM 安全地管理身分和對 HealthLake 資源的存取。
- [Amazon Simple Storage Service](#) – 使用 Amazon S3 做為暫存區域，將 DICOM 資料匯入 HealthLake。
- [AWS CloudTrail](#) – 使用 CloudTrail 追蹤 HealthLake 使用者活動和 API 用量。
- [Amazon CloudWatch](#) – 使用 CloudWatch 觀察和監控 HealthLake 資源。
- [AWS CloudFormation](#) – 使用 CloudFormation 實作基礎設施即程式碼 (IaC) 範本，以在 HealthLake 中建立資源。
- [AWS PrivateLink](#) – 使用 Amazon VPC 在 HealthLake 和 [Amazon Virtual Private Cloud](#) 之間建立連線，而不會將資料公開至網際網路。
- [Amazon EventBridge](#) – 使用 EventBridge 透過建立將 HealthLake 事件路由至目標的規則來建立可擴展的事件驅動型應用程式。
- [AWS Lake Formation](#) – 使用 Lake Formation 集中管理、保護和共用 HealthLake 資料，以進行分析和機器學習。
- [Amazon Athena](#) – 使用 Athena 查詢具有 SQL 的 HealthLake 資料，以允許進行更深入的分析。

存取 AWS HealthLake

您可以使用 AWS HealthLake AWS 管理主控台 AWS Command Line Interface 和 AWS SDKs 存取。本指南提供 AWS 管理主控台 和 AWS CLI SDKs 程式碼 AWS 範例的程序說明。

AWS Command Line Interface (AWS CLI)

為廣泛的 AWS 產品 AWS CLI 提供命令，並支援 Windows、Mac 和 Linux。如需詳細資訊，請參閱 [「AWS Command Line Interface 使用者指南」](#)。

AWS SDKs

AWS SDKs 為軟體開發人員提供程式庫、程式碼範例和其他資源。這些程式庫提供基本函數，可自動化任務，例如以密碼編譯方式簽署您的請求、重試請求，以及處理錯誤回應。如需詳細資訊，請參閱 [要建置的工具 AWS](#)。

AWS 管理主控台

AWS 管理主控台 提供用於管理 HealthLake 及其相關聯資源的 Web 型使用者介面。如果您已註冊 AWS 帳戶，您可以登入 [HealthLake 主控台](#)。

HIPAA 資格和資料安全

此為 HIPAA 合格服務。如需 1996 年 AWS 美國健康保險流通與責任法案 (HIPAA) 以及使用 AWS 服務來處理、存放和傳輸受保護醫療資訊 (PHI) 的詳細資訊，請參閱 [HIPAA 概觀](#)。

包含 PHI 和個人身分識別資訊 (PII) 的 HealthLake 連線必須加密。根據預設，所有 HealthLake 連線都會透過 TLS 使用 HTTPS。HealthLake 會存放加密的客戶內容，並根據 [AWS 共同責任模型](#) 運作。

定價

如需 HealthLake 定價資訊，請參閱 [AWS HealthLake 定價](#)。若要預估成本，請使用 [HealthLake 定價計算器](#)。

入門 AWS HealthLake

若要開始使用 AWS HealthLake，請設定 AWS 帳戶並建立 AWS Identity and Access Management 使用者。若要使用 [AWS CLI](#) 或 [AWS SDKs](#)，您必須安裝和設定它們。

Note

本指南的 [參考資料](#) 章節為 FHIR、FHIR R4 和 上的 SMART 提供支援內容 AWS HealthLake。例如，您可以找到有關 FHIR 組態的 SMART、支援的 FHIR 設定檔驗證和 HealthLake 端點的資訊。

了解 HealthLake 概念和設定後，提供包含程式碼範例的簡短教學課程，協助您開始使用。

主題

- [AWS HealthLake 概念](#)
- [設定 AWS HealthLake](#)
- [AWS HealthLake 教學課程](#)

AWS HealthLake 概念

下列術語和概念是您了解和使用的核心 AWS HealthLake。

概念

- [資料存放區授權策略](#)
- [整合式 NLP](#)
- [整合式分析](#)

資料存放區授權策略

HealthLake 資料存放區是 FHIR R4 運作狀態資料的儲存庫，位於單一 AWS 區域。HealthLake 支援下列資料存放區授權策略。

- SigV4 授權 — HealthLake 使用 [AWS Signature 第 4 版 \(SigV4\)](#) 授權授權 FHIR API 呼叫。

- FHIR 授權上的 SMART — HealthLake 在 FHIR 授權上使用 [可替代醫療應用程式和可重複使用技術 \(SMART\) 授權 FHIR API 呼叫](#)。

如需詳細資訊，請參閱[建立 HealthLake 資料存放區](#)。

整合式 NLP

AWS HealthLake 與 HIPAA 合格自然語言處理 (NLP) 程式庫整合，從非結構化醫療文字中擷取有意義的運作狀態資料。NLP 程式庫可識別醫療實體，例如條件、藥物、用量、測試、處理和程序。它們可識別實體之間的關係，並將其連結至醫療體學程式庫，例如 ICD-10-CM 和 RxNorm。如需詳細資訊，請參閱[HealthLake 的整合式自然語言處理 \(NLP\)](#)。

整合式分析

AWS HealthLake 超越 FHIR search 和 bundle APIs，提供整合式分析，用於查詢和分析大量運作狀態資料。在匯入期間，HealthLake 會自動產生 SQL 索引和查詢的資料表。這可讓您從複雜的醫療保健資料中取得可行的洞見，而不需要大量的資料工程工作。如需詳細資訊，請參閱[使用 Amazon Athena 查詢 HealthLake 資料](#)及[AWS HealthLake 範例專案](#)。

設定 AWS HealthLake

在本章中，您可以使用 AWS 管理主控台 來設定開始使用 AWS HealthLake 和建立資料存放區所需的許可。若要設定建立資料存放區的許可，您可以建立屬於資料湖管理員和 HealthLake 管理員的 IAM 使用者或角色。您會讓此使用者成為 AWS Lake Formation 中的資料湖管理員。資料湖管理員授予 Lake Formation 使用 Amazon Athena 查詢資料存放區所需的資源存取權。建立 HealthLake 資料存放區之後，您可以設定匯入和匯出檔案的許可。

主題

- [註冊 AWS 帳戶](#)
- [設定 IAM 使用者或角色以使用 HealthLake \(IAM 管理員\)](#)
- [在 Lake Formation 中將使用者或角色新增為 Data Lake 管理員 \(IAM 管理員\)](#)
- [建立 S3 儲存貯體](#)
- [建立資料存放區](#)
- [設定匯入任務的許可](#)
- [設定匯出任務的許可](#)
- [安裝 AWS CLI](#)

註冊 AWS 帳戶

若要開始使用 AWS，您需要 AWS 帳戶。如需建立的相關資訊 AWS 帳戶，請參閱《AWS 帳戶管理參考指南》中的 [入門 AWS 帳戶](#)。

設定 IAM 使用者或角色以使用 HealthLake (IAM 管理員)

人物：IAM 管理員

可建立 IAM 使用者和角色，並可新增資料湖管理員的使用者。

本主題中的這些步驟必須由 IAM 管理員執行。

若要將 HealthLake 資料存放區連線至 Athena，您需要建立 IAM 使用者或角色，其為資料湖管理員和 HealthLake 管理員。此新使用者或角色會透過 AWS Lake Formation 授予資料存放區中資源的存取權，並將 AmazonHealthLakeFullAccess AWS 受管政策新增至其使用者或角色。

Important

作為資料湖管理員的 IAM 使用者或角色無法建立新的資料湖管理員。若要新增其他資料湖管理員，您必須使用已授予 AdministratorAccess 存取權的 IAM 使用者或角色。

建立管理員

1. 將 **AmazonHealthlakeFullAccess** IAM AWS 受管政策新增至組織中的使用者或角色。

如果您不熟悉建立 IAM 使用者，請參閱《[IAM 使用者指南](#)》中的 [建立 IAM 使用者](#) 和 [AWS IAM 政策概觀](#)。

2. 授予 IAM 使用者或角色對 AWS Lake Formation 的存取權。

- 將下列 IAM AWS 受管政策新增至組織中的使用者或角色：**AWSLakeFormationDataAdmin**

Note

此 AWSLakeFormationDataAdmin 政策會授予所有 AWS Lake Formation 資源的存取權。建議您一律使用完成任務所需的最低許可。如需詳細資訊，請參《[IAM 使用者指南](#)》中的 IAM 最佳實務。

- 將下列內嵌政策新增至使用者或角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[內嵌政策](#)。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket/*",
        "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ram:GetResourceShareInvitations",
        "ram:AcceptResourceShareInvitation",
        "glue:CreateDatabase",
        "glue>DeleteDatabase"
      ],
      "Resource": "*"
    }
  ]
}
```

如需AWSLakeFormationDataAdmin政策的詳細資訊，請參閱 [Lake Formation 開發人員指南](#)中的 [Lake Formation 角色和 IAM 許可參考](#)。AWS

在 Lake Formation 中將使用者或角色新增為 Data Lake 管理員 (IAM 管理員)

Note

如果您要整合，則需要此步驟[SQL 索引和查詢](#)。

接著，IAM 管理員必須將上一個步驟中建立的使用者或角色新增為 Lake Formation 中的資料湖管理員。

將 IAM 使用者或角色新增為資料湖管理員

1. 開啟 AWS Lake Formation 主控台：<https://console.aws.amazon.com/lakeformation/>

Note

如果這是您第一次造訪 Lake Formation，會出現歡迎使用 Lake Formation 對話方塊，要求您定義 Lake Formation 管理員。

2. 將新使用者或角色指派為 AWS Lake Formation 資料湖管理員。

- 選項 1：如果您收到歡迎使用 Lake Formation 對話方塊。
 1. 選擇新增其他 AWS 使用者或角色。
 2. 選擇向下箭頭 (▼)。
 3. 選擇您想要同時成為 Lake Formation 管理員的 HealthLake 管理員。
 4. 選擇開始使用。
- 選項 2：使用導覽窗格 (≡)。
 1. 選擇導覽窗格 (≡)。
 2. 在許可下，選擇管理角色和任務。
 3. 在資料湖管理員區段中，選取選擇管理員。
 4. 在管理資料湖管理員對話方塊中，選擇向下箭頭 (▼)。
 5. 接著，選取或搜尋您也想要成為 Lake Formation 管理員的 HealthLake 管理員使用者或角色。

6. 選擇儲存。
3. 變更要由 Lake Formation 管理的預設安全設定。HealthLake 資料存放區資源需要由 Lake Formation 而非 IAM 管理。若要更新，請參閱 AWS Lake Formation 開發人員指南中的[變更預設許可模型](#)。

建立 S3 儲存貯體

若要將 FHIR R4 資料匯入 AWS HealthLake，建議使用兩個 Amazon S3 儲存貯體。Amazon S3 輸入儲存貯體會保留要匯入的 FHIR 資料，且 HealthLake 會從此儲存貯體讀取資料。Amazon S3 輸出儲存貯體會將匯入任務的處理結果和 HealthLake 寫入（日誌）儲存到此儲存貯體。

Note

由於 AWS Identity and Access Management (IAM) 政策，您的 Amazon S3 儲存貯體名稱必須是唯一的。如需詳細資訊，請參閱《Amazon Simple Storage Service 使用者指南》中的[儲存貯體命名規則](#)。

基於本指南的目的，我們會在本文稍後設定[匯入許可](#)時指定下列 Amazon S3 輸入和輸出儲存貯體。

- 輸入儲存貯體：`arn:aws:s3:::amzn-s3-demo-source-bucket`
- 輸出儲存貯體：`arn:aws:s3:::amzn-s3-demo-logging-bucket`

如需詳細資訊，請參閱《Amazon S3 使用者指南》中的[建立儲存貯體](#)。

建立資料存放區

HealthLake 資料存放區是位於單一 AWS 區域內的 FHIR R4 資料的儲存庫。AWS 帳戶可以有零或多個資料存放區。HealthLake 支援兩種資料存放區[授權策略](#)。

Important (重要)

建立 HealthLake 資料存放區之前，請檢閱 AWS 組織中可能限制建立或管理 HealthLake 資料的[服務控制政策 \(SCPs\)](#)。即使您的 IAM 許可設定正確，SCPs 也無法成功建立 HealthLake 資料存放區。

當您建立 HealthLake 資料存放區時，`datastoreID`會產生。在本文稍後設定[匯入許可](#)`datastoreID`時，您必須使用。

若要建立 HealthLake 資料存放區，請參閱 [建立 HealthLake 資料存放區](#)。

設定匯入任務的許可

將檔案匯入資料存放區之前，您必須授予 HealthLake 許可，才能存取 Amazon S3 中的輸入和輸出儲存貯體。若要授予 HealthLake 存取權，您可以為 HealthLake 建立 IAM 服務角色、將信任政策新增至角色以授予 HealthLake 擔任角色許可，並將許可政策連接至角色，以授予其存取您的 Amazon S3 儲存貯體。

當您建立匯入任務時，請為指定此角色的 Amazon Resource Name (ARN) `DataAccessRoleArn`。如需 IAM 角色和信任政策的詳細資訊，請參閱 [IAM 角色](#)。

設定許可之後，您就可以使用匯入任務將檔案匯入資料存放區。如需詳細資訊，請參閱 [啟動 FHIR 匯入任務](#)。

設定匯入許可

1. 如果尚未建立輸出日誌檔案的目的地 Amazon S3 儲存貯體。Amazon S3 儲存貯體必須與服務位於相同的 AWS 區域，且所有選項都必須開啟封鎖公開存取。若要進一步了解，請參閱 [使用 Amazon S3 封鎖公開存取](#)。Amazon 擁有或客戶擁有的 KMS 金鑰也必須用於加密。若要進一步了解如何使用 KMS 金鑰，請參閱 [Amazon Key Management Service](#)。
2. 為 HealthLake 建立資料存取服務角色，並授予 HealthLake 服務使用下列信任政策擔任該角色的許可。HealthLake 使用此值來寫入輸出 Amazon S3 儲存貯體。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "healthlake.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountID"
        }
      }
    }
  ]
}
```

```

        },
        "ArnEquals": {
            "aws:SourceArn": "arn:aws:healthlake:us-
west-2:111122223333:datastore/fhir/datastoreID"
        }
    }
}
]
}

```

3. 將許可政策新增至資料存取角色，以允許其存取 Amazon S3 儲存貯體。amzn-s3-demo-bucket 將取代之為您儲存貯體的名稱。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetEncryptionConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey*"
      ],
      "Resource": [

```

```
        "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-  
        f4c43ef46e83"  
      ],  
      "Effect": "Allow"  
    }  
  }  
}
```

設定匯出任務的許可

從資料存放區匯出檔案之前，您必須授予 HealthLake 在 Amazon S3 中存取輸出儲存貯體的許可。若要授予 HealthLake 存取權，您可以為 HealthLake 建立 IAM 服務角色、將信任政策新增至角色以授予 HealthLake 擔任角色許可，並將許可政策連接至角色，以授予其存取您的 Amazon S3 儲存貯體。

如果您已為 HealthLake 建立角色，您可以重複使用該角色，並授予此主題中列出的匯出 Amazon S3 儲存貯體的其他許可。若要進一步了解 IAM 角色和信任政策，請參閱 [IAM 政策和許可](#)。

Important (重要)

HealthLake 同時支援[原生 SDK 匯出請求](#)和 [FHIR R4 \\$export](#) 操作。根據您決定使用的匯出 API，必須提供個別的 IAM 動作。這可讓您分別處理 allow 和 deny 許可。如果您想要同時限制 HealthLake SDK 和 FHIR REST API 匯出，您必須將拒絕許可套用至個別的 IAM 動作。如果您提供使用者 HealthLake 的完整存取權，則不需要 IAM 使用者許可變更。

使用 AWS CLI 和 AWS SDKs：

下列原生 HealthLake 動作可用於使用 AWS CLI 和 AWS SDKs 從資料存放區匯出資料：

- StartFHIRExportJob
- DescribeFHIRExportJob
- ListFHIRExportJobs

使用 FHIR APIs：

下列 IAM 動作可用於從 HealthLake 資料存放區匯出資料，以及使用 FHIR \$export 操作取消（刪除）匯出任務：

POST:

- StartFHIRExportJobWithPost

GET:

- StartFHIRExportJobWithGet
- DescribeFHIRExportJobWithGet
- GetExportedFile

DELETE:

- CancelFHIRExportJobWithDelete

設定許可的使用者或角色必須具有建立角色、建立政策以及將政策連接至角色的許可。下列 IAM 政策會授予這些許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": "iam:PassRole",
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "healthlake.amazonaws.com"
        }
      }
    }
  ]
}
```

```
}
```

設定匯出許可

1. 如果尚未，請為您要從資料存放區匯出的資料建立目的地 Amazon S3 儲存貯體。Amazon S3 儲存貯體必須與服務位於相同的 AWS 區域，且所有選項都必須開啟封鎖公開存取。若要進一步了解，請參閱[使用 Amazon S3 封鎖公開存取](#)。Amazon 擁有或客戶擁有的 KMS 金鑰也必須用於加密。若要進一步了解如何使用 KMS 金鑰，請參閱[Amazon Key Management Service](#)。
2. 如果您尚未建立 HealthLake 的資料存取服務角色，並授予 HealthLake 服務使用下列信任政策擔任該角色的許可。HealthLake 使用此值來寫入輸出 Amazon S3 儲存貯體。如果您已在 中建立一個[設定匯入任務的許可](#)，則可以重複使用它，並在下一個步驟中授予 Amazon S3 儲存貯體的許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "healthlake.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "accountID"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:healthlake:us-
west-2:111122223333:datastore/fhir/data store ID"
        }
      }
    }
  ]
}
```

3. 將許可政策新增至資料存取角色，以允許它存取您的輸出 Amazon S3 儲存貯體。amzn-s3-demo-bucket 將取代為您儲存貯體的名稱。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetEncryptionConfiguration"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-source-bucket"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-logging-bucket/*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:DescribeKey",
        "kms:GenerateDataKey*"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-b56c-4216-a250-f4c43ef46e83"
      ],
      "Effect": "Allow"
    }
  ]
}
```

安裝 AWS CLI

AWS CLI 需要描述和列出 HealthLake 匯入和匯出任務屬性。您也可以使用 HealthLake SDKs 請求此資訊。

若要設定 AWS CLI

1. 下載和設定 AWS CLI。如需說明，請參閱《AWS Command Line Interface 使用者指南》中的下列主題。
 - [安裝或更新最新版本的 AWS CLI](#)
 - [開始使用 AWS CLI](#)
2. 在 AWS CLI config 檔案中，新增管理員的具名設定檔。您在執行 AWS CLI 命令時使用此設定檔。在最低權限的安全原則下，我們建議您建立具有所執行任務特定權限的個別 IAM 角色。如需具名設定檔的詳細資訊，請參閱 AWS Command Line Interface 《使用者指南》中的[組態和登入資料檔案設定](#)。

```
[default]
aws_access_key_id = default access key ID
aws_secret_access_key = default secret access key
region = region
```

3. 使用下列 help 命令驗證設定。

```
aws healthlake help
```

如果 AWS CLI 設定正確，您會看到的 AWS HealthLake 簡短描述和可用命令的清單。

AWS HealthLake 教學課程

目標

在本教學課程中，您將使用原生 HealthLake 動作將 FHIR R4 資料匯入 HealthLake 資料存放區。接著，您將使用 FHIR RESTful APIs 管理（建立、讀取、更新、刪除）FHIR 資源。若要結束教學課程，您將使用原生 HealthLake 動作匯出 FHIR 資料。

先決條件

中列出的所有程序 [設定](#) 都需要完成本教學課程。

教學步驟

1. [啟動 FHIR 匯入任務](#)
2. [取得 FHIR 匯入任務屬性](#)
3. [建立 FHIR 資源](#)
4. [讀取 FHIR 資源](#)
5. [更新 FHIR 資源](#)
6. [刪除 FHIR 資源](#)
7. [匯出 FHIR 資料](#)
8. [刪除資料存放區](#)

使用 管理資料存放區 AWS HealthLake

使用 AWS HealthLake，您可以建立和管理 FHIR R4 資源的資料存放區。當您建立 HealthLake 資料存放區時，可透過 RESTful API [端點](#)提供 FHIR 資料儲存庫。建立時，您可以選擇將（預先載入）Synthea 開放原始碼 FHIR R4 運作狀態資料匯入資料存放區。如需詳細資訊，請參閱[預先載入的資料類型](#)。

Important (重要)

HealthLake 支援兩種類型的 FHIR 資料存放區授權策略：AWS SigV4 或 FHIR 上的 SMART。您必須先選擇其中一個授權策略，才能建立 HealthLake FHIR 資料存放區。如需詳細資訊，請參閱[資料存放區授權策略](#)。

若要尋找作用中 HealthLake 資料存放區的 FHIR 相關功能（行為），請擷取其[功能陳述](#)式。

下列主題說明如何使用 HealthLake 雲端原生動作來使用、AWS SDKs 和 建立、描述、列出、更新 AWS CLI、標記和刪除 FHIR 資料存放區 AWS 管理主控台。

主題

- [建立 HealthLake 資料存放區](#)
- [取得 HealthLake 資料存放區屬性](#)
- [列出 HealthLake 資料存放區](#)
- [更新 HealthLake 資料存放區](#)
- [標記 HealthLake 資料存放區](#)
- [刪除 HealthLake 資料存放區](#)

建立 HealthLake 資料存放區

使用 CreateFHIRDatastore 建立符合 FHIR R4 規格 AWS HealthLake 的資料存放區。HealthLake 資料存放區用於匯入、管理、搜尋和匯出 FHIR 資料。建立時，您可以選擇將（預先載入）Synthea 開放原始碼 FHIR R4 運作狀態資料匯入資料存放區。如需詳細資訊，請參閱[預先載入的資料類型](#)。

i Important (重要)

HealthLake 支援兩種類型的 FHIR 資料存放區授權策略：AWS SigV4 或 FHIR 上的 SMART。在建立 HealthLake FHIR 資料存放區之前，您必須選擇其中一個授權策略。如需詳細資訊，請參閱[資料存放區授權策略](#)。

當您建立 HealthLake 資料存放區時，可透過 RESTful API [端點](#)提供 FHIR 資料儲存庫。建立 HealthLake 資料存放區之後，您可以請求其[功能陳述](#)式來尋找所有相關的 FHIR 相關功能（行為）。

建立資料存放區之後，您可以變更其名稱、預設 FHIR 驗證設定檔、NLP 組態、分析組態和身分提供者組態。加密組態無法變更。如需詳細資訊，請參閱[更新資料存放區](#)。

下列功能表提供 AWS CLI 和 AWS SDKs 的範例，以及 的程序 AWS 管理主控台。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [CreateFHIRDatastore](#)。

建立 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

範例 1：建立 SigV4-enabled HealthLake 資料存放區

下列 create-fhir-datastore 範例示範如何在 AWS HealthLake 中建立新的資料存放區。

```
aws healthlake create-fhir-datastore \  
  --datastore-type-version R4 \  
  --datastore-name "FhirTestDatastore"
```

輸出：

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Data store ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Data store ID)",  
  "DatastoreStatus": "CREATING",
```

```
"DatastoreId": "(Data store ID)"
}
```

範例 2：在啟用 FHIR 的 HealthLake 資料存放區上建立 SMART

下列 `create-fhir-datastore` 範例示範如何在啟用 FHIR 的資料存放區 in AWS HealthLake 上建立新的 SMART。

```
aws healthlake create-fhir-datastore \
  --datastore-name "your-data-store-name" \
  --datastore-type-version R4 \
  --preload-data-config PreloadDataType="SYNTHEA" \
  --sse-configuration '{ "KmsEncryptionConfig": { "CmkType":
"CUSTOMER_MANAGED_KMS_KEY", "KmsKeyId": "arn:aws:kms:us-east-1:your-account-
id:key/your-key-id" } }' \
  --identity-provider-configuration file://
identity_provider_configuration.json
```

`identity_provider_configuration.json` 的內容：

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR_V1",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
  "Metadata": "{\"issuer\": \"https://ehr.example.com\", \"jwks_uri\":
\"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint
\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://
ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\":
[\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential
\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/
register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"],
\"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://
ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://
ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],
\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\"]}"
}
```

輸出：

```
{
```

```
"DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
(Data store ID)/r4/",
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/
(Data store ID)",
  "DatastoreStatus": "CREATING",
  "DatastoreId": "(Data store ID)"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [CreateFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def create_fhir_datastore(
    self,
    datastore_name: str,
    sse_configuration: dict[str, any] = None,
    identity_provider_configuration: dict[str, any] = None,
) -> dict[str, str]:
    """
    Creates a new HealthLake data store.
    When creating a SMART on FHIR data store, the following parameters are
    required:
    - sse_configuration: The server-side encryption configuration for a SMART
    on FHIR-enabled data store.
    - identity_provider_configuration: The identity provider configuration
    for a SMART on FHIR-enabled data store.
```

```
    :param datastore_name: The name of the data store.
    :param sse_configuration: The server-side encryption configuration for a
SMART on FHIR-enabled data store.
    :param identity_provider_configuration: The identity provider
configuration for a SMART on FHIR-enabled data store.
    :return: A dictionary containing the data store information.
    """
    try:
        parameters = {"DatastoreName": datastore_name,
"DatastoreTypeVersion": "R4"}
        if (
            sse_configuration is not None
            and identity_provider_configuration is not None
        ):
            # Creating a SMART on FHIR-enabled data store
            parameters["SseConfiguration"] = sse_configuration
            parameters[
                "IdentityProviderConfiguration"
            ] = identity_provider_configuration

        response =
self.health_lake_client.create_fhir_datastore(**parameters)
        return response
    except ClientError as err:
        logger.exception(
            "Couldn't create data store %s. Here's why %s",
            datastore_name,
            err.response["Error"]["Message"],
        )
        raise
```

下列程式碼示範啟用 FHIR 的 HealthLake 資料存放區上的 SMART 參數範例。

```
sse_configuration = {
    "KmsEncryptionConfig": {"CmkType": "AWS_OWNED_KMS_KEY"}
}
# TODO: Update the metadata to match your environment.
metadata = {
    "issuer": "https://ehr.example.com",
    "jwks_uri": "https://ehr.example.com/.well-known/jwks.json",
```

```

        "authorization_endpoint": "https://ehr.example.com/auth/
authorize",
        "token_endpoint": "https://ehr.token.com/auth/token",
        "token_endpoint_auth_methods_supported": [
            "client_secret_basic",
            "foo",
        ],
        "grant_types_supported": ["client_credential", "foo"],
        "registration_endpoint": "https://ehr.example.com/auth/register",
        "scopes_supported": ["openId", "profile", "launch"],
        "response_types_supported": ["code"],
        "management_endpoint": "https://ehr.example.com/user/manage",
        "introspection_endpoint": "https://ehr.example.com/user/
introspect",
        "revocation_endpoint": "https://ehr.example.com/user/revoke",
        "code_challenge_methods_supported": ["S256"],
        "capabilities": [
            "launch-ehr",
            "sso-openid-connect",
            "client-public",
        ],
    }
    # TODO: Update the IdpLambdaArn.
    identity_provider_configuration = {
        "AuthorizationStrategy": "SMART_ON_FHIR_V1",
        "FineGrainedAuthorizationEnabled": True,
        "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-
id:function:your-lambda-name",
        "Metadata": json.dumps(metadata),
    }
    data_store = self.create_fhir_datastore(
        datastore_name, sse_configuration,
        identity_provider_configuration
    )

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [CreateFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_name = 'MyHealthLakeDataStore'  
  oo_result = lo_hll->createfhirdatastore(  
    iv_datastorename = iv_datastore_name  
    iv_datastoretypeversion = 'R4'  
  ).  
  MESSAGE 'Data store created successfully.' TYPE 'I'.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
  CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).  
    lv_error = |Internal server error: { lo_internal_ex->av_err_code }-  
{ lo_internal_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
  CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).  
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-  
{ lo_throttling_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_throttling_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [CreateFHIRDatastore](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

注意

下列程序會使用 [AWS SigV4](#) 授權建立 HealthLake 資料存放區。HealthLake 主控台不支援在 FHIR 資料存放區上建立 SMART。

使用 AWS SigV4 授權建立 HealthLake 資料存放區

1. 登入 HealthLake 主控台上的 [建立資料存放區](#) 頁面。
2. 選擇建立資料存放區。
3. 在資料存放區設定區段中，針對資料存放區名稱指定名稱。
4. （選用）在資料存放區設定區段中，針對預先載入範例資料，選取核取方塊以預先載入 Synthea 資料。Synthea 資料是開放原始碼範例資料集。如需詳細資訊，請參閱 [HealthLake 的 Synthea 預先載入資料類型](#)。
5. 在資料存放區加密區段中，選擇使用 AWS 擁有的金鑰（預設）或選擇不同的 AWS KMS 金鑰（進階）。
6. 在標籤 - 選用區段中，您可以將標籤新增至資料存放區。若要進一步了解如何標記資料存放區，請參閱 [標記 HealthLake 資料存放區](#)。
7. 選擇建立資料存放區。

資料存放區的狀態可在資料存放區頁面上取得。

取得 HealthLake 資料存放區屬性

使用 `DescribeFHIRDatastore` 取得 AWS HealthLake 資料存放區的屬性。下列功能表提供 AWS 管理主控台 和 SDKs 的 AWS CLI 和 AWS 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [DescribeFHIRDatastore](#)。

取得 HealthLake 資料存放區的屬性

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

描述 FHIR 資料存放區

下列 `describe-fhir-datastore` 範例示範如何尋找資料存放區 in AWS HealthLake 的屬性。

```
aws healthlake describe-fhir-datastore \  
  --datastore-id "1f2f459836ac6c513ce899f9e4f66a59"
```

輸出：

```
{  
  "DatastoreProperties": {  
    "PreloadDataConfig": {  
      "PreloadDataType": "SYNTHEA"  
    },  
    "SseConfiguration": {  
      "KmsEncryptionConfig": {  
        "CmkType": "CUSTOMER_MANAGED_KMS_KEY",  
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
      }  
    },  
    "DatastoreName": "Demo",  
    "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/  
<Data store ID>",  
  }  
}
```

```

    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/
datastore/<Data store ID>/r4/",
    "DatastoreStatus": "ACTIVE",
    "DatastoreTypeVersion": "R4",
    "CreatedAt": 1603761064.881,
    "DatastoreId": "<Data store ID>",
    "IdentityProviderConfiguration": {
        "AuthorizationStrategy": "AWS_AUTH",
        "FineGrainedAuthorizationEnabled": false
    }
}
}
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```

@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def describe_fhir_datastore(self, datastore_id: str) -> dict[str, any]:
    """
    Describes a HealthLake data store.
    :param datastore_id: The data store ID.
    :return: The data store description.
    """
    try:
        response = self.health_lake_client.describe_fhir_datastore(
            DatastoreId=datastore_id
        )
    
```

```
        return response["DatastoreProperties"]
    except ClientError as err:
        logger.exception(
            "Couldn't describe data store with ID %s. Here's why %s",
            datastore_id,
            err.response["Error"]["Message"],
        )
        raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    oo_result = lo_h11->describefhirdatastore(
        iv_datastoreid = iv_datastore_id
    ).
    DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).
    IF lo_datastore_properties IS BOUND.
        DATA(lv_datastore_name) = lo_datastore_properties-
>get_datastorename( ).
        DATA(lv_datastore_status) = lo_datastore_properties-
>get_datastorestatus( ).
```

```
        MESSAGE 'Data store described successfully.' TYPE 'I'.
    ENDIF.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
        lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRDatastore](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的 [資料存放區](#) 頁面。
2. 選擇資料存放區。

資料存放區詳細資訊頁面隨即開啟，且所有 HealthLake 資料存放區屬性都可用。

列出 HealthLake 資料存放區

使用 `ListFHIRDatastores` 列出使用者帳戶中的所有 HealthLake 資料存放區，無論資料存放區狀態為何。下列功能表提供 AWS CLI AWS SDKs AWS 管理主控台的 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [ListFHIRDatastores](#)。

列出所有 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

列出 FHIR 資料存放區

下列 `list-fhir-datastores` 範例顯示如何使用 命令，以及使用者如何根據 AWS HealthLake 中的資料存放區狀態篩選結果。

```
aws healthlake list-fhir-datastores \  
  --filter DatastoreStatus=ACTIVE
```

輸出：

```
{  
  "DatastorePropertiesList": [  
    {  
      "PreloadDataConfig": {  
        "PreloadDataType": "SYNTHEA"  
      },  
      "SseConfiguration": {  
        "KmsEncryptionConfig": {  
          "CmkType": "CUSTOMER_MANAGED_KMS_KEY",  
          "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
        }  
      },  
      "DatastoreName": "Demo",  
      "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/  
<Data store ID>",  
      "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/  
datastore/<Data store ID>/r4/",  
      "DatastoreStatus": "ACTIVE",  
      "DatastoreTypeVersion": "R4",  
      "CreatedAt": 1603761064.881,  
      "DatastoreId": "<Data store ID>",  
      "IdentityProviderConfiguration": {  
        "AuthorizationStrategy": "AWS_AUTH",  
        "FineGrainedAuthorizationEnabled": false  
      }  
    }  
  ]  
}
```

```
]
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRDatastores](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_fhir_datastores(self) -> list[dict[str, any]]:
    """
    Lists all HealthLake data stores.
    :return: A list of data store descriptions.
    """
    try:
        next_token = None
        datastores = []

        # Loop through paginated results.
        while True:
            parameters = {}
            if next_token is not None:
                parameters["NextToken"] = next_token
            response =
self.health_lake_client.list_fhir_datastores(**parameters)
            datastores.extend(response["DatastorePropertiesList"])
            if "NextToken" in response:
                next_token = response["NextToken"]
            else:
```

```

        break

    return datastores
except ClientError as err:
    logger.exception(
        "Couldn't list data stores. Here's why %s", err.response["Error"]
["Message"]
    )
    raise

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRDatastores](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
    oo_result = lo_hll->listfhirdatastores( ).
    DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
    DATA(lv_datastore_count) = lines( lt_datastores ).
    MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).

```

```
lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-  
{ lo_throttling_ex->av_err_msg }|.  
MESSAGE lv_error TYPE 'I'.  
RAISE EXCEPTION lo_throttling_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIRDatastores](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

- 登入 HealthLake 主控台上的 [資料存放區](#) 頁面。

所有 HealthLake 資料存放區都會列在資料存放區區段下。

更新 HealthLake 資料存放區

使用 UpdateFHIRDatastore 更新現有 AWS HealthLake 資料存放區的組態。您可以更新資料存放區名稱、自然語言處理 (NLP) 組態、分析組態、預設 FHIR 驗證設定檔，以及身分提供者組態。您在建立資料存放區時選擇的加密組態無法變更。

Note

更新身分提供者組態會將其完全取代，包括您要保留的每個欄位。您省略的任何欄位都會清除。

下列選單提供 AWS CLI 和 AWS SDKs 的範例。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [UpdateFHIRDatastore](#)。

i Important (重要)

NLP 和分析變更會透過非同步工作流程套用：資料存放區狀態會變更為 UPDATING，並在更新完成ACTIVE時傳回，否則會顯示 UPDATE_FAILED。資料存放區名稱、FHIR 驗證設定檔和身分提供者變更會立即生效，而且不會變更狀態。一次只能有一個資料存放區的更新正在進行中；當資料存放區執行時提交的第二個更新會傳回 ConflictException。使用 DescribeFHIRDatastore 追蹤更新的狀態。

更新 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

AWS CLI

範例 1：重新命名資料存放區

```
aws healthlake update-fhir-datastore \  
  --datastore-id "datastore-id" \  
  --datastore-name "RenamedFhirDatastore"
```

範例 2：啟用 NLP

```
aws healthlake update-fhir-datastore \  
  --datastore-id "datastore-id" \  
  --nlp-configuration '{ "Status": "ENABLED" }'
```

範例 3：暫停分析

```
aws healthlake update-fhir-datastore \  
  --datastore-id "datastore-id" \  
  --analytics-configuration '{ "Status": "PAUSED" }'
```

範例 4：更新預設 FHIR 驗證設定檔

```
aws healthlake update-fhir-datastore \  
  --datastore-id "datastore-id" \  
  --validation-configuration '{ "Status": "ENABLED" }'
```

```
--profile-configuration '{ "DefaultProfiles": ["us-core-3.1.1", "carin-
bb-2.0.0"] }'
```

回應會傳回完整 DatastoreProperties- 與傳回的形狀相同 DescribeFHIRDatastore。

```
{
  "DatastoreProperties": {
    "DatastoreId": "datastore-id",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:account-
id:datastore/datastore-id",
    "DatastoreName": "RenamedFhirDatastore",
    "DatastoreStatus": "UPDATING",
    "DatastoreTypeVersion": "R4",
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/
datastore/datastore-id/r4/",
    "NlpConfiguration": { "Status": "ENABLING" },
    "AnalyticsConfiguration": { "Status": "PAUSING" },
    "ProfileConfiguration": { "DefaultProfiles": [ "us-core-3.1.1", "carin-
bb-2.0.0" ] },
    "IdentityProviderConfiguration": {
      "AuthorizationStrategy": "SMART_ON_FHIR_V1",
      "FineGrainedAuthorizationEnabled": true
    }
  }
}
```

如需 API 詳細資訊，請參閱 AWS CLI 命令參考中的 [update-fhir-datastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```
def update_fhir_datastore(
    self,
    datastore_id: str,
    **kwargs,
) -> dict[str, any]:
    """
    Updates the configuration of an existing HealthLake data store.

    Pass any of DatastoreName, NlpConfiguration, AnalyticsConfiguration,
    ProfileConfiguration, or IdentityProviderConfiguration as keyword
    arguments. Omitted fields are left unchanged.
```

```
:param datastore_id: The ID of the data store to update.
:return: The response, including the full DatastoreProperties.
"""
try:
    return self.health_lake_client.update_fhir_datastore(
        DatastoreId=datastore_id, **kwargs
    )
except ClientError as err:
    logger.exception(
        "Couldn't update data store %s. Here's why: %s",
        datastore_id,
        err.response["Error"]["Message"],
    )
    raise
```

如需 API 詳細資訊，請參閱《適用於 AWS Python (Boto3) 的 SDK API 參考》中的 [UpdateFHIRDatastore](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

標記 HealthLake 資料存放區

您可以將中繼資料以標籤形式指派給 HealthLake 資料存放區。每個標籤都是由使用者定義的金鑰和值組成的標籤。標籤可協助您管理、識別、組織、搜尋和篩選資料存放區。

Important (重要)

請勿在標籤中存放受保護醫療資訊 (PHI)、個人身分識別資訊 (PII) 或其他機密或敏感資訊。標籤不適用於私有或敏感資料。

下列主題說明如何使用 AWS 管理主控台、AWS CLI 和 AWS SDKs 來使用 HealthLake 標記操作。如需詳細資訊，請參閱《AWS 一般參考指南》中的 [標記您的 AWS 資源](#)。

主題

- [標記 HealthLake 資料存放區](#)
- [列出 HealthLake 資料存放區的標籤](#)
- [取消標記 HealthLake 資料存放區](#)

標記 HealthLake 資料存放區

使用 `TagResource` 標記 HealthLake 資料存放區。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [TagResource](#)。

標記 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

將標籤新增至資料存放區

以下 `tag-resource` 範例示範如何將標籤新增至資料存放區。

```
aws healthlake tag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/  
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \  
  --tags '[{"Key": "key1", "Value": "value1"}]'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [TagResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
```

```
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def tag_resource(self, resource_arn: str, tags: list[dict[str, str]]) ->
None:
    """
    Tags a HealthLake resource.
    :param resource_arn: The resource ARN.
    :param tags: The tags to add to the resource.
    """
    try:
        self.health_lake_client.tag_resource(ResourceARN=resource_arn,
        Tags=tags)
    except ClientError as err:
        logger.exception(
            "Couldn't tag resource %s. Here's why %s",
            resource_arn,
            err.response["Error"]["Message"],
        )
        raise
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [TagResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP


適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/  
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
    lo_hll->tagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tags = it_tags  
    ).  
    MESSAGE 'Resource tagged successfully.' TYPE 'I'.  
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [TagResource](#)。

 可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的[資料存放區](#)頁面。
2. 選擇資料存放區。

資料存放區詳細資訊頁面隨即開啟。

3. 在標籤區段下，選擇管理標籤。

隨即開啟管理標籤頁面。

4. 選擇 Add new tag (新增標籤)。
5. 輸入金鑰和值 (選用)。
6. 選擇儲存。

列出 HealthLake 資料存放區的標籤

使用 `ListTagsForResource` 列出 HealthLake 資料存放區的標籤。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [ListTagsForResource](#)。

列出 HealthLake 資料存放區的標籤

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

列出資料存放區的標籤

下列 `list-tags-for-resource` 範例列出與指定資料存放區相關聯的標籤：

```
aws healthlake list-tags-for-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/  
  fhir/0725c83f4307f263e16fd56b6d8ebdbe"
```

輸出：

```
{
  "tags": {
    "key": "value",
    "key1": "value1"
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListTagsForResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_tags_for_resource(self, resource_arn: str) -> dict[str, str]:
    """
    Lists the tags for a HealthLake resource.
    :param resource_arn: The resource ARN.
    :return: The tags for the resource.
    """
    try:
        response = self.health_lake_client.list_tags_for_resource(
            ResourceARN=resource_arn
        )
        return response["Tags"]
    except ClientError as err:
        logger.exception(
            "Couldn't list tags for resource %s. Here's why %s",
            resource_arn,
            err.response["Error"]["Message"],
```

```
)
raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListTagsForResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    DATA(lo_result) = lo_hll->listtagsforresource(
        iv_resourcearn = iv_resource_arn
    ).
    ot_tags = lo_result->get_tags( ).
    DATA(lv_tag_count) = lines( ot_tags ).
    MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
```

```
MESSAGE lv_error TYPE 'I'.  
RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListTagsForResource](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的 [資料存放區](#) 頁面。
2. 選擇資料存放區。

資料存放區詳細資訊頁面隨即開啟。在標籤區段下，會列出所有資料存放區標籤。

取消標記 HealthLake 資料存放區

使用從 HealthLake 資料存放區 `UntagResource` 移除標籤。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [UntagResource](#)。

取消標記 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

從資料存放區移除標籤。

下列 `untag-resource` 範例示範如何從資料存放區移除標籤。

```
aws healthlake untag-resource \
```

```
--resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
b91723d65c6fdeb1d26543a49d2ed1fa" \
--tag-keys '["key1"]'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [UntagResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def untag_resource(self, resource_arn: str, tag_keys: list[str]) -> None:
    """
    Untags a HealthLake resource.
    :param resource_arn: The resource ARN.
    :param tag_keys: The tag keys to remove from the resource.
    """
    try:
        self.health_lake_client.untag_resource(
            ResourceARN=resource_arn, TagKeys=tag_keys
        )
    except ClientError as err:
        logger.exception(
            "Couldn't untag resource %s. Here's why %s",
            resource_arn,
            err.response["Error"]["Message"],
        )
        raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [UntagResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/  
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
    lo_hll->untagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tagkeys = it_tag_keys  
    ).  
    MESSAGE 'Resource untagged successfully.' TYPE 'I'.  
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [UntagResource](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的 [資料存放區](#) 頁面。
2. 選擇資料存放區。

資料存放區詳細資訊頁面隨即開啟。

3. 在標籤區段下，選擇管理標籤。

隨即開啟管理標籤頁面。

4. 選擇您要移除之標籤旁的移除。
5. 選擇儲存。

刪除 HealthLake 資料存放區

使用 `DeleteFHIRDatastore` 刪除 HealthLake 資料存放區。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [DeleteFHIRDatastore](#)。

刪除 HealthLake 資料存放區

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

刪除 FHIR 資料存放區

下列delete-fhir-datastore範例示範如何刪除資料存放區及其在 AWS HealthLake 中的所有內容。

```
aws healthlake delete-fhir-datastore \  
  --datastore-id (Data store ID)
```

輸出：

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Data store ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Data store ID)",  
  "DatastoreStatus": "DELETING",  
  "DatastoreId": "(Data store ID)"  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DeleteFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.  
    """  
    health_lake_client = boto3.client("healthlake")  
    return cls(health_lake_client)  
  
def delete_fhir_datastore(self, datastore_id: str) -> None:  
    """  
    Deletes a HealthLake data store.  
    :param datastore_id: The data store ID.  
    """
```

```
try:

self.health_lake_client.delete_fhir_datastore(DatastoreId=datastore_id)
except ClientError as err:
    logger.exception(
        "Couldn't delete data store with ID %s. Here's why %s",
        datastore_id,
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DeleteFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  oo_result = lo_hll->deletefhirdatastore(
    iv_datastoreid = iv_datastore_id
  ).
  MESSAGE 'Data store deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
  DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
  { lo_access_ex->av_err_msg }|.

```

```
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_access_ex.
CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
{ lo_conflict_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_conflict_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DeleteFHIRDatastore](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的 [資料存放區](#) 頁面。
2. 選擇資料存放區。

資料存放區詳細資訊頁面隨即開啟。

3. 選擇 刪除。

刪除資料存放區頁面隨即開啟。

4. 若要確認刪除資料存放區，請在文字輸入欄位中輸入資料存放區名稱。
5. 選擇刪除。

在中管理 FHIR 訂閱 AWS HealthLake

AWS HealthLake 支援 FHIR 訂閱，可讓您在特定醫療保健資料發生變更時接收即時通知。此功能實作 FHIR R5 Backport 主題型訂閱模型，相較於傳統 FHIR R4 訂閱模型，可提供更佳的可擴展性和彈性。

透過 FHIR 訂閱，您可以建置事件驅動的醫療保健應用程式，以立即回應臨床資料的變化，從而及時介入、自動化工作流程和增強護理協調。

主題

- [FHIR 訂閱的運作方式](#)
- [關鍵元件](#)
- [最佳實務](#)
- [使用的 FHIR 訂閱生命週期 AWS HealthLake](#)
- [使用 建立 FHIR 訂閱 AWS HealthLake](#)
- [使用 搜尋 FHIR 訂閱 AWS HealthLake](#)
- [使用 篩選通知 AWS HealthLake](#)

FHIR 訂閱的運作方式

HealthLake 中的 FHIR 訂閱在主題型模型上運作，其中：

1. 建立主題以定義事件：建立訂閱主題以指定可觸發通知的事件
2. 您訂閱：使用特定篩選條件建立這些主題的訂閱
3. HealthLake 監控：服務會持續監控符合您條件的事件
4. 傳送通知：CWhen，HealthLake 會透過您選擇的管道傳送通知

關鍵元件

FHIR 訂閱包含下列元件。

訂閱主題

訂閱主題是通知系統的基礎，並定義：

- 觸發事件：哪些變更觸發通知（例如：資源建立、更新、刪除）
- 可用的篩選條件：訂閱者可使用哪些篩選選項
- 通知內容：通知中包含哪些資料

下表列出常見的主題類型。

事件類型	Description	常用案例
資源建立	建立資源時觸發	新病患註冊，新觀察記錄
資源更新	修改資源時觸發	狀態變更、臨床更新
資源刪除	刪除資源時觸發	稽核與合規追蹤

Subscriptions

訂閱是您接收訂閱主題所定義特定事件通知的請求。每個訂閱包括：

- 主題參考：指定您要訂閱的訂閱主題
- 篩選條件：選取哪些事件產生通知的條件
- 頻道組態：應傳送通知的位置和方式
- 承載偏好設定：通知中應包含詳細程度

通知管道

HealthLake 支援下列通知管道：

頻道類型	使用案例
EventBridge	企業整合、無伺服器工作流程、跨AWS 服務協同運作

頻道類型	使用案例	
REST Hook	直接端點通知、第三方系統整合	

通知承載

根據您的需求選擇適當的承載類型：

承載類型	Description	安全考量
僅限 ID	僅包含資源識別符	最低 PHI 暴露
完整資源	包含大小上限為 256 KB 的完整資源內容。如果大小大於 256KB，則會回到僅限 ID	包含 PHI；驗證安全處理

最佳實務

效能最佳化

- 使用聚焦篩選條件：縮小您的條件以僅接收基本通知
- 選擇適當的承載類型：盡可能使用僅限 ID 的承載以獲得更好的效能
- 實作有效率的接收者：確保通知接收者快速處理訊息

安全考量

- 安全端點：為 REST Hook 端點實作適當的身分驗證
- PHI 保護：請謹慎處理完整資源承載，因為其中包含 PHI
- 存取控制：僅限授權使用者建立訂閱

卓越營運

- 設定適當的結束日期：使用暫時訂閱的結束日期
- 監控訂閱狀態：定期檢查訂閱的狀態

- 實作錯誤處理：設計您的應用程式來處理通知交付失敗

使用的 FHIR 訂閱生命週期 AWS HealthLake

請依照下列步驟了解 FHIR 訂閱生命週期：

1. 建立 SubscriptionTopic

- 建立SubscriptionTopic狀態為的 "unknown"

2. 建立 Subscription

- 建立Subscription狀態為的 "requested"
- HealthLake 驗證Subscription組態
- Subscription 必須參考已存在的主題（主題必須處於狀態 unknown、draft、)active。

3. 啟用

- 如果有效，HealthLake 會將的狀態更新Subscription為 "active"
- 建立時Subscription，如果指定的主題處於狀態"unknown"，HealthLake 會在訂閱也處於作用中"active"狀態時將狀態更新為
- 成功建立訂閱通常需要 5-10 分鐘
- 如果 Subscription 未成功建立，狀態將變更為您應執行 DELETE 操作error的位置，然後重試建立訂閱。您可以在訂閱資源中檢視 "error" 欄位，以查看訂閱無法成功建立的原因。

4. 訂閱為時的擷取 active

5. 雖然 Subscription是 active

- HealthLake 會監控符合您條件的事件
- 發生相符項目時，通知會傳送至設定的端點

6. 錯誤處理

- HealthLake 會嘗試重試 14 天，然後停止重試這些事件

7. 停用

- Subscription 可以停用：

設定結束日期（自動停用）

```
{
  "resourceType": "Subscription",
  "meta": {
    "profile": [
```

```

    "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-
subscription"
  ],
  },
  "status": "requested",
  "end": "2026-07-31T05:38:17.2404292+00:00",
  "reason": "Test subscription for walkthrough",
  "criteria": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<datastoreId>/r4/SubscriptionTopic/<your topic id>",
  "_criteria": {
    "extension": [
      {
        "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-filter-criteria",
        "valueString": "Encounter?subject=Patient/<patient id>"
      }
    ]
  },
  "channel": {
    "type": "event-bridge",
    "endpoint": "<your event bus arn>",
    "payload": "application/fhir+json",
    "_payload": {
      "extension": [
        {
          "url": "http://hl7.org/fhir/uv/subscriptions-backport/
StructureDefinition/backport-payload-content",
          "valueCode": "id-only"
        }
      ]
    }
  }
}

```

刪除Subscription資源

```
DELETE https://<baseHealthLakeURL>/Subscription/<your subscription resource id>
```

使用 建立 FHIR 訂閱 AWS HealthLake

下列指南說明如何使用 建立 FHIR 訂閱 AWS HealthLake。

建立 FHIR 訂閱

1. 建立 SubscriptionTopic。

範例訂閱主題資源：

```
{
  "resourceType": "SubscriptionTopic",
  "url": "http://example.org/FHIR/SubscriptionTopic/encounter-create",
  "version": "1.0.0-fhir.r4b",
  "title": "encounter-create",
  "status": "unknown",
  "description": "Example topic for new encounters",
  "resourceTrigger": [
    {
      "description": "Encounter Create",
      "resource": "Encounter",
      "supportedInteraction": ["create", "update"]
    }
  ]
}
```

2. 準備您的通知端點（自訂頻道）。下列步驟是確保端點會收到通知的必要步驟

使用 REST Hook 時

- 如果使用 CM_CMK 資料存放區，events.amazonaws.com請信任您的 KMS 金鑰政策。
- 如果使用 CM_CMK 資料存放區，您必須將EventBridgeApiDestinations標籤新增至值為的 KMS 金鑰 true
- HealthLake 使用 OAuth 來驗證 REST Hook 端點。因此，建立 REST hook 訂閱時，您必須在 channel_type.extension **【*】** 中傳入 client-id、client-secret 和 oAuth-endpoint-url。

如果使用 CM_CMK 資料存放區，範例 KMS 金鑰政策：

```
{
  "Sid": "AllowEventBridgeToUseKMSKey",
  "Effect": "Allow",
  "Principal": {
    "Service": ["events.amazonaws.com", "healthlake.amazonaws.com"]
  },
  "Action": ["kms:GenerateDataKey*", "kms:Decrypt", "kms:DescribeKey"],
  "Resource": "*"
}
```

```
}

```

使用 EventBridge 時

- 如果使用 CM_CMK 資料存放區，events.amazonaws.com 請信任您的 KMS 金鑰政策。
- 驗證您的 EventBridge 資源政策信任 healthlake.amazonaws.com 為服務主體。
- 使用 CM_CMK 和 EventBridge 是端點時，請確認您使用與資料存放區 KMS 金鑰相同的 KMS 金鑰來加密 EventBridge 匯流排。
- 確認您的 EventBridge 匯流排至少有 1 個規則符合 HealthLake 產生的事件。

EventBridge 頻道匯流排的資源政策範例：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "allowHealthlakeToPutEvents",
      "Effect": "Allow",
      "Principal": {
        "Service": "healthlake.amazonaws.com"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:healthlake:us-east-1:111122223333:event-bus/
FhirSubscriptions-bus"
    }
  ]
}
```

從 HealthLake 接收事件的 EventBridge 規則事件模式範例：

```
{
  "detail-type": ["FHIR Subscription Notification"],
  "source": ["healthlake"]
}
```

Note

HealthLake 支援 2 個來源：

- “healthlake”：僅適用於訂閱。
- “aws.healthlake”：接收 HealthLake 服務事件。

建立 FHIR 訂閱事件匯流排的規則時，請使用 “healthlake” 做為來源。

3. 建立您的 Subscription

使用下列項目提交訂閱資源：

- 狀態："requested"
- 您所選 ID SubscriptionTopic 的參考
- 篩選條件。如需詳細資訊，請參閱篩選支援篩選條件的通知。
- 頻道組態

範例訂閱承載

下列程式碼範例示範如何建立訂閱承載。

EventBridge

具有 event-bridge 頻道和 id-only 承載類型的訂閱。

```
{
  "resourceType": "Subscription",
  "meta": {
    "profile": [
      "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-
subscription"
    ]
  },
  "status": "requested",
  "end": "2026-07-31T05:38:17.2404292+00:00",
  "reason": "Test subscription for walkthrough",
  "criteria": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<datastoreId/r4/
SubscriptionTopic/<your topic id>",
  "_criteria": {
    "extension": [
      {
        "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-filter-criteria",
        "valueString": "Encounter?subject=Patient/<patient id>"
      }
    ]
  },
  "channel": {
```

```

    "type": "event-bridge",
    "endpoint": "arn:aws:healthlake:eu-west-2:111122223333:event-bus/FhirSubscriptions-
bus",
    "payload": "application/fhir+json",
    "_payload": {
      "extension": [
        {
          "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-payload-content",
          "valueCode": "id-only"
        }
      ]
    }
  }
}

```

具有event-bridge端點和full-resource承載類型的訂閱。

```

{
  "resourceType": "Subscription",
  "meta": {
    "profile": [
      "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-
subscription"
    ]
  },
  "status": "requested",
  "end": "2026-07-31T05:38:17.2404292+00:00",
  "reason": "Test subscription for walkthrough",
  "criteria": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<datastoreId>/
r4/SubscriptionTopic/<your topic id>",
  "_criteria": {
    "extension": [
      {
        "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-filter-criteria",
        "valueString": "Encounter?subject=Patient/<patient id>"
      }
    ]
  },
  "channel": {
    "type": "event-bridge",
    "endpoint": "<your event bus arn>",
  }
}

```

```

    "payload": "application/fhir+json",
    "_payload": {
      "extension": [
        {
          "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-payload-content",
          "valueCode": "full-resource"
        }
      ]
    }
  }
}

```

Rest 勾點

具有rest-hook端點和id-only承載類型的訂閱。

```

{
  "resourceType": "Subscription",
  "meta": {
    "profile": [
      "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-
subscription"
    ]
  },
  "status": "requested",
  "end": "2026-07-31T05:38:17.2404292+00:00",
  "reason": "Test subscription for walkthrough",
  "criteria": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<datastoreId>/
r4/SubscriptionTopic/<your topic id>",
  "_criteria": {
    "extension": [
      {
        "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-filter-criteria",
        "valueString": "Encounter?subject=Patient/<your patient id>"
      }
    ]
  },
  "channel": {
    "type": "rest-hook",
    "_type": {
      "extension": [
        {

```

```

        "url": "http://healthlake.amazonaws.com/channel-type-clientId",
        "valueString": "<CLIENT_ID>"
    },
    {
        "url": "http://healthlake.amazonaws.com/channel-type-clientSecret",
        "valueString": "<CLIENT_SECRET>"
    },
    {
        "url": "http://healthlake.amazonaws.com/channel-type-oauth-endpoint",
        "valueUri": "<OAUTH_ENDPOINT_URL>"
    }
]
},
"endpoint": "<YOUR_REST_HOOK_ENDPOINT>",
"payload": "application/fhir+json",
"_payload": {
    "extension": [
        {
            "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-payload-content",
            "valueCode": "id-only"
        }
    ]
}
}
}
}
}

```

具有rest-hook頻道和full-resource承載類型的訂閱。

```

{
    "resourceType": "Subscription",
    "meta": {
        "profile": [
            "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-
subscription"
        ]
    },
    "status": "requested",
    "end": "2026-07-31T05:38:17.2404292+00:00",
    "reason": "Test subscription for walkthrough",
    "criteria": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<datastoreId>/
r4/SubscriptionTopic/<your topic id>",
    "_criteria": {

```

```
    "extension": [
      {
        "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-filter-criteria",
        "valueString": "Encounter?subject=Patient/test-patient-id"
      }
    ]
  },
  "channel": {
    "type": "rest-hook",
    "_type": {
      "extension": [
        {
          "url": "http://healthlake.amazonaws.com/channel-type-clientId",
          "valueString": "<CLIENT_ID>"
        },
        {
          "url": "http://healthlake.amazonaws.com/channel-type-clientSecret",
          "valueString": "<CLIENT_SECRET>"
        },
        {
          "url": "http://healthlake.amazonaws.com/channel-type-oauth-endpoint",
          "valueUri": "<OAUTH_ENDPOINT_URL>"
        }
      ]
    },
    "endpoint": "<YOUR_REST_HOOK_ENDPOINT>",
    "payload": "application/fhir+json",
    "_payload": {
      "extension": [
        {
          "url": "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/
backport-payload-content",
          "valueCode": "full-resource"
        }
      ]
    }
  }
}
```

通知承載範例

建立訂閱時，HealthLake 會傳送交握套件到您設定的頻道，以檢查訂閱設定是否成功。下列承載是交握套件的範例。

```
{
  "version": "0",
  "id": "<your-id>",
  "detail-type": "FHIR Subscription Notification",
  "source": "healthlake",
  "account": "436845984719",
  "time": "2025-09-04T23:43:50Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "subscriptionUrl": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>",
    "notificationBundlePayload": {
      "resourceType": "Bundle",
      "id": "<BUNDLE_ID>",
      "type": "history",
      "timestamp": "2025-09-04T23:43:50.341791934Z",
      "status": "requested",
      "entry": [
        {
          "fullUrl": "urn:uuid:<HANDSHAKE_RESOURCE_ID>",
          "resource": {
            "resourceType": "SubscriptionStatus",
            "id": "<HANDSHAKE_RESOURCE_ID>",
            "status": "requested",
            "type": "handshake",
            "eventsSinceSubscriptionStart": "0",
            "subscription": {
              "reference": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>"
            },
            "topic": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<DS_ID>/
r4/SubscriptionTopic/<TOPIC_ID>"
          }
        ]
      ]
    }
  }
}
```

}

僅限 ID 通知套件範例。

```
{
  "version": "0",
  "id": "<your-id>",
  "detail-type": "FHIR Subscription Notification",
  "source": "healthlake",
  "account": "436845984719",
  "time": "2025-09-05T00:18:43Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "subscriptionUrl": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>",
    "notificationBundlePayload": {
      "resourceType": "Bundle",
      "id": "c74ea02a-9c69-4e34-85d6-e72720189574",
      "type": "history",
      "timestamp": "2025-09-05T00:18:43.393688851Z",
      "status": "requested",
      "entry": [
        {
          "fullUrl": "urn:uuid:173135e3-3c80-4b90-a10a-e01a1420fdea",
          "resource": {
            "resourceType": "SubscriptionStatus",
            "id": "173135e3-3c80-4b90-a10a-e01a1420fdea",
            "status": "active",
            "type": "event-notification",
            "eventsSinceSubscriptionStart": "-1",
            "subscription": {
              "reference": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>"
            },
            "topic": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<DS_ID>/
r4/SubscriptionTopic/<TOPIC_ID>",
            "notificationEvent": [
              {
                "eventNumber": "0",
                "timestamp": "2025-09-05T00:18:43.393775234Z",
                "focus": "Encounter/c5ae898f-bd96-44dd-a509-87fdbcf23b19",
```

```

        "additionalContext": "Encounter/c5ae898f-bd96-44dd-a509-87fdbcf23b19/
_history/1",
        "id": "8f4e9c1a-2b3d-4e5f-6a7b-8c9d0e1f2a3b"
    }
]
}
]
}
}
}
}
}
}

```

完整資源通知套件範例。

```

{
  "version": "0",
  "id": "d142bed8-db3f-445f-c4db-a843ad84121a",
  "detail-type": "FHIR Subscription Notification",
  "source": "healthlake",
  "account": "436845984719",
  "time": "2025-09-05T00:18:43Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "subscriptionUrl": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>",
    "notificationBundlePayload": {
      "resourceType": "Bundle",
      "id": "3d42c70f-4fa9-4b1a-98a7-43c0d0441115",
      "type": "history",
      "timestamp": "2025-09-05T00:18:43.845821667Z",
      "status": "requested",
      "entry": [
        {
          "fullUrl": "urn:uuid:1d005a09-a15c-4010-9675-1e8043ce08a8",
          "resource": {
            "resourceType": "SubscriptionStatus",
            "id": "1d005a09-a15c-4010-9675-1e8043ce08a8",
            "status": "active",
            "type": "event-notification",
            "eventsSinceSubscriptionStart": "-1",

```

```

        "subscription": {
            "reference": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/
<DS_ID>/r4/Subscription/<SUBSCRIPTION_ID>"
        },
        "topic": "https://healthlake.<AWS_REGION>.amazonaws.com/datastore/<DS_ID>/
r4/SubscriptionTopic/<TOPIC_ID>",
        "notificationEvent": [
            {
                "eventNumber": "0",
                "timestamp": "2025-09-05T00:18:43.845970754Z",
                "focus": "Encounter/82776529-59a0-4d63-bedb-82f6726d65b5",
                "additionalContext": "Encounter/82776529-59a0-4d63-bedb-82f6726d65b5/
_history/1",
                "id": "7a8b9c0d-1e2f-3a4b-5c6d-7e8f9a0b1c2d"
            }
        ]
    },
    {
        "fullUrl": "Encounter/82776529-59a0-4d63-bedb-82f6726d65b5",
        "resource": {
            "resourceType": "Encounter",
            "id": "82776529-59a0-4d63-bedb-82f6726d65b5",
            "status": "finished",
            "class": {
                "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
                "code": "AMB",
                "display": "ambulatory"
            },
            "subject": {
                "reference": "Patient/test-patient-id"
            },
            "meta": {
                "lastUpdated": "2025-09-05T00:18:43.219652906Z",
                "versionId": "1"
            }
        },
        "request": {
            "method": "CREATE",
            "url": "Encounter/82776529-59a0-4d63-bedb-82f6726d65b5"
        }
    }
]
}

```

```
}  
}
```

事件版本控制

HealthLake 預設支援 FHIR 歷史記錄。

若要了解您在通知套件中收到的資源版本：

- **full-resource**：由於完整資源套件包含整個資源，因此套件中 `entry[*]` 每個資源的版本都會包含在中。
- **僅限 ID**：套件不會包含任何資源資訊。HealthLake 包含透過 `entry[0].notificationEvent[*].additionalContext` 欄位比對並包含在套件中的版本。此欄位的格式為 `<ResourceType>/<ResourceId>/_history/<Version Id>`。如需詳細資訊，請參閱範例僅限 ID 承載中的 `additionalContext` 欄位。

事件重複偵測

HealthLake 的 FHIR 訂閱功能保證至少一個交付。這表示您可能會多次收到相同的事件，無論是在相同的套件中，或在不同的套件中。為了識別重複項目，HealthLake 為 中的通知套件中的每個事件提供唯一的 `IDentry[0].notificationEvent[*].id`。

此 ID 對符合和交付的特定事件版本是唯一的。例如，如果相同的事件更新兩次，且兩個更新都符合篩選條件，則您將會收到兩個具有相同事件參考的個別事件。它們將具有相同的 `notificationEvent[*].focus`，但將具有唯一的 `notificationEvent[*].id`。此外，這些事件可能會以不同的套件傳送，或在相同的通知套件中傳送。

使用 搜尋 FHIR 訂閱 AWS HealthLake

`Subscription` 和 `SubscriptionTopic` 資源也可以搜尋。HealthLake 支援 `Subscription` 和 `SubscriptionTopic` 資源的所有常見[搜尋參數](#)。

此外，我們透過下列參數支援其他搜尋功能：

訂閱

搜尋參數	Description	範例
contact	搜尋 R4 核心規格中的 Subscription.contact 欄位	Subscription?contact=phone
條件	搜尋 R4 核心規格中的 Subscription.criteria 欄位	Subscription?criteria=[baseUrl]/datastore/[datastoreId]/r4/SubscriptionTopic/[topicId]
payload	搜尋 R4 核心規格中的 Subscription.channel.payload 欄位	Subscription?payload=application/fhir+json
status	搜尋 R4 核心規格中的 Subscription.status 欄位	Subscription?status=error
type	搜尋 R4 核心規格中的 Subscription.channel.type 欄位	Subscription?topic=event-bridge
url	搜尋 R4 核心規格中的 Subscription.channel.endpoint 欄位	Subscription?url=[Subscription.channel.endpoint]
filter-criteria	以 URL "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-filter-criteria" 做為字串搜尋條件延伸欄位	Subscription?filter-criteria=Encounter?
自訂頻道	使用 URL "http://hl7.org/fhir/uv/subscriptions-backport/StructureDefinition/backport-channel-type" 做為編碼搜尋自訂頻道類型延伸欄位	Subscription?custom-channel=[System] [code]

搜尋參數	Description	範例
	範例訂閱承載	
payload-type	搜尋您指定承載格式的承載類型延伸欄位 (id-only 或 full-resource)	Subscription?payload-type=full-resource
主題	搜尋 Subscription.criteria 欄位，其中新增主題	Subscription?topic=[topicId]

SubscriptionTopic

搜尋參數	Description	範例
date	搜尋 SubscriptionTopic 資源中的日期欄位	SubscriptionTopic?date=[SubscriptionTopic.date]
derived-or-self	搜尋 SubscriptionTopic 資源中的 url 或 derivedFrom 欄位	SubscriptionTopic?derived-or-self=[SubscriptionTopic.url SubscriptionTopic.derivedFrom]
identifier	搜尋 SubscriptionTopic.identifier 欄位	SubscriptionTopic?identifier=[SubscriptionTopic.identifier]
資源	搜尋 SubscriptionTopic.resourceTrigger.resource 欄位	SubscriptionTopic?resource=Encounter
status	搜尋 SubscriptionTopic 的狀態	SubscriptionTopic?status=unknown

搜尋參數	Description	範例
標題	搜尋 SubscriptionTopic 的標題	SubscriptionTopic?title=admission
trigger-description	搜尋 SubscriptionTopic. resourceTrigger.description	SubscriptionTopic. trigger-description=resource moving to state 'in-progress'
url	在 URL 上搜尋 SubscriptionTopic	SubscriptionTopic?url=[SubscriptionTopic.url]
version	搜尋 SubscriptionTopic 的版本	SubscriptionTopic?version=1

使用 篩選通知 AWS HealthLake

在您的Subscription條件中使用標準 FHIR 搜尋參數來精簡您的通知。

支援的篩選條件

下表顯示 HealthLake 支援的訂閱篩選條件清單。

搜尋參數類型	FHIR 資料類型	Modifiers (修改器)	支援的字首
String	string	確切，包含遺失	
	HumanName		
	地址		
權杖	boolean	不是、文字、缺少	
	code		
	string		
	編碼		

搜尋參數類型	FHIR 資料類型	Modifiers (修改器)	支援的字首
	CodeableConcept 識別符 ContactPoint id		
Number	integer decimal positiveInt unsignedInt	missing	"eq"、"ne"、"gt"、"lt" "、"ge"、"le"、"sa"、" eb"、"ap"
Date	date dateTime 即時 Period Timing (計時)		"eq"、"ne"、"gt"、"lt" "、"ge"、"le"、"sa"、" eb"、"ap"
數量	數量 金錢 範圍 SimpleQuantity		"eq"、"ne"、"gt"、"lt" "、"ge"、"le"、"sa"、" eb"、"ap"
參考資料	參考資料	遺失、識別符、類型	

搜尋參數類型	FHIR 資料類型	Modifiers (修改器)	支援的字首
URI	uri	missing	
	url		
	正式		
	uid		
	oid		

支援的篩選條件範例

下表顯示 HealthLake 支援的訂閱篩選條件範例：

用途	篩選條件	Description
病患特定的觀察	<code>Observation?patient=Patient/[id]&status=final</code>	完成特定病患的觀察時取得通知
病患特定的觀察	<code>Patient?birthdate=gt2021</code>	在註冊或更新 2021 年之後出生的患者時取得通知
病患特定的觀察	<code>Condition?code=http://snomed.info/sct 39065001</code>	取得具有特定 SNOMED 代碼之條件的通知
病患特定的觀察	<code>Observation?code=http://loinc.org 8480-6&value-quantity=gt160</code>	取得高收縮壓讀數的通知

使用 匯入 FHIR 資料 AWS HealthLake

建立 HealthLake 資料存放區後，下一步是從 Amazon Simple Storage Service (S3) 儲存貯體匯入檔案。您可以使用 AWS 管理主控台 AWS CLI、或 AWS SDKs 啟動 FHIR 匯入任務。使用原生 AWS HealthLake 動作來啟動、描述和列出 FHIR 匯入任務。

Important (重要)

HealthLake 支援用於醫療保健資料交換的 [FHIR R4 規格](#)。如有需要，您可以與 [AWS HealthLake 合作夥伴](#) 合作，在匯入之前將運作狀態資料轉換為 FHIR R4 格式。

啟動 FHIR 匯入任務時，您可以指定 Amazon S3 儲存貯體輸入位置、Amazon S3 儲存貯體輸出位置（用於任務處理結果）、授予 HealthLake 存取 Amazon S3 儲存貯體的 IAM 角色，以及客戶擁有或 AWS 擁有的 AWS Key Management Service 金鑰。如需詳細資訊，請參閱 [設定匯入任務的許可](#)。

Note

您可以將匯入任務排入佇列。非同步匯入任務會以 FIFO（先進先出）方式處理。您可以像開始匯入任務一樣，將任務排入佇列。如果正在進行，它只會排入佇列。您可以在匯入任務進行時建立、讀取、更新或刪除 FHIR 資源。

HealthLake 會為每個 FHIR 匯入任務產生 `manifest.json` 檔案。檔案說明 FHIR 匯入任務的成功和失敗。HealthLake 會將 `manifest.json` 檔案輸出到啟動 FHIR 匯入任務時指定的 Amazon S3 儲存貯體。日誌檔案會組織成兩個資料夾，名為 SUCCESS 和 FAILURE。使用 `manifest.json` 檔案做為故障匯入任務疑難排解的第一步，因為它提供每個檔案的詳細資訊。

```
{
  "inputDataConfig": {
    "s3Uri": "s3://amzn-s3-demo-source-bucket/healthlake-input/invalidInput/"
  },
  "outputDataConfig": {
    "s3Uri": "s3://amzn-s3-demo-logging-bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/",
    "encryptionKeyID": "arn:aws:kms:us-west-2:123456789012:key/fbbbfee3-20b3-42a5-a99d-c48c655ed545"
  },
  "successOutput": {
```

```

    "successOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
SUCCESS/"
  },
  "failureOutput": {
    "failureOutputS3Uri": "s3://amzn-s3-demo-logging-
bucket/32839038a2f47f17c2fe0f53f0c3a0ba-FHIR_IMPORT-19dd7bb7bcc8ee12a09bf6d322744a3d/
FAILURE/"
  },
  "numberOfScannedFiles": 1,
  "numberOfFilesImported": 1,
  "sizeOfScannedFilesInMB": 0.023627,
  "sizeOfDataImportedSuccessfullyInMB": 0.011232,
  "numberOfResourcesScanned": 9,
  "numberOfResourcesImportedSuccessfully": 4,
  "numberOfResourcesWithCustomerError": 5,
  "numberOfResourcesWithServerError": 0
}

```

設定匯入的驗證層級

啟動 FHIR 匯入任務時，您可以選擇指定 `ValidationLevel` 要套用至每個資源的。AWS HealthLake 目前支援下列驗證層級：

- `strict`：資源會根據資源的設定檔元素進行驗證，如果沒有設定檔，則為 R4 規格。這是的預設驗證層級 AWS HealthLake。
- `structure-only`：資源會根據 R4 驗證，忽略任何參考的設定檔。
- `minimal`：最少驗證資源，忽略某些 R4 規則。搜尋/分析所需的結構檢查失敗的資源將更新為包含稽核警告。

使用 `minimal` 驗證層級匯入時，可能會在名為 `extension` 的資料夾中產生其他日誌檔案 `SUCCESS_WITH_SEARCH_VALIDATION_FAILURES`。即使搜尋相關驗證檢查失敗，此資料夾日誌檔案中的資源仍會擷取至您的資料存放區。這表示 FHIR 資源的某些層面根據 FHIR 無效，格式不正確的欄位可能無法搜尋。這些資源將 `extension` 附加至描述上述失敗的。

主題

- [啟動 FHIR 匯入任務](#)
- [取得 FHIR 匯入任務屬性](#)
- [列出 FHIR 匯入任務](#)

啟動 FHIR 匯入任務

使用在 HealthLake 資料存放區中 `StartFHIRImportJob` 啟動 FHIR 匯入任務。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [StartFHIRImportJob](#)。

Important (重要)

HealthLake 支援用於醫療保健資料交換的 [FHIR R4 規格](#)。如有需要，您可以與 [AWS HealthLake 合作夥伴](#) 合作，在匯入之前將運作狀態資料轉換為 FHIR R4 格式。

啟動 FHIR 匯入任務

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

啟動 FHIR 匯入任務

下列 `start-fhir-import-job` 範例示範如何使用 AWS HealthLake 啟動 FHIR 匯入任務。

```
aws healthlake start-fhir-import-job \  
  --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --job-output-data-config '{"S3Configuration": {"S3Uri": "s3://(Bucket Name)/  
(Prefix Name)/", "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-  
b56c-4216-a250-f4c43ef46e83"}}' \  
  --datastore-id (Data store ID) \  
  --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)"
```

輸出：

```
{  
  "DatastoreId": "(Data store ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "c145fbb27b192af392f8ce6e7838e34f"  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [StartFHIRImportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def start_fhir_import_job(
    self,
    job_name: str,
    datastore_id: str,
    input_s3_uri: str,
    job_output_s3_uri: str,
    kms_key_id: str,
    data_access_role_arn: str,
) -> dict[str, str]:
    """
    Starts a HealthLake import job.
    :param job_name: The import job name.
    :param datastore_id: The data store ID.
    :param input_s3_uri: The input S3 URI.
    :param job_output_s3_uri: The job output S3 URI.
    :param kms_key_id: The KMS key ID associated with the output S3 bucket.
    :param data_access_role_arn: The data access role ARN.
    :return: The import job.
    """
    try:
        response = self.health_lake_client.start_fhir_import_job(
            JobName=job_name,
            InputDataConfig={"S3Uri": input_s3_uri},
            JobOutputDataConfig={
```

```
        "S3Configuration": {
            "S3Uri": job_output_s3_uri,
            "KmsKeyId": kms_key_id,
        }
    },
    DataAccessRoleArn=data_access_role_arn,
    DatastoreId=datastore_id,
)
return response
except ClientError as err:
    logger.exception(
        "Couldn't start import job. Here's why %s",
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [StartFHIRImportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

TRY.

```
" iv_job_name = 'MyImportJob'
" iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'
" iv_job_output_s3_uri = 's3://my-bucket/import/output/'
```

```

    " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
    " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeImportRole'
    oo_result = lo_hll->startfhirimportjob(
      iv_jobname = iv_job_name
      io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =
iv_input_s3_uri )
      io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
        io_s3configuration = NEW /aws1/cl_hlls3configuration(
          iv_s3uri = iv_job_output_s3_uri
          iv_kmskeyid = iv_kms_key_id
        )
      )
      iv_dataaccessrolearn = iv_data_access_role_arn
      iv_datastoreid = iv_datastore_id
    ).
    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
  ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [StartFHIRImportJob](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的[資料存放區](#)頁面。
2. 選擇資料存放區。
3. 選擇匯入。

匯入頁面隨即開啟。

4. 在輸入資料區段下，輸入下列資訊：

- Amazon S3 中的輸入資料位置

5. 在匯入輸出檔案區段下，輸入下列資訊：

- 在 Amazon S3 中匯入輸出檔案位置

- 匯入輸出檔案加密

6. 在存取許可區段下，選擇使用現有的 IAM 服務角色，然後從服務角色名稱功能表中選取角色，或選擇建立 IAM 角色。
7. 選擇匯入資料。

Note

在匯入期間，選擇頁面頂端橫幅上的複製任務 ID。您可以使用 [JobID](#) 來請求使用匯入任務屬性 AWS CLI。如需詳細資訊，請參閱[取得 FHIR 匯入任務屬性](#)。

取得 FHIR 匯入任務屬性

使用 `DescribeFHIRImportJob` 取得 FHIR 匯入任務屬性。下列功能表提供 AWS CLI AWS SDKs 的 AWS 管理主控台 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [DescribeFHIRImportJob](#)。

取得 FHIR 匯入任務屬性

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

描述 FHIR 匯入任務

下列 `describe-fhir-import-job` 範例顯示如何使用 AWS HealthLake 學習 FHIR 匯入任務的屬性。

```
aws healthlake describe-fhir-import-job \  
  --datastore-id (Data store ID) \  
  --job-id c145fbb27b192af392f8ce6e7838e34f
```

輸出：

```
{  
  "ImportJobProperties": {  
    "InputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"  
      { "arrayitem2": 2 }  
    },  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "COMPLETED",  
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",  
    "SubmitTime": 1606272542.161,  
    "EndTime": 1606272609.497,  
    "DatastoreId": "(Data store ID)"  
  }  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRImportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":
```

```
"""
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def describe_fhir_import_job(
    self, datastore_id: str, job_id: str
) -> dict[str, any]:
    """
    Describes a HealthLake import job.
    :param datastore_id: The data store ID.
    :param job_id: The import job ID.
    :return: The import job description.
    """
    try:
        response = self.health_lake_client.describe_fhir_import_job(
            DatastoreId=datastore_id, JobId=job_id
        )
        return response["ImportJobProperties"]
    except ClientError as err:
        logger.exception(
            "Couldn't describe import job with ID %s. Here's why %s",
            job_id,
            err.response["Error"]["Message"],
        )
        raise
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRImportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  oo_result = lo_hll->describefhirimportjob(
    iv_datastoreid = iv_datastore_id
    iv_jobid = iv_job_id
  ).
  DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).
  IF lo_import_job_properties IS BOUND.
    DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).
    MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.
  ENDIF.
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_notfound_ex.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRImportJob](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

Note

FHIR 匯入任務資訊不適用於 HealthLake 主控台。反之，使用 AWS CLI 搭配 `DescribeFHIRImportJob` 來請求匯入任務屬性，例如 [JobStatus](#)。如需詳細資訊，請參閱此頁面 AWS CLI 的範例。

列出 FHIR 匯入任務

使用 `ListFHIRImportJobs` 列出作用中 HealthLake 資料存放區的 FHIR 匯入任務。下列功能表提供 AWS 管理主控台 和 SDKs 的 AWS CLI 和 AWS 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [ListFHIRImportJobs](#)。

列出 FHIR 匯入任務

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

列出所有 FHIR 匯入任務

下列 `list-fhir-import-jobs` 範例示範如何使用命令來檢視與帳戶相關聯的所有匯入任務的清單。

```
aws healthlake list-fhir-import-jobs \  
  --datastore-id (Data store ID) \  
  --submitted-before (DATE Like 2024-10-13T19:00:00Z) \  
  --submitted-after (DATE Like 2020-10-13T19:00:00Z ) \  
  --output-type text
```

```
--job-name "FHIR-IMPORT" \  
--job-status SUBMITTED \  
-max-results (Integer between 1 and 500)
```

輸出：

```
{  
  "ImportJobPropertiesList": [  
    {  
      "JobId": "c0fddb76f238297632d4aebdbfc9ddf",  
      "JobStatus": "COMPLETED",  
      "SubmitTime": "2024-11-20T10:08:46.813000-05:00",  
      "EndTime": "2024-11-20T10:10:09.093000-05:00",  
      "DatastoreId": "(Data store ID)",  
      "InputDataConfig": {  
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"  
      },  
      "JobOutputDataConfig": {  
        "S3Configuration": {  
          "S3Uri": "s3://(Bucket Name)/  
import/6407b9ae4c2def3cb6f1a46a0c599ec0-FHIR_IMPORT-  
c0fddb76f238297632d4aebdbfc9ddf/",  
          "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/b7f645cb-  
e564-4981-8672-9e012d1ff1a0"  
        }  
      },  
      "JobProgressReport": {  
        "TotalNumberOfScannedFiles": 1,  
        "TotalSizeOfScannedFilesInMB": 0.001798,  
        "TotalNumberOfImportedFiles": 1,  
        "TotalNumberOfResourcesScanned": 1,  
        "TotalNumberOfResourcesImported": 1,  
        "TotalNumberOfResourcesWithCustomerError": 0,  
        "TotalNumberOfFilesReadWithCustomerError": 0,  
        "Throughput": 0.0  
      },  
      "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)"  
    }  
  ]  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRImportJobs](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_fhir_import_jobs(
    self,
    datastore_id: str,
    job_name: str = None,
    job_status: str = None,
    submitted_before: datetime = None,
    submitted_after: datetime = None,
) -> list[dict[str, any]]:
    """
    Lists HealthLake import jobs satisfying the conditions.
    :param datastore_id: The data store ID.
    :param job_name: The import job name.
    :param job_status: The import job status.
    :param submitted_before: The import job submitted before the specified
    date.
    :param submitted_after: The import job submitted after the specified
    date.
    :return: A list of import jobs.
    """
    try:
        parameters = {"DatastoreId": datastore_id}
        if job_name is not None:
            parameters["JobName"] = job_name
        if job_status is not None:
            parameters["JobStatus"] = job_status
        if submitted_before is not None:
            parameters["SubmittedBefore"] = submitted_before
```

```
    if submitted_after is not None:
        parameters["SubmittedAfter"] = submitted_after
    next_token = None
    jobs = []
    # Loop through paginated results.
    while True:
        if next_token is not None:
            parameters["NextToken"] = next_token
        response =
self.health_lake_client.list_fhir_import_jobs(**parameters)
        jobs.extend(response["ImportJobPropertiesList"])
        if "NextToken" in response:
            next_token = response["NextToken"]
        else:
            break
    return jobs
except ClientError as err:
    logger.exception(
        "Couldn't list import jobs. Here's why %s",
        err.response["Error"]["Message"],
    )
    raise
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRImportJobs](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  IF iv_submitted_after IS NOT INITIAL.  
    oo_result = lo_hll->listfhirimportjobs(  
      iv_datastoreid = iv_datastore_id  
      iv_submittedafter = iv_submitted_after  
    ).  
  ELSE.  
    oo_result = lo_hll->listfhirimportjobs(  
      iv_datastoreid = iv_datastore_id  
    ).  
  ENDIF.  
  DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).  
  DATA(lv_job_count) = lines( lt_import_jobs ).  
  MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_validation_ex.  
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
  lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
  MESSAGE lv_error TYPE 'I'.  
  RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIRImportJobs](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

Note

FHIR 匯入任務資訊不適用於 HealthLake 主控台。反之，使用 AWS CLI 搭配 `ListFHIRImportJobs` 列出所有 FHIR 匯入任務。如需詳細資訊，請參閱此頁面 AWS CLI 的範例。

在 中管理 FHIR 資源 AWS HealthLake

使用 FHIR R4 RESTful API 互動來管理 HealthLake 資料存放區中的 FHIR 資源。下列各節說明可用於 FHIR 資源管理的所有 HealthLake 支援的 FHIR R4 RESTful API 互動。如需 HealthLake 資料存放區功能及其支援哪些 FHIR 規格部分的相關資訊，請參閱 [的 FHIR R4 功能陳述式 AWS HealthLake](#)。

Note

本章列出的 FHIR 互動是根據 HL7 FHIR R4 標準的醫療保健資料交換而建置。由於它們是 HL7 FHIR 服務的表示，因此不會透過 AWS CLI 和 AWS SDKs 提供。如需詳細資訊，請參閱 FHIR [R4 RESTful API](#) 文件中的 RESTful API。 R4 RESTful

下表列出 支援的 FHIR R4 互動 AWS HealthLake。如需 HealthLake 支援的 FHIR 資源類型相關資訊，請參閱 [資源類型](#)。

支援的 FHIR R4 互動 AWS HealthLake

互動	Description
整個系統互動	
capabilities	取得系統的 功能陳述式。請參閱 的 FHIR R4 功能陳述式 AWS HealthLake 。
batch	在單一互動中更新、建立或刪除一組資源。請參閱 綁定 FHIR 資源 。
類型層級互動	
create	使用伺服器指派的 ID 建立新的資源。請參閱 建立 FHIR 資源 。
search	根據一些篩選條件搜尋資源類型。請參閱 搜尋 FHIR 資源 。
history	擷取特定資源類型的變更歷史記錄。請參閱 讀取 FHIR 資源歷史記錄 。
執行個體層級互動	
read	讀取資源的目前狀態。請參閱 讀取 FHIR 資源 。

互動	Description
history	讀取特定資源的變更歷史記錄。請參閱 讀取 FHIR 資源歷史記錄 。
vread	讀取特定版本資源的狀態。請參閱 讀取版本特定的 FHIR 資源歷史記錄 。
update	依資源的 ID 更新資源（如果是新的，則建立資源）。請參閱 更新 FHIR 資源 。
delete	刪除資源。請參閱 刪除 FHIR 資源 。

主題

- [建立 FHIR 資源](#)
- [讀取 FHIR 資源](#)
- [讀取 FHIR 資源歷史記錄](#)
- [更新 FHIR 資源](#)
- [使用 PATCH 操作修改資源](#)
- [綁定 FHIR 資源](#)
- [刪除 FHIR 資源](#)
- [冪等和並行](#)

建立 FHIR 資源

FHIR create 互動會在 HealthLake 資料存放區中建立新的 FHIR 資源。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [create](#) 中的。

建立 FHIR 資源

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
2. 決定 Resource 要建立的 FHIR 類型。如需詳細資訊，請參閱 [資源類型](#)。
3. 使用 HealthLake region 和 的收集值來建構請求的 URLdatastoreId。同時包含要建立的 FHIR Resource 類型。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource
```

4. 為請求建構 JSON 內文，並指定新資源的 FHIR 資料。基於此程序的目的，我們使用 FHIR Patient 資源，因此請將檔案儲存為 create-patient.json。

```
{
  "resourceType": "Patient",
  "identifier": [
    {
      "system": "urn:oid:1.2.36.146.595.217.0.1",
      "value": "12345"
    }
  ],
  "name": [
    {
      "family": "Silva",
      "given": [
        "Ana",
        "Carolina"
      ]
    }
  ],
  "gender": "female",
  "birthDate": "1992-02-10"
}
```

5. 傳送 請求。FHIR create 互動使用具有 [AWS 簽章第 4 版](#) 或 FHIR 授權上的 SMART 的 POST 請求。下列範例會使用 curl 或 HealthLake 主控台在 HealthLake 中建立 FHIR Patient 資源。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request POST \
  'https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/Patient' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json' \
  --data @create-patient.json
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\": \"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\": [\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"], \"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"], \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\", \"permission-v2\"]}"
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

AWS Console

Note

HealthLake 主控台僅支援 [AWS SigV4](#) 授權。

1. 登入 HealthLake 主控台上的 [執行查詢](#) 頁面。
2. 在查詢設定區段下，進行下列選擇。
 - 資料存放區 ID — 選擇資料存放區 ID 以產生查詢字串。
 - 查詢類型 — 選擇 Create。
 - 資源類型 — 選擇要建立的 FHIR [資源類型](#)。

- 請求內文 - 建構請求的 JSON 內文，指定新資源的 FHIR 資料。

3. 選擇 Run query (執行查詢)。

設定資源建立的驗證層級

建立 FHIR 資源時，您可以選擇指定 `x-amzn-healthlake-fhir-validation-level` HTTP 標頭來設定資源的驗證層級。AWS HealthLake 目前支援下列驗證層級：

- `strict`：資源會根據資源的設定檔元素進行驗證，如果沒有設定檔，則為 R4 規格。這是的預設驗證層級 AWS HealthLake。
- `structure-only`：資源會根據 R4 驗證，忽略任何參考的設定檔。
- `minimal`：最少驗證資源，忽略某些 R4 規則。搜尋/分析所需的結構檢查失敗的資源將更新為包含稽核警告。

雖然搜尋索引所需的驗證失敗，但使用最低驗證層級建立的資源仍可能會擷取至資料存放區。在這種情況下，資源將更新為包含 Healthlake 特定的延伸，以記錄上述失敗：

```
{
  "url": "http://healthlake.amazonaws.com/fhir/StructureDefinition/validation-issue",
  "valueString": "{\"resourceType\":\"OperationOutcome\",\"issue\":[{\"severity\":\"error\",\"code\":\"processing\",\"details\":{\"text\":\"FHIR resource in payload failed FHIR validation rules.\"}},{\"diagnostics\":\"FHIR resource in payload failed FHIR validation rules.\"}]}"
}
```

此外，以下 HTTP 回應標頭會包含在「true」的值中：

```
x-amzn-healthlake-validation-issues : true
```

Note

如果存在這些錯誤，則可能無法如預期搜尋根據 R4 規格格式不正確擷取的資料。

讀取 FHIR 資源

FHIR read 互動會讀取 HealthLake 資料存放區中資源的目前狀態。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [read](#) 中的。

讀取 FHIR 資源

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
2. 決定 Resource 要讀取和收集相關聯 id 值的 FHIR 類型。如需詳細資訊，請參閱 [資源類型](#)。
3. 使用 HealthLake region 和 的收集值來建構請求的 URL datastoreId。同時包含 FHIR Resource 類型及其相關聯的 id。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id
```

4. 傳送 請求。FHIR read 互動使用具有 [AWS 簽章第 4](#) 版或 FHIR 授權上的 SMART 的 GET 請求。下列 curl 範例會讀取 HealthLake 中 FHIR Patient 資源的目前狀態。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request GET \  
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id' \  
 \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{  
  "AuthorizationStrategy": "SMART_ON_FHIR",  
  "FineGrainedAuthorizationEnabled": true,  
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",  
}
```

```
"Metadata": "{\n  \"issuer\": \"https://ehr.example.com\", \"jwks_uri\":\n  \"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint\":\n  \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://\n  ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\":\n  [\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential\",\n  \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"],\n  \"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://\n  ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://\n  ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://\n  ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],\n  \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\",\n  \"permission-v2\"]\n}"
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

AWS Console

1. 登入 HealthLake 主控台上的 [執行查詢](#) 頁面。
2. 在查詢設定區段下，進行下列選擇。
 - 資料存放區 ID — 選擇資料存放區 ID 以產生查詢字串。
 - 查詢類型 — 選擇 Read。
 - 資源類型 — 選擇要讀取的 FHIR [資源類型](#)。
 - 資源 ID — 輸入 FHIR 資源 ID。
3. 選擇 Run query (執行查詢)。

讀取 FHIR 資源歷史記錄

FHIR history 互動會擷取 HealthLake 資料存放區中特定 FHIR 資源的歷史記錄。使用此互動，您可以判斷 FHIR 資源的內容如何隨時間變更。在協調稽核日誌時，查看修改前後的資源狀態也很有用。FHIR 互動 create、update 和 delete 會產生要儲存的歷史資源版本。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [history](#) 中的。

Note

您可以選擇不history接收特定 FHIR 資源類型。若要選擇退出，請使用 [建立案例AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。

讀取 FHIR 資源歷史記錄

1. 收集 HealthLake region和 datastoreId 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 決定Resource要讀取和收集相關聯id值的 FHIR 類型。如需詳細資訊，請參閱[資源類型](#)。
3. 使用 HealthLake region和 的收集值來建構請求的 URLdatastoreId。也包括 FHIR Resource類型、其相關聯的 id，以及選用的搜尋參數。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id/  
_history{?[parameters]}
```

HealthLake 支援的 FHIR **history** 互動搜尋參數

參數	Description
<code>_count : integer</code>	頁面上搜尋結果的數量上限。伺服器將傳回請求的數量，或資料存放區預設允許的搜尋結果數量上限，以較低者為準。
<code>_since : instant</code>	僅包含在特定即時時間或之後建立的資源版本。
<code>_at : date(Time)</code>	僅包含日期時間值中指定期間內某個時間點的目前資源版本。如需詳細資訊，請參閱 HL7 FHIR RESTful API 文件 date 中的。

4. 傳送 請求。FHIR history 互動使用具有[AWS 簽章第 4](#) 版或 FHIR 授權上的 SMART 的 GET 請求。下列 curl 範例使用 `_count` 搜尋參數，針對 HealthLake 中的 FHIR Patient 資源傳回每頁 100 個歷史搜尋結果。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request GET \  
  'https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/Patient/id/  
_history?_count=100' \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{  
  "AuthorizationStrategy": "SMART_ON_FHIR",  
  "FineGrainedAuthorizationEnabled": true,  
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-  
lambda-name",  
  "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\":  
\"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint  
\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://  
ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\":  
[\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credentia  
l\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/  
register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"],  
\"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://  
ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://  
ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://  
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],  
\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\",  
\"permission-v2\"]}"  
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

history 互動的傳回內容包含在 FHIR 資源 中Bundle，類型設定為 history。它包含指定的版本歷史記錄，最後以最舊的版本排序，並包含已刪除的資源。如需詳細資訊，請參閱 FHIR R4 文件 [Resource Bundle](#) 中的。

讀取版本特定的 FHIR 資源歷史記錄

FHIR vread 互動會對 HealthLake 資料存放區中的資源執行版本特定讀取。使用此互動，您可以像過去特定時間一樣檢視 FHIR 資源的內容。

Note

如果您在不使用的情況下使用 FHIR history 互動 vread，HealthLake 一律會傳回資源中繼資料的最新版本。

HealthLake 宣告支援 [CapabilityStatement.rest.resource.versioning](#) 針對每個支援的資源在中進行版本控制。所有 HealthLake 資料存放區在所有資源上都包含 Resource.meta.versionId(vid)。

啟用 FHIR history 互動時（預設為 10/25/2024 之後建立的資料存放區，或請求較舊的資料存放區），Bundle 回應會包含 vid 作為 [location](#) 元素的一部分。在下列範例中，vid 會顯示為數字 1。若要檢視完整範例，請參閱 [Bundle/bundle-response \(JSON\) 範例](#)。

```
"response" : {
  "status" : "201 Created",
  "location" : "Patient/12423/_history/1",
  ...}
```

讀取版本特定的 FHIR 資源歷史記錄

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
2. 決定要讀取和收集相關聯 id 和 vid 值的 FHIR Resource 類型。如需詳細資訊，請參閱 [資源類型](#)。
3. 使用針對 HealthLake 和 FHIR 收集的值來建構請求的 URL。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id/  
_history/vid
```

- 傳送 請求。FHIR history 互動使用具有 [AWS 簽章第 4 版](#) 或 FHIR 授權上的 SMART 的 GET 請求。下列 vread 互動會針對 指定的資源中繼資料版本，傳回具有 FHIR Patient 資源指定內容的單一執行個體 vid。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request GET \  
  'https://healthlake.region.amazonaws.com/datastore/datastore-id/r4/Patient/id/  
_history/vid' \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{  
  "AuthorizationStrategy": "SMART_ON_FHIR",  
  "FineGrainedAuthorizationEnabled": true,  
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",  
  "Metadata": "{\n\"issuer\": \"https://ehr.example.com\", \n\"jwks_uri\":  
\n\"https://ehr.example.com/.well-known/jwks.json\", \n\"authorization_endpoint  
\n\": \"https://ehr.example.com/auth/authorize\", \n\"token_endpoint\": \"https://  
ehr.token.com/auth/token\", \n\"token_endpoint_auth_methods_supported\":  
[\n\"client_secret_basic\", \n\"foo\"], \n\"grant_types_supported\": [\n\"client_credential  
\n\", \n\"foo\"], \n\"registration_endpoint\": \"https://ehr.example.com/auth/  
register\", \n\"scopes_supported\": [\n\"openid\", \n\"profile\", \n\"launch\"],  
\n\"response_types_supported\": [\n\"code\"], \n\"management_endpoint\": \"https://  
ehr.example.com/user/manage\", \n\"introspection_endpoint\": \"https://  
ehr.example.com/user/introspect\", \n\"revocation_endpoint\": \"https://  
ehr.example.com/user/revoke\", \n\"code_challenge_methods_supported\": [\n\"S256\"],  
\n\"capabilities\": [\n\"launch-ehr\", \n\"sso-openid-connect\", \n\"client-public\",  
\n\"permission-v2\"]}"
```

```
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱[OAuth 2.0 範圍](#)。

更新 FHIR 資源

FHIR update 互動會為現有資源建立新的目前版本，如果指定的 尚不存在資源，則會建立初始版本 id。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件[update](#)中的。

更新 FHIR 資源

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 決定 Resource 要更新並收集相關聯 id 值的 FHIR 類型。如需詳細資訊，請參閱[資源類型](#)。
3. 使用 HealthLake region 和 的收集值來建構請求的 URL datastoreId。同時包含 FHIR Resource 類型及其相關聯的 id。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
PUT https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id
```

4. 為請求建構 JSON 內文，指定要進行的 FHIR 資料更新。基於此程序的目的，請將檔案儲存為 update-patient.json。

```
{
  "id": "2de04858-ba65-44c1-8af1-f2fe69a977d9",
  "resourceType": "Patient",
  "active": true,
  "name": [
    {
      "use": "official",
      "family": "Doe",
      "given": [
        "Jane"
      ]
    },
    {
      "use": "usual",
      "given": [
        "Jane"
      ]
    }
  ]
}
```

```

    ],
    "gender": "female",
    "birthDate": "1985-12-31"
  }

```

5. 傳送 請求。FHIR update 互動使用具有 [AWS 簽章第 4 版](#) 或 FHIR 授權上的 SMART 的 PUT 請求。下列 curl 範例會更新 HealthLake 中的 Patient 資源。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```

curl --request PUT \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id' \
  \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json' \
  --data @update-patient.json

```

如果現有資源已更新，您的請求將傳回 200 HTTP 狀態碼，如果建立新資源，則傳回 201 HTTP 狀態碼。

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```

{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\n\"issuer\": \"https://ehr.example.com\", \n\"jwks_uri\": \n\"https://ehr.example.com/.well-known/jwks.json\", \n\"authorization_endpoint\": \n\"https://ehr.example.com/auth/authorize\", \n\"token_endpoint\": \"https://ehr.token.com/auth/token\", \n\"token_endpoint_auth_methods_supported\": [\n\"client_secret_basic\", \n\"foo\"], \n\"grant_types_supported\": [\n\"client_credential\", \n\"foo\"], \n\"registration_endpoint\": \"https://ehr.example.com/auth/register\", \n\"scopes_supported\": [\n\"openid\", \n\"profile\", \n\"launch\"], \n\"response_types_supported\": [\n\"code\"], \n\"management_endpoint\": \"https://ehr.example.com/user/manage\", \n\"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \n\"revocation_endpoint\": \"https://

```

```
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],  
  \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\",  
  \"permission-v2\"]}]\"  
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

AWS Console

1. 登入 HealthLake 主控台上的 [執行查詢](#) 頁面。
2. 在查詢設定區段下，進行下列選擇。
 - 資料存放區 ID — 選擇資料存放區 ID 以產生查詢字串。
 - 查詢類型 — 選擇 Update (PUT)。
 - 資源類型 — 選擇要更新或建立的 FHIR [資源類型](#)。
 - 請求內文 - 建構請求的 JSON 內文，指定要更新資源的 FHIR 資料。
3. 選擇 Run query (執行查詢)。

根據條件更新 FHIR 資源

條件式更新可讓您根據某些識別搜尋條件更新現有資源，而不是透過邏輯 FHIR id。當伺服器處理更新時，它會針對 資源類型使用其標準搜尋功能執行搜尋，目標是解決請求id的單一邏輯。

伺服器採取的動作取決於找到多少相符項目：

- 沒有相符項目，請求內文中未 **id** 提供：伺服器會建立 FHIR 資源。
- 不存在相符項目、**id** 提供的和資源：**id** 伺服器會將互動視為更新為建立互動。
- 沒有相符項目，**id** 已提供且已存在：伺服器拒絕更新時發生錯誤 409 Conflict。
- One Match，未 **id** 提供資源 OR (**id** 提供的資源，且符合找到的資源)：伺服器會針對相符的資源執行更新，如上所示，如果資源已更新，伺服器 SHALL 會傳回 200 OK。
- One Match，**id** 提供的資源，但不符合找到的資源：伺服器傳回 409 Conflict 錯誤，指出用戶端 ID 規格是 的問題 `OperationOutcome`
- 多個相符項目：伺服器傳回 412 Precondition Failed 錯誤，指出用戶端的條件不夠選擇性，最好使用 `OperationOutcome`

下列範例會更新名為 peter Patient 的資源，出生日期為 2000 年 1 月 1 日，電話號碼為 1234567890。

```
PUT https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
name=peter&birthdate=2000-01-01&phone=1234567890
```

設定資源更新的驗證層級

更新 FHIR 資源時，您可以選擇指定 `x-amzn-healthlake-fhir-validation-level` HTTP 標頭來設定資源的驗證層級。AWS HealthLake 目前支援下列驗證層級：

- `strict`：資源會根據資源的設定檔元素進行驗證，如果沒有設定檔，則為 R4 規格。這是的預設驗證層級 AWS HealthLake。
- `structure-only`：資源會根據 R4 驗證，忽略任何參考的設定檔。
- `minimal`：最少驗證資源，忽略某些 R4 規則。搜尋/分析所需的結構檢查失敗的資源將更新為包含稽核警告。

使用最低驗證層級更新的資源可能會擷取至資料存放區，即使搜尋索引所需的驗證失敗。在這種情況下，資源將更新為包含 Healthlake 特定的延伸，以記錄上述失敗：

```
{
  "url": "http://healthlake.amazonaws.com/fhir/StructureDefinition/validation-issue",
  "valueString": "{\"resourceType\":\"OperationOutcome\",\"issue\":[{\"severity\":\"error\",\"code\":\"processing\",\"details\":{\"text\":\"FHIR resource in payload failed FHIR validation rules.\"},\"diagnostics\":\"FHIR resource in payload failed FHIR validation rules.\"}]}"
}
```

此外，以下 HTTP 回應標頭會包含在「true」的值中：

```
x-amzn-healthlake-validation-issues : true
```

Note

請注意，如果出現這些錯誤，根據 R4 規格格式不正確擷取的資料可能無法如預期搜尋。

使用 PATCH 操作修改資源

AWS HealthLake 支援 FHIR 資源的 PATCH 操作，可讓您透過鎖定要新增、取代或刪除的特定元素來修改資源，而無需更新整個資源。當您需要執行下列動作時，此操作特別有用：

- 對大型資源進行目標性更新
- 減少網路頻寬用量
- 對特定資源元素執行原子修改
- 將覆寫並行變更的風險降至最低
- 在批次和交易工作流程中更新資源

支援的修補程式格式

AWS HealthLake 支援兩種標準 PATCH 格式：

JSON 修補程式 (RFC 6902)

使用 JSON 指標語法，依其在資源結構中的位置來鎖定元素。

Content-Type : application/json-patch+json

FHIRPath 修補程式 (FHIR R4 規格)

使用 FHIRPath 表達式，依其內容和關係來鎖定元素，提供 FHIR 原生的修補方法。

Content-Type : application/fhir+json

Usage

直接修補操作

您可以使用 PATCH HTTP 方法，直接在 FHIR 資源上叫用 PATCH 操作：

```
PATCH [base]/[resource-type]/[id]{?_format=[mime-type]}
```

套件中的 PATCH

PATCH 操作可以包含在類型 batch 或 的 FHIR 套件中作為項目 transaction，讓您能夠在單一請求中將修補程式操作與其他 FHIR 互動（建立、讀取、更新、刪除）結合。

- 交易套件：所有項目都會以原子方式成功或失敗
- 批次套件：每個項目都會獨立處理

JSON 修補程式格式

受支援的 操作

作業	Description
add	將新值新增至資源
remove	從資源移除值
replace	取代 資源中的現有值
move	從一個位置移除值，並將其新增至另一個位置
copy	將值從一個位置複製到另一個位置
test	測試目標位置的值是否等於指定的值

路徑語法

JSON 修補程式使用 JSON 指標語法 (RFC 6901)：

路徑範例	Description
/name/0/family	名字的系列元素
/telecom/-	附加至電信陣列
/active	根層級作用中元素
/address/0/ line/1	第一地址的第二行

範例

具有多個操作的直接 JSON 修補程式請求

```
PATCH [base]/Patient/example
Content-Type: application/json-patch+json
If-Match: W/"1"
```

```
[
  {
    "op": "replace",
    "path": "/name/0/family",
    "value": "Smith"
  },
  {
    "op": "add",
    "path": "/telecom/-",
    "value": {
      "system": "phone",
      "value": "555-555-5555",
      "use": "home"
    }
  },
  {
    "op": "remove",
    "path": "/address/0"
  },
  {
    "op": "move",
    "from": "/name/0/family",
    "path": "/name/1/family"
  },
  {
    "op": "test",
    "path": "/gender",
    "value": "male"
  },
  {
    "op": "copy",
    "from": "/name/0",
    "path": "/name/1"
  }
]
```

```
]
```

使用單一操作的直接 JSON 修補程式請求

```
PATCH [base]/Patient/example  
Content-Type: application/json-patch+json
```

```
[  
  {  
    "op": "replace",  
    "path": "/active",  
    "value": false  
  }  
]
```

套件中的 JSON 修補程式

使用包含 base64 編碼 JSON 修補程式承載的二進位資源：

```
{  
  "resourceType": "Bundle",  
  "type": "transaction",  
  "entry": [{  
    "resource": {  
      "resourceType": "Binary",  
      "contentType": "application/json-patch+json",  
      "data":  
      "W3sib3Ai0iJhZGQiLCJwYXRoIjoiL2JpcnRoRGF0ZSIsInZhbHVlIjoiMTk5MC0wMS0wMSJ9XQ=="  
    },  
    "request": {  
      "method": "PATCH",  
      "url": "Patient/123"  
    }  
  }  
}]  
}
```

FHIRPath 修補程式格式

受支援的 操作

作業	Description
add	將新元素新增至資源
insert	在清單中的特定位置插入 元素
delete	從資源移除 元素
replace	取代現有元素的值
move	重新排序清單中的元素

路徑語法

FHIRPath 修補程式使用 FHIRPath 表達式，支援：

- 索引型存取：`Patient.name[0]`
- 使用篩選 `where()`：`Patient.name.where(use = 'official')`
- 布林值邏輯：`Patient.telecom.where(system = 'phone' and use = 'work')`
- 子設定函數：`first()`、`last()`
- 存在性檢查：`exists()`、`count()`
- 多態導覽：`Observation.value`

範例

Direct FHIRPath 修補程式請求

```
PATCH [base]/Patient/123
Content-Type: application/fhir+json
Authorization: ...

{
```

```
"resourceType": "Parameters",
"parameter": [{
  "name": "operation",
  "part": [
    { "name": "type", "valueCode": "add" },
    { "name": "path", "valueString": "Patient" },
    { "name": "name", "valueString": "birthDate" },
    { "name": "value", "valueDate": "1990-01-01" }
  ]
}]
}
```

套件中的 FHIRPath 修補程式

透過 使用參數資源做為項目資源method: PATCH :

```
{
  "resourceType": "Bundle",
  "type": "transaction",
  "entry": [{
    "resource": {
      "resourceType": "Parameters",
      "parameter": [{
        "name": "operation",
        "part": [
          { "name": "type", "valueCode": "add" },
          { "name": "path", "valueString": "Patient" },
          { "name": "name", "valueString": "birthDate" },
          { "name": "value", "valueDate": "1990-01-01" }
        ]
      }]
    },
    "request": {
      "method": "PATCH",
      "url": "Patient/123"
    }
  ]
}
```

請求標頭

標頭	必要	描述
Content-Type	是	application/json-patch+json 適用於 JSON 修補程式或 application/fhir+json 適用於 FHIRPath 修補程式
If-Match	否	使用 ETag 的特定版本條件更新

回應範例

操作會傳回具有新版本資訊的更新資源：

```
HTTP/1.1 200 OK
Content-Type: application/fhir+json
ETag: W/"2"
Last-Modified: Mon, 05 May 2025 10:10:10 GMT

{
  "resourceType": "Patient",
  "id": "example",
  "active": true,
  "name": [
    {
      "family": "Smith",
      "given": ["John"]
    }
  ],
  "telecom": [
    {
      "system": "phone",
      "value": "555-555-5555",
      "use": "home"
    }
  ],
  "meta": {
    "versionId": "2",
    "lastUpdated": "2025-05-05T10:10:10Z"
  }
}
```

}

Behavior (行為)

PATCH 操作：

- 根據適當的規格驗證修補程式語法 (RFC 6902 for JSON Patch、FHIR R4 for FHIRPath Patch)
- 以原子方式套用操作 - 所有操作成功或全部失敗
- 更新資源版本 ID 並建立新的歷史記錄項目
- 在套用變更之前，保留歷史記錄中的原始資源
- 套用修補程式後驗證 FHIR 資源限制
- 支援使用 If-Match 標頭搭配 ETag 進行條件式更新

錯誤處理

操作會處理下列錯誤條件：

- 400 錯誤的請求：無效的修補程式語法（不符合的請求或格式錯誤的修補程式文件）
- 找不到 404：找不到資源（指定的 ID 不存在）
- 409 衝突：版本衝突（提供並行更新或非目前版本 ID）
- 422 無法處理的實體：修補程式操作無法套用至指定的資源元素

功能摘要

功能	JSON 修補程式	FHIRPath 修補程式
內容類型	application/json-patch+json	application/fhir+json
路徑格式	JSON 指標 (RFC 6901)	FHIRPath 表達式
Direct PATCH API	支援	支援
套件批次	支援（透過二進位）	支援（透過參數）

功能	JSON 修補程式	FHIRPath 修補程式
套件交易	支援 (透過二進位)	支援 (透過參數)
操作	新增、移除、取代、移動、複製、測試	新增、插入、刪除、取代、移動

限制

- 不支援使用搜尋條件的條件式 PATCH 操作
- 套件中的 JSON 修補程式必須使用具有 base64 編碼內容的二進位資源
- 套件中的 FHIRPath 修補程式必須使用參數資源

其他資源

如需 PATCH 操作的詳細資訊，請參閱：

- [FHIR R4 修補文件](#)
- [FHIR R4 FHIRPath 修補程式規格](#)
- [RFC 6902 - JSON 修補程式](#)
- [RFC 6901 - JSON 指標](#)

綁定 FHIR 資源

FHIR Bundle 是 FHIR 資源集合的容器 AWS HealthLake。AWS HealthLake 支援兩種具有不同處理行為的套件類型。

Batch 套件會獨立處理每個資源。如果某個資源失敗，剩餘的資源仍然可以成功。每個操作都會個別處理，即使某些操作失敗，也會繼續處理。將批次套件用於可接受部分成功的大量操作，例如上傳多個不相關的患者記錄。

Transaction 套件會以原子方式將所有資源處理為單一單位。所有資源操作都會成功，或不會 AWS HealthLake 遞交任何操作。當您需要跨相關資源的保證參考完整性時，請使用交易套件，例如建立具有相關觀察和條件的患者，其中所有資料必須一起記錄。

批次與交易套件之間的差異

功能	批次	交易
處理模型	每個操作都會獨立成功或失敗。	所有操作都會以單一原子單位的形式成功或失敗。
失敗處理	即使個別操作失敗，處理仍會繼續。	如果任何單一操作失敗，整個套件都會失敗。
執行順序	不保證執行順序。	操作會依指定的順序處理。
參考完整性	未跨操作強制執行。	針對套件內的本機參考資源強制執行。
最適合用於	可接受部分成功的大量操作。	必須一起建立或更新的相關資源。

您可以綁定相同或不同類型的 FHIR 資源，它們可以包含 FHIR 操作的混合，例如 create、read、update、delete 和 patch。如需詳細資訊，請參閱 FHIR R4 文件中的[資源套件](#)。

以下是每個套件類型的範例使用案例。

批次套件

- 在夜間資料同步期間，從不同的設施上傳多個不相關的病患記錄。
- 大量上傳一些記錄可能有驗證問題的歷史藥物記錄。
- 載入參考資料，例如組織和從業人員，其中個別失敗不會影響其他項目。

交易套件

- 在緊急部門許可期間建立具有相關觀察和條件的患者，其中所有資料必須一起記錄。
- 更新必須保持一致的患者藥物清單和相關敏感資訊。
- 將與病患、觀察、程序和帳單資訊的完整事件記錄為單一原子單位。

Important

批次和交易套件都使用相同的 Bundle 資源結構。唯一的差異是 type 欄位的值。

下列範例顯示具有多種資源類型和操作的交易套件。

```
{
  "resourceType": "Bundle",
  "type": "transaction",
  "entry": [
    {
      "fullUrl": "urn:uuid:4f6a30fb-cd3c-4ab6-8757-532101f72065",
      "resource": {
        "resourceType": "Patient",
        "id": "new-patient",
        "active": true,
        "name": [
          {
            "family": "Johnson",
            "given": [
              "Sarah"
            ]
          }
        ],
        "gender": "female",
        "birthDate": "1985-08-12",
        "telecom": [
          {
            "system": "phone",
            "value": "555-123-4567",
            "use": "home"
          }
        ]
      },
      "request": {
        "method": "POST",
        "url": "Patient"
      }
    },
    {
      "fullUrl": "urn:uuid:7f83f473-d8cc-4a8d-86d3-9d9876a3248b",
      "resource": {
        "resourceType": "Observation",
        "id": "blood-pressure",
        "status": "final",
        "code": {
          "coding": [
            {
```

```
        "system": "http://loinc.org",
        "code": "85354-9",
        "display": "Blood pressure panel"
    }
],
"text": "Blood pressure panel"
},
"subject": {
  "reference": "urn:uuid:4f6a30fb-cd3c-4ab6-8757-532101f72065"
},
"effectiveDateTime": "2023-10-15T09:30:00Z",
"component": [
  {
    "code": {
      "coding": [
        {
          "system": "http://loinc.org",
          "code": "8480-6",
          "display": "Systolic blood pressure"
        }
      ]
    },
    "valueQuantity": {
      "value": 120,
      "unit": "mmHg",
      "system": "http://unitsofmeasure.org",
      "code": "mm[Hg]"
    }
  },
  {
    "code": {
      "coding": [
        {
          "system": "http://loinc.org",
          "code": "8462-4",
          "display": "Diastolic blood pressure"
        }
      ]
    },
    "valueQuantity": {
      "value": 80,
      "unit": "mmHg",
      "system": "http://unitsofmeasure.org",
      "code": "mm[Hg]"
    }
  }
]
```

```
    }
  }
]
},
"request": {
  "method": "POST",
  "url": "Observation"
}
},
{
  "resource": {
    "resourceType": "Appointment",
    "id": "appointment-123",
    "status": "booked",
    "description": "Annual physical examination",
    "start": "2023-11-15T09:00:00Z",
    "end": "2023-11-15T09:30:00Z",
    "participant": [
      {
        "actor": {
          "reference": "urn:uuid:4f6a30fb-cd3c-4ab6-8757-532101f72065"
        },
        "status": "accepted"
      }
    ]
  },
  "request": {
    "method": "PUT",
    "url": "Appointment/appointment-123"
  }
},
{
  "request": {
    "method": "DELETE",
    "url": "MedicationRequest/med-request-456"
  }
}
]
```

將 FHIR 資源綁定為獨立實體

將 FHIR 資源綁定為獨立實體

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 使用 HealthLake region 和 的收集值來建構請求的 URLdatastoreId。請勿在 URL 中指定 FHIR 資源類型。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
```

3. 為請求建構 JSON 內文，將每個 HTTP 動詞指定為method元素的一部分。下列範例使用與 Bundle 資源的batch類型互動來建立新的 Medication Patient和資源。所有必要區段都會據此加上註解。基於此程序的目的，請將檔案儲存為 batch-independent.json。

```
{
  "resourceType": "Bundle",
  "id": "bundle-batch",
  "meta": {
    "lastUpdated": "2014-08-18T01:43:30Z"
  },
  "type": "batch",
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "meta": {
          "lastUpdated": "2022-06-03T17:53:36.724Z"
        },
        "text": {
          "status": "generated",
          "div": "Some narrative"
        },
        "active": true,
        "name": [
          {
            "use": "official",
            "family": "Jackson",
            "given": [
              "Mateo",
              "James"
            ]
          }
        ],
        "gender": "male",
        "birthDate": "1974-12-25"
      },
    },
  ],
}
```

```
    "request": {
      "method": "POST",
      "url": "Patient"
    }
  },
  {
    "resource": {
      "resourceType": "Medication",
      "id": "med0310",
      "contained": [
        {
          "resourceType": "Substance",
          "id": "sub03",
          "code": {
            "coding": [
              {
                "system": "http://snomed.info/sct",
                "code": "55452001",
                "display": "Oxycodone (substance)"
              }
            ]
          }
        }
      ],
      "code": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "430127000",
            "display": "Oral Form Oxycodone (product)"
          }
        ]
      },
      "form": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "385055001",
            "display": "Tablet dose form (qualifier value)"
          }
        ]
      },
      "ingredient": [
        {
```

```

        "itemReference": {
            "reference": "#sub03"
        },
        "strength": {
            "numerator": {
                "value": 5,
                "system": "http://unitsofmeasure.org",
                "code": "mg"
            },
            "denominator": {
                "value": 1,
                "system": "http://terminology.hl7.org/CodeSystem/
v3-orderableDrugForm",
                "code": "TAB"
            }
        }
    ],
},
"request": {
    "method": "POST",
    "url": "Medication"
}
}
]
}

```

- 傳送 請求。FHIR Bundle 批次類型使用具有 [AWS 簽章第 4 版](#) 或 FHIR 授權上的 SMART 的 POST 請求。下列程式碼範例使用 curl 命令列工具進行示範。

SigV4

SigV4 授權

```

curl --request POST \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json' \
  --data @batch-type.json

```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\n\"issuer\": \"https://ehr.example.com\", \n\"jwks_uri\": \"https://ehr.example.com/.well-known/jwks.json\", \n\"authorization_endpoint\": \"https://ehr.example.com/auth/authorize\", \n\"token_endpoint\": \"https://ehr.token.com/auth/token\", \n\"token_endpoint_auth_methods_supported\": [\"client_secret_basic\", \"foo\"], \n\"grant_types_supported\": [\"client_credentials\", \"foo\"], \n\"registration_endpoint\": \"https://ehr.example.com/auth/register\", \n\"scopes_supported\": [\"openid\", \"profile\", \"launch\"], \n\"response_types_supported\": [\"code\"], \n\"management_endpoint\": \"https://ehr.example.com/user/manage\", \n\"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \n\"revocation_endpoint\": \"https://ehr.example.com/user/revoke\", \n\"code_challenge_methods_supported\": [\"S256\"], \n\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\", \"permission-v2\"]}"
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

伺服器會傳回回應，顯示因Bundle批次類型請求而建立的 Medication Patient和資源。

套件中的條件式 PUTs

AWS HealthLake 支援使用下列查詢參數的套件內條件式更新：

Note

條件式 PUTs僅支援 batch 套件。Transaction套件不支援條件式 PUTs。

- `_id` (獨立)
- `_id` 搭配下列其中一項：

- `_tag`
- `_createdAt`
- `_lastUpdated`

當您在套件中使用條件式 PUTs 時，會根據現有資源 AWS HealthLake 評估查詢參數，並根據相符結果採取動作。

條件式更新行為

案例	HTTP 狀態	採取的動作
未提供 ID 的資源	201 已建立	一律建立新的資源。
具有新 ID 的資源 (不相符)	201 已建立	使用指定的 ID 建立新的資源。
具有現有 ID 的資源 (單一相符項目)	200 OK	更新相符的資源。
具有現有 ID 的資源 (偵測到衝突)	409 衝突	傳回錯誤。不會進行任何變更。
具有現有 ID 的資源 (ID 不相符)	400 錯誤的請求	傳回錯誤。不會進行任何變更。
多個資源符合條件	412 先決條件失敗	傳回錯誤。不會進行任何變更。

在具有條件式更新的下列範例套件中，只有符合條件時，具有 FHIR ID `_lastUpdated=lt2025-04-20 Patient`的資源才會更新。

```
{
  "resourceType": "Bundle",
  "id": "bundle-batch",
  "meta": {
    "lastUpdated": "2014-08-18T01:43:30Z"
  },
  "type": "batch",
  "entry": [
    {
      "resource": {
        "resourceType": "Patient",
        "id": "476",
```

```
    "meta": {
      "lastUpdated": "2022-06-03T17:53:36.724Z"
    },
    "active": true,
    "name": [
      {
        "use": "official",
        "family": "Jackson",
        "given": [
          "Mateo",
          "James"
        ]
      }
    ],
    "gender": "male",
    "birthDate": "1974-12-25"
  },
  "request": {
    "method": "PUT",
    "url": "Patient?_id=476&_lastUpdated=lt2025-04-20"
  }
},
{
  "resource": {
    "resourceType": "Medication",
    "id": "med0310",
    "contained": [
      {
        "resourceType": "Substance",
        "id": "sub03",
        "code": {
          "coding": [
            {
              "system": "http://snomed.info/sct",
              "code": "55452001",
              "display": "Oxycodone (substance)"
            }
          ]
        }
      }
    ]
  },
  "code": {
    "coding": [
      {
```

```

        "system": "http://snomed.info/sct",
        "code": "430127000",
        "display": "Oral Form Oxycodone (product)"
      }
    ]
  },
  "form": {
    "coding": [
      {
        "system": "http://snomed.info/sct",
        "code": "385055001",
        "display": "Tablet dose form (qualifier value)"
      }
    ]
  },
  "ingredient": [
    {
      "itemReference": {
        "reference": "#sub03"
      },
      "strength": {
        "numerator": {
          "value": 5,
          "system": "http://unitsofmeasure.org",
          "code": "mg"
        },
        "denominator": {
          "value": 1,
          "system": "http://terminology.hl7.org/CodeSystem/v3-
orderableDrugForm",
          "code": "TAB"
        }
      }
    }
  ]
},
"request": {
  "method": "POST",
  "url": "Medication"
}
]
}

```

將 FHIR 資源綁定為單一實體

將 FHIR 資源綁定為單一實體

1. 收集 HealthLake `region` 和 `datastoreId` 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 使用 HealthLake `region` 和 的收集值來建構請求的 URL `datastoreId`。在 Bundle URL 中包含 FHIR 資源類型。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Bundle
```

3. 為請求建構 JSON 內文，指定要分組的 FHIR 資源。下列範例會將 HealthLake 中的兩個 `Patient` 資源分組。基於此程序的目的，請將檔案儲存為 `batch-single.json`。

```
{
  "resourceType": "Bundle",
  "id": "bundle-minimal",
  "language": "en-US",
  "identifier": {
    "system": "urn:oid:1.2.3.4.5",
    "value": "28b95815-76ce-457b-b7ae-a972e527db4f"
  },
  "type": "document",
  "timestamp": "2020-12-11T14:30:00+01:00",
  "entry": [
    {
      "fullUrl": "urn:uuid:f40b07e3-37e8-48c3-bf1c-ae70fe12dabf",
      "resource": {
        "resourceType": "Composition",
        "id": "f40b07e3-37e8-48c3-bf1c-ae70fe12dabf",
        "status": "final",
        "type": {
          "coding": [
            {
              "system": "http://loinc.org",
              "code": "60591-5",
              "display": "Patient summary Document"
            }
          ]
        },
        "date": "2020-12-11T14:30:00+01:00",
        "author": [
```

```

        {
            "reference":
"urn:uuid:45271f7f-63ab-4946-970f-3daaaa0663ff"
        }
    ],
    "title": "Patient Summary as of December 7, 2020 14:30"
}
},
{
    "fullUrl": "urn:uuid:45271f7f-63ab-4946-970f-3daaaa0663ff",
    "resource": {
        "resourceType": "Practitioner",
        "id": "45271f7f-63ab-4946-970f-3daaaa0663ff",

        "active": true,
        "name": [
            {
                "family": "Doe",
                "given": [
                    "John"
                ]
            }
        ]
    }
}
]
}
}

```

- 傳送 請求。FHIR Bundle 文件類型使用具有 [AWS Signature 第 4 版](#) 簽署通訊協定的POST請求。下列程式碼範例使用curl命令列工具進行示範。

SigV4

SigV4 授權

```

curl --request POST \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Bundle' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json' \
  --data @document-type.json

```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\"issuer\": \"https://ehr.example.com\", \"jwks_uri\": \"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\": [\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"], \"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"], \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\", \"permission-v2\"]}"
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

伺服器會傳回回應，顯示因Bundle文件類型請求而建立的兩個Patient資源。

設定套件的驗證層級

綁定 FHIR 資源時，您可以選擇指定 `x-amzn-healthlake-fhir-validation-level` HTTP 標頭來設定資源的驗證層級。此驗證層級將針對 bundle 中的所有建立和更新請求設定。AWS HealthLake 目前支援下列驗證層級：

- `strict`：資源會根據資源的設定檔元素進行驗證，如果沒有設定檔，則為 R4 規格。這是的預設驗證層級 AWS HealthLake。
- `structure-only`：資源會根據 R4 驗證，忽略任何參考的設定檔。
- `minimal`：最少驗證資源，忽略某些 R4 規則。搜尋/分析所需的結構檢查失敗的資源將更新為包含稽核警告。

與最低驗證層級綁定的資源可能會擷取到資料存放區，即使搜尋索引所需的驗證失敗。在此情況下，資源將更新為包含 Healthlake 特定延伸，以記錄上述失敗，而 Bundle 回應中的項目將包含 OperationOutcome 資源，如下所示：

```
{
  "resourceType": "Bundle",
  "type": "batch-response",
  "timestamp": "2025-08-25T22:58:48.846287342Z",
  "entry": [
    {
      "response": {
        "status": "201",
        "location": "Patient/195abc49-ba8e-4c8b-95c2-abc88fef7544/_history/1",
        "etag": "W/\"1\"",
        "lastModified": "2025-08-25T22:58:48.801245445Z",
        "outcome": {
          "resourceType": "OperationOutcome",
          "issue": [
            {
              "severity": "error",
              "code": "processing",
              "details": {
                "text": "FHIR resource in payload failed FHIR
validation rules."
              },
              "diagnostics": "FHIR resource in payload failed FHIR
validation rules."
            }
          ]
        }
      }
    }
  ]
}
```

此外，以下 HTTP 回應標頭會包含在「true」的值中：

```
x-amzn-healthlake-validation-issues : true
```

Note

請注意，如果出現這些錯誤，根據 R4 規格格式不正確擷取的資料可能無法如預期搜尋。

對 Bundle 類型 "message" 的有限支援

HealthLake message 透過內部轉換程序提供 FHIR Bundle 類型的有限支援。此支援適用於訊息套件無法在來源重新格式化的情況，例如從舊版醫院系統擷取 ADT (Admission、Discharge、Transfer) 饋送。

Warning

此功能需要明確的 AWS 帳戶允許清單，且不會強制執行 FHIR R4 訊息語意或參考完整性。使用訊息套件之前，請聯絡 AWS Support 請求啟用您的帳戶。

與標準訊息處理的主要差異

- 訊息套件 (FHIR 規格)：第一個項目必須是參考其他資源 MessageHeader 的。資源缺少個別 request 物件，MessageHeader 事件決定處理動作。
- HealthLake Processing：透過自動將 PUT 操作指派給每個資源項目，將訊息套件轉換為批次套件。資源會獨立處理，不會強制執行訊息語意或參考完整性。

重要限制

- 未強制執行 FHIR R4 訊息特定處理規則
- 資源之間沒有交易完整性
- 未驗證資源間參考
- 需要明確的帳戶允許清單

訊息套件結構範例

```
{
    "resourceType": "Bundle",
    "type": "message",
}
```

```
    "entry": [
      {
        "resource": {
          "resourceType": "MessageHeader",
          "eventCoding": {
            "system": "http://hl7.org/fhir/us/davinci-alerts/CodeSystem/
notification-event",
            "code": "notification-admit"
          },
          "focus": [{"reference": "Encounter/example-id"}]
        }
      },
      {
        "resource": {"resourceType": "Patient", "id": "example-id"}
      },
      {
        "resource": {"resourceType": "Encounter", "id": "example-id"}
      }
    ]
  }
}
```

Note

每個資源都會獨立儲存，就像透過個別 PUT 操作提交一樣。如果需要完整的 FHIR 訊息語意或參考完整性驗證，請在提交之前預先處理訊息套件或實作應用程式層級驗證。

非同步套件交易

AWS HealthLake 支援非同步 Bundle 類型 transaction，可讓您提交最多 500 個資源的交易。當您提交非同步交易時，HealthLake 會將其排入佇列進行處理，並立即傳回輪詢 URL。您可以使用此 URL 來檢查狀態並擷取回應。這遵循 [FHIR 非同步套件模式](#)。

何時使用非同步交易

- 您需要在單一交易中提交超過 100 個資源（同步限制）。
- 您希望避免在等待交易處理完成時封鎖應用程式。
- 您需要處理具有更高輸送量的大量相關資源。

⚠ Important

輪詢結果會在交易完成後 90 天內提供。在此 90 天期間之後，輪詢 URL 不會再傳回結果。設計您的整合，在此視窗中擷取和儲存結果。

ℹ Note

同步Bundle類型transaction會持續支援最多 100 個資源，並且是預設的處理模式。如果您在沒有 `Prefer: respond-async` 標頭的情況下提交transaction具有超過 100 個資源的Bundle類型，HealthLake 會傳回422 Unprocessable Entity錯誤。非同步處理batch不支援類型的套件 - 只能非同步提交 Bundle類型 transaction (最多 500 個操作)。

ℹ Note

PATCH 非同步套件交易不支援操作和條件式 PUTs。

提交非同步交易

若要提交非同步交易，請使用 `Prefer: respond-async` 標頭傳送POST請求至資料存放區端點。套件必須具有類型 `transaction`。非同步套件處理batch不支援類型的套件。

HealthLake 會在提交時對套件進行初始驗證。如果驗證成功，HealthLake 會傳回 HTTP 202 已接受，其中包含輪詢 URL 的 `content-location` 回應標頭。

提交非同步Bundle類型 transaction

1. 將POST請求傳送至 HealthLake 資料存放區端點。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
```

2. 使用套件類型 為請求建構 JSON 內文transaction。基於此程序的目的，請將檔案儲存為 `async-transaction.json`。

```
{  
  "resourceType": "Bundle",
```

```
"type": "transaction",
"entry": [
  {
    "resource": {
      "resourceType": "Patient",
      "active": true,
      "name": [
        {
          "use": "official",
          "family": "Smith",
          "given": ["Jane"]
        }
      ],
      "gender": "female",
      "birthDate": "1990-01-15"
    },
    "request": {
      "method": "POST",
      "url": "Patient"
    }
  },
  {
    "resource": {
      "resourceType": "Observation",
      "status": "final",
      "code": {
        "coding": [
          {
            "system": "http://loinc.org",
            "code": "85354-9",
            "display": "Blood pressure panel"
          }
        ]
      },
      "subject": {
        "reference": "urn:uuid:example-patient-id"
      }
    },
    "request": {
      "method": "POST",
      "url": "Observation"
    }
  }
]
```

```
}

```

3. 使用 `Prefer: respond-async` 標頭傳送請求。FHIR Bundle 交易類型使用具有 [AWS 簽章第 4 版](#) 或 FHIR 授權上的 SMART 的 POST 請求。下列程式碼範例使用 `curl` 命令列工具進行示範。

SigV4

SigV4 授權

```
curl --request POST \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json' \
  --header 'Prefer: respond-async' \
  --data @async-transaction.json

```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\"issuer\":\"https://ehr.example.com\", \"jwks_uri\": \"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\": [\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"], \"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"], \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\", \"permission-v2\"]}"
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱[OAuth 2.0 範圍](#)。

- 成功提交時，伺服器會傳回 HTTP 202 已接受。content-location 回應標頭包含輪詢 URL。回應內文是 OperationOutcome 資源。

```
HTTP/1.1 202 Accepted
content-location: https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
Transaction/transactionId
```

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "information",
      "code": "informational",
      "diagnostics": "Submitted Asynchronous Bundle Transaction",
      "location": [
        "https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
Transaction/transactionId"
      ]
    }
  ]
}
```

輪詢交易狀態

提交非同步交易之後，請使用content-location回應標頭中的輪詢 URL 來檢查交易狀態。將GET請求傳送至輪詢 URL。

Note

對於啟用 FHIR 的 SMART 資料存放區，授權字符必須包含Transaction資源類型的read許可，以輪詢交易狀態。如需 FHIR 範圍的 SMART 詳細資訊，請參閱 [HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART](#)。

將GET請求傳送至輪詢 URL。下列範例使用 curl命令列工具。

SigV4

SigV4 授權

```
curl --request GET \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
Transaction/transactionId' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header 'Accept: application/json'
```

SMART on FHIR

FHIR 授權上的 SMART。授權字符必須包含Transaction資源類型的read許可。

```
curl --request GET \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
Transaction/transactionId' \
  --header 'Authorization: Bearer $SMART_ACCESS_TOKEN' \
  --header 'Accept: application/json'
```

下表說明可能的回應。

輪詢回應代碼

HTTP 狀態	意義	回應主體
202 已接受	交易已排入佇列	OperationOutcome 使用診斷「SUBMITTED」
202 已接受	正在處理交易	OperationOutcome 使用診斷 "IN_PROGRESS"
200 OK	交易已成功完成	Bundle 使用 類型 transaction-response
4xx/5xx	交易失敗	OperationOutcome 含錯誤詳細資訊

下列範例顯示每種回應類型。

已排入佇列的交易 (202)

```
{
  "resourceType": "OperationOutcome",
  "id": "transactionId",
  "issue": [
    {
      "severity": "information",
      "code": "informational",
      "diagnostics": "SUBMITTED"
    }
  ]
}
```

交易處理 (202)

```
{
  "resourceType": "OperationOutcome",
  "id": "transactionId",
  "issue": [
    {
      "severity": "information",
      "code": "informational",
      "diagnostics": "IN_PROGRESS"
    }
  ]
}
```

交易已完成 (200)

```
{
  "resourceType": "Bundle",
  "type": "transaction-response",
  "entry": [
    {
      "response": {
        "status": "201",
        "location": "Patient/example-id/_history/1",
        "etag": "W/\"1\"",
      }
    }
  ]
}
```

```

        "lastModified": "2024-01-15T10:30:00.000Z"
      }
    },
    {
      "response": {
        "status": "201",
        "location": "Observation/example-id/_history/1",
        "etag": "W/\"1\"",
        "lastModified": "2024-01-15T10:30:00.000Z"
      }
    }
  ]
}

```

交易失敗 (4xx/5xx)

```

{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "exception",
      "diagnostics": "Transaction failed: conflict detected on resource Patient/
example-id"
    }
  ]
}

```

處理順序

類型的非同步套件transaction會排入佇列，但不會以嚴格的提交順序處理。HealthLake 會根據可用容量和系統負載來最佳化處理。

Important

請勿依賴依提交順序處理的交易。例如，如果您在上午 10:00 提交交易 A，並在上午 10:01 提交交易 B，交易 B 可能會在交易 A 之前完成。將您的應用程式設計為：

- 處理out-of-order完成。
- 使用輪詢 URL 獨立追蹤每個交易。

- 如果您的使用案例順序很重要，請實作應用程式層級排序。

配額和限流

下列配額和速率限制適用於非同步交易。

非同步交易配額

配額	Value	可調整
每個非同步交易的最大操作數	500	否
每個資料存放區的待處理交易上限	500	是

- 非同步交易會共用 下定義的相同 API 速率限制 [Service Quotas](#)。
- 輪詢交易狀態與 FHIR 資源的讀取 (GET) 操作共用相同的 API 速率限制。
- 如果達到待處理交易限制，後續提交會傳回錯誤，直到現有交易完成為止。

錯誤處理

對於「交易」套件，套件中包含的所有 FHIR 資源都會處理為原子操作。操作中的所有資源都必須成功，否則套件中不會處理任何操作。

錯誤分為兩個類別：HealthLake 同步傳回的提交錯誤，以及處理您透過輪詢擷取的錯誤。

提交錯誤

HealthLake 會在提交時驗證套件，並在交易排入佇列之前同步傳回錯誤。提交錯誤包括無效的 FHIR 資源驗證錯誤、不支援的資源類型、超過 500 個操作限制，以及搭配批次套件使用 `Prefer: respond-async` 標頭。如果已達到資料存放區的待處理交易限制，HealthLake 會傳回 `ThrottlingException`。發生提交錯誤時，交易將不會排入佇列。

處理錯誤

處理錯誤會在交易排入佇列後發生，並透過輪詢 URL 傳回。這些包括交易衝突，其中另一個操作修改了屬於交易一部分的資源，以及處理期間的伺服器錯誤。發生處理錯誤時，不會對交易中的資源進行任何資源變動。輪詢 URL 將傳回 `OperationOutcome` 具有錯誤詳細資訊的。

刪除 FHIR 資源

FHIR delete 互動會從 HealthLake 資料存放區移除現有的 FHIR 資源。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [delete](#) 中的。

刪除 FHIR 資源

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
2. 決定 Resource 要刪除的 FHIR 類型，並收集相關聯的 id 值。如需詳細資訊，請參閱 [資源類型](#)。
3. 使用 HealthLake region 和 的收集值來建構請求的 URL datastoreId。同時包含 FHIR Resource 類型及其相關聯的 id。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
DELETE https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id
```

4. 傳送 請求。FHIR delete 互動使用具有 [AWS 簽章第 4](#) 版或 FHIR 授權上的 SMART 的 DELETE 請求。下列 curl 範例會從 HealthLake 資料存放區移除現有的 FHIR Patient 資源。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request DELETE \  
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id' \  
 \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

伺服器會傳回 204 HTTP 狀態碼，確認已從 HealthLake 資料存放區中移除資源。如果刪除請求失敗，您將收到 400 一系列 HTTP 狀態碼，指出請求失敗的原因。

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{  
  "AuthorizationStrategy": "SMART_ON_FHIR",  
  "FineGrainedAuthorizationEnabled": true,  
}
```

```

    "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
    "Metadata": "{ \"issuer\": \"https://ehr.example.com\", \"jwks_uri\":
\"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint
\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://
ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\":
[\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credentia
l\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/
register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"],
\"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://
ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://
ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],
\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\",
\"permission-v2\"]}"
}

```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

AWS Console

1. 登入 HealthLake 主控台上的 [執行查詢](#) 頁面。
2. 在查詢設定區段下，進行下列選擇。
 - 資料存放區 ID — 選擇資料存放區 ID 以產生查詢字串。
 - 查詢類型 — 選擇 Delete。
 - 資源類型 — 選擇要刪除的 FHIR [資源類型](#)。
 - 資源 ID — 輸入 FHIR 資源 ID。
3. 選擇 Run query (執行查詢)。

根據條件刪除 FHIR 資源

當您不知道特定 FHIR 資源 ID，但具有您要刪除之資源的其他識別資訊時，條件式刪除特別有用。

條件式刪除可讓您根據搜尋條件而非邏輯 FHIR ID 刪除現有資源。當伺服器處理刪除請求時，它會使用資源類型的標準搜尋功能執行搜尋，以解析請求的單一邏輯 ID。

條件式刪除的運作方式

伺服器的動作取決於找到多少相符項目：

1. 沒有相符項目：伺服器嘗試一般刪除並適當回應 (404 找不到不存在的資源，204 找不到已刪除資源的內容)
2. 一個相符項目：伺服器對相符的資源執行一般刪除
3. 多個相符項目：傳回 412 先決條件失敗錯誤，指出用戶端的條件不夠選擇性

回應案例

AWS HealthLake 使用以下回應模式處理條件式刪除操作：

成功的操作

- 當您的搜尋條件成功識別單一作用中資源時，系統會在完成刪除後傳回 204 無內容，就像標準刪除操作一樣。

以 ID 為基礎的條件式刪除

根據 id 和其他參數 (createdAt、tag 或 _lastUpdated) 執行條件式刪除時：

- 204 無內容：資源已刪除
- 找不到 404：資源不存在
- 409 衝突：ID 相符，但其他參數不相符

Non-ID-Based 條件式刪除

id 未提供 或使用 createdAt、tag 或 以外的參數時 _lastUpdated：

- 找不到 404：找不到相符項目

衝突情況

數個案例會產生 412 先決條件失敗回應：

- 多個資源符合您的搜尋條件 (條件不夠具體)

- 搭配使用 ETag 標頭時的版本衝突 If-Match
- 在搜尋和刪除操作之間發生資源更新

成功條件式刪除的範例

下列範例會根據特定條件刪除病患資源：

```
DELETE https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
name=peter&birthdate=2000-01-01&phone=1234567890
```

此請求會刪除病患資源，其中：

- 名稱為「peter」
- 出生日期是 2000 年 1 月 1 日
- 電話號碼為 1234567890

最佳實務

1. 使用特定的搜尋條件來避免多個相符項目，並防止 412 錯誤。
2. 需要處理並行修改時，請考慮使用版本控制的 ETag 標頭。
3. 適當地處理錯誤回應：
 - 對於 404：精簡您的搜尋條件
 - 對於 412：讓條件更具體或解決版本衝突
4. 準備在高並行環境中的時間衝突，其中可能會在搜尋和刪除操作之間修改資源。

冪等和並行

冪等性金鑰

AWS HealthLake 支援 FHIR POST 操作的冪等性金鑰，提供強大的機制，以確保資源建立期間的資料完整性。透過在請求標頭中將唯一的 UUID 作為冪等性金鑰，醫療保健應用程式可以保證每個 FHIR 資源只建立一次，即使在涉及網路不穩定或自動重試的情況下也是如此。

此功能對於醫療保健系統特別重要，因為重複的醫療記錄可能會造成嚴重的後果。當收到具有與先前請求相同冪等性金鑰的請求時，HealthLake 會傳回原始資源，而不是建立重複的資源。例如，這可能會

在重試迴圈期間發生，或是因為備援請求管道。使用等冪性金鑰可讓 HealthLake 維持資料一致性，同時為處理間歇性連線問題的用戶端應用程式提供無縫體驗。

實作

```
POST /<baseURL>/Patient
x-amz-fhir-idempotency-key: 123e4567-e89b-12d3-a456-426614174000
{
  "resourceType": "Patient",
  "name": [...]
}
```

回應案例

第一個請求 (已建立 201)

- 已成功建立新資源
- 回應包含資源 ID

重複請求 (409 衝突)

- 偵測到相同的等冪性金鑰
- 傳回的原始資源
- 未建立新的資源

無效請求 (400 個無效的請求)

- 格式錯誤的 UUID
- 缺少必要欄位

最佳實務

- 為每個新資源建立產生唯一的 UUID
- 儲存等冪索引鍵以重試邏輯
- 使用一致的金鑰格式：建議使用 UUID v4
- 在處理資源建立的用戶端應用程式中實作

Note

對於需要嚴格資料準確性和防止重複醫療記錄的醫療保健系統來說，此功能特別重要。

AWS HealthLake 中的 ETag

AWS HealthLake 使用 ETags 在 FHIR 資源中實現樂觀並行控制，提供可靠的機制來管理並行修改和維護資料一致性。ETag 是代表資源特定版本的唯一識別符，可透過 HTTP 標頭做為版本控制系統。讀取或修改資源時，應用程式可以使用 ETags 來防止意外覆寫和確保資料完整性，尤其是在具有潛在並行更新的情況下。

實作範例

```
// Initial Read
GET /fhir/Patient/123
Response:
ETag: W/"1"

// Update with If-Match
PUT /fhir/Patient/123
If-Match: W/"1"
{resource content}

// Create with If-None-Match
PUT /fhir/Patient/123
If-None-Match: *
{resource content}
// Succeeds only if resource doesn't exist
// Fails with 412 if resource exists
```

回應案例

操作成功 (200 OK 或 204 No Content)

- ETag 符合目前版本
- 操作會如預期進行

版本衝突 (412 先決條件失敗)

- ETag 與目前版本不相符
- 拒絕更新以防止資料遺失

最佳實務

- 在所有更新和刪除操作中包含 ETags
- 實作處理版本衝突的重試邏輯
- 使用 If-None-Match : * 表示 create-if-not-exists 案例
- 修改前一律驗證 ETag 新鮮度

此並行控制系統對於維護醫療保健資料的完整性至關重要，尤其是在具有多個使用者或系統存取和修改相同資源的環境中。

在中搜尋 FHIR 資源 AWS HealthLake

使用 FHIR [search](#) 互動，根據一些篩選條件在 HealthLake 資料存放區中搜尋一組 FHIR 資源。您可以使用 GET 或 POST 請求來執行 search 互動。對於涉及個人身分識別資訊 (PII) 或受保護醫療資訊 (PHI) 的搜尋，建議使用 POST 請求，因為 PII 和 PHI 會新增為請求內文的一部分，並在傳輸中加密。

Note

本章所述的 FHIR search 互動是根據 HL7 FHIR R4 標準的醫療保健資料交換而建置。由於它是 HL7 FHIR 服務的表示，因此不會透過 AWS CLI 和 AWS SDKs 提供。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [search](#) 中的。

HealthLake 支援 FHIR R4 搜尋參數的子集。如需詳細資訊，請參閱 [HealthLake 的 FHIR R4 搜尋參數](#)。

主題

- [使用 GET 搜尋 FHIR 資源](#)
- [使用 POST 搜尋 FHIR 資源](#)
- [FHIR 搜尋一致性層級](#)

使用 GET 搜尋 FHIR 資源

您可以使用 GET 請求來搜尋 HealthLake 資料存放區。使用時 GET，HealthLake 支援提供搜尋參數做為 URL 的一部分，但不支援做為請求內文的一部分。如需詳細資訊，請參閱 [HealthLake 的 FHIR R4 搜尋參數](#)。

Important (重要)

對於涉及個人身分識別資訊 (PII) 或受保護醫療資訊 (PHI) 的搜尋，安全最佳實務會使用 POST 請求，因為 PII 和 PHI 會新增為請求內文的一部分，並在傳輸中加密。如需詳細資訊，請參閱 [使用 POST 搜尋 FHIR 資源](#)。

以下程序後面接著使用 GET 搜尋 HealthLake 資料存放區的範例。

使用 搜尋 HealthLake 資料存放區 GET

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 決定要搜尋的 FHIR 資源類型，並收集相關聯的 id 值。如需詳細資訊，請參閱[資源類型](#)。
3. 使用 HealthLake region 和 的收集值來建構請求的 URLdatastoreId。同時包含 FHIR Resource 類型和支援的[搜尋參數](#)。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource{?  
[parameters]}&_format=[mime-type]}}
```

4. 使用 [AWS Signature 第 4 版](#) 或 FHIR 上的 SMART 授權傳送 GET 請求。下列 curl 範例會傳回 HealthLake 資料存放區中的 Patient 資源總數。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request GET \  
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?  
_total=accurate' \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{  
  "AuthorizationStrategy": "SMART_ON_FHIR",  
  "FineGrainedAuthorizationEnabled": true,  
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-  
lambda-name",  
  "Metadata": "{\n\"issuer\": \"https://ehr.example.com\", \n\"jwks_uri\":  
\n\"https://ehr.example.com/.well-known/jwks.json\", \n\"authorization_endpoint  
\n\": \"https://ehr.example.com/auth/authorize\", \n\"token_endpoint\": \"https://  
ehr.token.com/auth/token\", \n\"token_endpoint_auth_methods_supported\":  
[\"client_secret_basic\", \"foo\"], \n\"grant_types_supported\": [\"client_credential  
\n\", \"foo\"], \n\"registration_endpoint\": \"https://ehr.example.com/auth/  
register\", \n\"scopes_supported\": [\"openid\", \"profile\", \"launch\"],  
\n\"response_types_supported\": [\"code\"], \n\"management_endpoint\": \"https://
```

```
ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://  
ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://  
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],  
\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\",  
\"permission-v2\"]}]\"  
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱 [OAuth 2.0 範圍](#)。

AWS Console

Note

HealthLake 主控台僅支援 SigV4 授權。透過 AWS CLI 和 AWS SDKs 支援 FHIR 上的 SMART 授權。

1. 登入 HealthLake 主控台上的 [執行查詢](#) 頁面。
2. 在查詢設定區段下，進行下列選擇。
 - 資料存放區 ID — 選擇資料存放區 ID 以產生查詢字串。
 - 查詢類型 — 選擇 Search with GET。
 - 資源類型 — 選擇要搜尋的 FHIR [資源類型](#)。
 - 搜尋參數 — 選取 [搜尋參數](#) 或搜尋參數組合，將您的查詢集中在特定記錄上。
3. 選擇 Run query (執行查詢)。

範例：使用 GET 搜尋

下列索引標籤提供使用 搜尋特定 FHIR 資源類型的範例 GET。這些範例示範如何在請求 URLs 中指定搜尋參數。

Note

HealthLake 主控台僅支援 SigV4 授權。透過 AWS CLI 和 AWS SDKs 支援 FHIR 上的 SMART 授權。

HealthLake 支援 FHIR R4 搜尋參數的子集。如需詳細資訊，請參閱 [搜尋參數](#)。

Patient (age)

雖然年齡不是 FHIR 中定義的資源類型，但會擷取為 [Patient](#) 資源類型中的元素。使用下列範例，使用 [birthDate](#) 元素和 [搜尋比較器](#) 對 [Patient](#) 資源類型提出 GET 以 為基礎的搜尋請求，以搜尋 1997 年出生的個人。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
birthdate=eq1997
```

Condition

使用下列範例對 [Condition](#) 資源類型提出 GET 請求。搜尋會在 HealthLake 資料存放區中尋找包含 SNOMED 醫療碼 的條件 72892002，其會轉譯為 Normal pregnancy。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Condition?
code=72892002
```

DocumentationReference

下列範例示範如何針對具有鏈球菌診斷的 Patient() 和也獲開立 amoxicillin 的人員，在 [DocumentReference](#) 資源類型上建立 GET 請求。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/
DocumentReference?_lastUpdated=le2021-12-19&infer-icd10cm-entity-text-concept-
score;=streptococcal|0.6&infer-rxnorm-entity-text-concept-score=Amoxicillin|0.8
```

Location

使用下列範例對 [Location](#) 資源類型提出 GET 請求。下列搜尋會尋找 HealthLake 資料存放區中包含城市名稱波士頓做為地址一部分的位置。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Location?
address=boston
```

Observation

使用下列範例，對 [Observation](#) 資源類型提出 GET 以 為基礎的搜尋請求。此搜尋使用 value-concept [搜尋參數](#) 來尋找醫療代碼 266919005，其會轉譯為 Never smoker。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Observation?
value-concept=266919005
```

使用 POST 搜尋 FHIR 資源

您可以使用 FHIR [search](#) 互動與 POST 請求來搜尋 HealthLake 資料存放區。使用時 POST，HealthLake 支援 URL 或請求內文中的搜尋參數，但您無法在單一請求中使用兩者。

Important (重要)

對於涉及個人身分識別資訊 (PII) 或受保護醫療資訊 (PHI) 的搜尋，安全最佳實務會使用 POST 請求，因為 PII 和 PHI 會新增為請求內文的一部分，並在傳輸中加密。

以下程序後面是使用 FHIR R4 與 [search](#) 互動 POST 來搜尋 HealthLake 資料存放區的範例。這些範例示範如何在 JSON 請求內文中指定搜尋參數。

使用 搜尋 HealthLake 資料存放區 POST

1. 收集 HealthLake `region` 和 `datastoreId` 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
2. 決定要搜尋的 FHIR 資源類型，並收集相關聯的 `id` 值。如需詳細資訊，請參閱 [資源類型](#)。
3. 使用 HealthLake `region` 和 的收集值來建構請求的 URL `datastoreId`。同時包含 FHIR `Resource` 類型和 `_search` 互動。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/  
_search
```

4. 為請求建構 JSON 內文，指定要搜尋的 FHIR 資料。基於此程序的目的，您將搜尋 `Observation` 資源，以探索從未抽過煙的患者。若要指定醫療代碼狀態 `Never smoker`，請在 JSON 請求內文 `value-concept=266919005` 中設定。儲存檔案為 `search-observation.json`。

```
value-concept=266919005
```

5. 傳送 請求。FHIR `search` 互動使用 GET 請求搭配 [AWS Signature 第 4 版](#) 或 FHIR 上的 SMART 授權。

Note

在 POST 請求內文中使用搜尋參數提出請求時，請使用 `Content-Type: application/x-www-form-urlencoded` 做為 標頭的一部分。

下列curl範例會針對 Observation 資源類型提出 POST 型搜尋請求。請求使用 [value-concept](#) 搜尋參數來尋找266919005指出值的醫療程式碼Never smoker。若要檢視整個範例，請捲動至複製按鈕。

SigV4

SigV4 授權

```
curl --request POST \
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Observation/_search' \
  --aws-sigv4 'aws:amz:region:healthlake' \
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \
  --header "Content-Type: application/x-www-form-urlencoded"
  --header "Accept: application/json"
  --data @search-observation.json
```

SMART on FHIR

[IdentityProviderConfiguration](#) 資料類型的 FHIR 上的 SMART 授權範例。

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-lambda-name",
  "Metadata": "{\"issuer\": \"https://ehr.example.com\", \"jwks_uri\": \"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\": [\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"], \"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"], \"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\", \"permission-v2\"]}"
```

```
}
```

發起人可以在授權 lambda 中指派許可。如需詳細資訊，請參閱[OAuth 2.0 範圍](#)。

範例：使用 POST 搜尋

下列索引標籤提供使用 搜尋特定 FHIR 資源類型的範例POST。這些範例示範如何在 URLs中指定請求。

Note

HealthLake 主控台僅支援 SigV4 授權。AWS CLI 和 AWS SDKs 支援 FHIR 上的 SMART 授權。

HealthLake 支援 FHIR R4 搜尋參數的子集。如需詳細資訊，請參閱[搜尋參數](#)。

Patient (age)

雖然年齡不是 FHIR 中定義的資源類型，但會擷取為[Patient](#)資源類型中的 元素。使用下列範例，對 Patient 資源類型提出POST以 為基礎的搜尋請求。下列搜尋範例使用eq[搜尋比較器](#)來搜尋 1997 年出生的個人。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/_search
```

若要在搜尋中指定年份 1997，請將下列元素新增至請求內文。

```
birthdate=eq1997
```

Condition

使用下列項目對Condition資源類型提出POST請求。此搜尋會尋找 HealthLake 資料存放區中包含醫療碼 的位置72892002。

您必須指定請求 URL 和請求內文。以下是請求 URL 的範例。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Condition/_search
```

若要指定您要搜尋的醫療代碼，請將下列 JSON 元素新增至請求內文。

```
code=72892002
```

DocumentReference

若要查看 HealthLake 在 DocumentReference 資源類型上提出 POST 請求時的整合自然語言處理 (NLP) 結果，請依下列方式格式化請求。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/  
DocumentReference/_search
```

若要指定要參考的 DocumentReference 搜尋參數，請參閱 [搜尋參數類型](#)。下列查詢字串使用多個搜尋參數來搜尋用於產生整合 NLP 結果的 Amazon Comprehend Medical API 操作。

```
_lastUpdated=1e2021-12-19&infer-icd10cm-entity-text-concept-score;=streptococcal|  
0.6&infer-rxnorm-entity-text-concept-score=Amoxicillin|0.8
```

Location

使用下列範例對 Location 資源類型提出 POST 請求。搜尋會尋找 HealthLake 資料存放區中包含城市名稱 Boston 做為地址一部分的位置。

您必須指定請求 URL 和請求內文。以下是請求 URL 的範例。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Location/  
_search
```

若要在搜尋 Boston 中指定，請將下列元素新增至請求內文：

```
address=Boston
```

Observation

使用下列範例，對 Observation 資源類型提出 POST 以 為基礎的搜尋請求。搜尋會使用 value-concept 搜尋參數來尋找 266919005 指出的醫療程式碼 Never smoker。

```
POST https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Observation/  
_search
```

若要指定狀態，請在 JSON 內文 `value-concept=266919005` 中設定 `Never smoker`。

```
value-concept=266919005
```

FHIR 搜尋一致性層級

AWS HealthLake 的搜尋索引在 SEARCH 操作 POST 的 GET 和最終一致性模型上運作。使用最終一致性，如果資源有待定的搜尋索引更新，搜尋結果會排除資源的 N-1 版本，直到索引更新完成為止。

AWS HealthLake 現在包含選取一致性模型對更新資源的行為能力。開發人員可以包含「最終一致性」、上述預設行為或「強一致性」。強一致性將允許具有待定搜尋索引更新之資源的 N-1 版本資源包含在搜尋結果中。這可用於即使搜尋索引更新尚未完成，結果中仍需要所有資源的使用案例。客戶可以使用 `x-amz-fhir-history-consistency-level` 請求標頭控制此行為。

一致性層級

高度的一致性

`x-amz-fhir-history-consistency-level: strong` 設定為傳回所有相符的記錄，包括有待定搜尋索引更新的記錄。當您需要在更新後立即搜尋資源時，請使用此選項。

最終一致性

`x-amz-fhir-history-consistency-level: eventual` 設定為僅傳回已完成搜尋索引更新的記錄。如果未指定一致性層級，則此為預設行為。

使用範例

1. 更新資源時：

```
POST <baseURL>/Patient
Content-Type: application/fhir+json
x-amz-fhir-history-consistency-level: strong

{
  "resourceType": "Patient",
  "id": "123",
  "meta": {
    "profile": ["http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"]
  }
}
```

```
  },
  "identifier": [
    {
      "system": "http://example.org/identifiers",
      "value": "123"
    }
  ],
  "active": true,
  "name": [
    {
      "family": "Smith",
      "given": ["John"]
    }
  ],
  "gender": "male",
  "birthDate": "1970-01-01"
}
```

2. 後續搜尋：

```
GET <baseURL>/Patient?_id=123
```

最佳實務

- 當您需要立即搜尋最近更新的資源時，請使用強大的一致性
- 針對即時可見性不重要的一般查詢使用最終一致性
- 考慮立即可見性和潛在效能影響之間的權衡

Note

一致性層級設定會影響已更新資源在搜尋結果中出現的速度，但不會影響資源的實際儲存。將選用 `x-amz-fhir-history-consistency-level` 標頭設定為 `'strong'` 會使每個資源的寫入容量耗用加倍。

此功能僅適用於已啟用版本歷史記錄的資料存放區（在 2024 年 10 月 25 日之後建立的所有資料存放區預設為啟用）。

使用 匯出 FHIR 資料 AWS HealthLake

使用原生 AWS HealthLake 動作來啟動、描述和列出 FHIR 匯出任務。您可以將匯出任務排入佇列。非同步匯出任務會以 FIFO（先進先出）方式處理。您可以像開始匯出任務一樣，將任務排入佇列。如果正在進行，它只會排入佇列。您可以在匯出任務進行時建立、讀取、更新或刪除 FHIR 資源。

Note

您也可以使用 FHIR R4 `$export` 操作從 HealthLake 資料存放區匯出 FHIR 資料。如需詳細資訊，請參閱 [使用 FHIR 匯出 HealthLake 資料 \\$export](#)。

主題

- [啟動 FHIR 匯出任務](#)
- [取得 FHIR 匯出任務屬性](#)
- [列出 FHIR 匯出任務](#)

啟動 FHIR 匯出任務

使用從 HealthLake 資料存放區 `StartFHIRExportJob` 啟動 FHIR 匯出任務。下列功能表提供 AWS 管理主控台和 AWS SDKs 的 AWS CLI 和程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [StartFHIRExportJob](#)。

注意

HealthLake 支援用於醫療保健資料交換的 [FHIR R4 規格](#)。因此，所有運作狀態資料都會以 FHIR R4 格式匯出。

啟動 FHIR 匯出任務

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

啟動 FHIR 匯出任務

下列 `start-fhir-export-job` 範例示範如何使用 AWS HealthLake 啟動 FHIR 匯出任務。

```
aws healthlake start-fhir-export-job \  
  --output-data-config '{"S3Configuration": {"S3Uri": "s3://(Bucket Name)/  
(Prefix Name)/", "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-  
b56c-4216-a250-f4c43ef46e83"}}' \  
  --datastore-id (Data store ID) \  
  --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)
```

輸出：

```
{  
  "DatastoreId": "(Data store ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "9b9a51943afaedd0a8c0c26c49135a31"  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [StartFHIRExportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.  
    """  
    health_lake_client = boto3.client("healthlake")
```

```
        return cls(health_lake_client)

    def start_fhir_export_job(
        self,
        job_name: str,
        datastore_id: str,
        output_s3_uri: str,
        kms_key_id: str,
        data_access_role_arn: str,
    ) -> dict[str, str]:
        """
        Starts a HealthLake export job.
        :param job_name: The export job name.
        :param datastore_id: The data store ID.
        :param output_s3_uri: The output S3 URI.
        :param kms_key_id: The KMS key ID associated with the output S3 bucket.
        :param data_access_role_arn: The data access role ARN.
        :return: The export job.
        """
        try:
            response = self.health_lake_client.start_fhir_export_job(
                OutputDataConfig={
                    "S3Configuration": {"S3Uri": output_s3_uri, "KmsKeyId":
kms_key_id}
                },
                DataAccessRoleArn=data_access_role_arn,
                DatastoreId=datastore_id,
                JobName=job_name,
            )

            return response
        except ClientError as err:
            logger.exception(
                "Couldn't start export job. Here's why %s",
                err.response["Error"]["Message"],
            )
            raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [StartFHIRExportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  " iv_job_name = 'MyExportJob'
  " iv_output_s3_uri = 's3://my-bucket/export/output/'
  " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
  " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
  oo_result = lo_hll->startfhirexportjob(
    iv_jobname = iv_job_name
    io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
      io_s3configuration = NEW /aws1/cl_hlls3configuration(
        iv_s3uri = iv_output_s3_uri
        iv_kmskeyid = iv_kms_key_id
      )
    )
    iv_dataaccessrolearn = iv_data_access_role_arn
    iv_datastoreid = iv_datastore_id
  ).
  DATA(lv_job_id) = oo_result->get_jobid( ).
  MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_validation_ex.
```

```
CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
  lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_throttling_ex.
CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
  lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.
  RAISE EXCEPTION lo_access_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [StartFHIRExportJob](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

1. 登入 HealthLake 主控台上的 [資料存放](#) 區頁面。
2. 選擇資料存放區。
3. 選擇 Export (匯出)。

匯出頁面隨即開啟。

4. 在輸出資料區段下，輸入下列資訊：
 - Amazon S3 中的輸出資料位置
 - 輸出排列
5. 在存取許可區段下，選擇使用現有的 IAM 服務角色，然後從角色名稱功能表中選取角色，或選擇建立 IAM 角色。
6. 選擇開始匯出。

Note

在匯出期間，選擇頁面頂端橫幅上的複製任務 ID。您可以使用 [JobID](#) 來請求使用匯出任務屬性 AWS CLI。如需詳細資訊，請參閱[取得 FHIR 匯出任務屬性](#)。

取得 FHIR 匯出任務屬性

使用從 HealthLake 資料存放區 `DescribeFHIRExportJob` 取得匯出任務屬性。下列功能表提供 AWS 管理主控台 和 AWS SDKs 的 AWS CLI 和 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [DescribeFHIRExportJob](#)。

注意

HealthLake 支援用於醫療保健資料交換的 [FHIR R4 規格](#)。因此，所有運作狀態資料都會以 FHIR R4 格式匯出。

描述 FHIR 匯出任務

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

描述 FHIR 匯出任務

下列 `describe-fhir-export-job` 範例顯示如何在 AWS HealthLake 中尋找 FHIR 匯出任務的屬性。

```
aws healthlake describe-fhir-export-job \  
  --datastore-id (Data store ID) \  
  --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

輸出：

```
{
  "ExportJobProperties": {
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",
    "JobStatus": "IN_PROGRESS",
    "JobId": "9009813e9d69ba7cf79bcb3468780f16",
    "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
    "EndTime": "2024-11-20T11:34:01.636000-05:00",
    "OutputDataConfig": {
      "S3Configuration": {
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-
b56c-4216-a250-f4c43ef46e83"
      }
    },
    "DatastoreId": "(Data store ID)"
  }
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRExportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def describe_fhir_export_job(
    self, datastore_id: str, job_id: str
) -> dict[str, any]:
    """
```

```
Describes a HealthLake export job.
:param datastore_id: The data store ID.
:param job_id: The export job ID.
:return: The export job description.
"""
try:
    response = self.health_lake_client.describe_fhir_export_job(
        DatastoreId=datastore_id, JobId=job_id
    )
    return response["ExportJobProperties"]
except ClientError as err:
    logger.exception(
        "Couldn't describe export job with ID %s. Here's why %s",
        job_id,
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRExportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
```

```

" iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
oo_result = lo_hll->describehireexportjob(
  iv_datastoreid = iv_datastore_id
  iv_jobid = iv_job_id
).
DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).
IF lo_export_job_properties IS BOUND.
  DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).
  MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.
ENDIF.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
lv_error = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_validation_ex.
ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRExportJob](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

Note

FHIR 匯出任務資訊不適用於 HealthLake 主控台。反之，使用 AWS CLI 搭配 `DescribeFHIRExportJob` 來請求匯出任務屬性，例如 [JobStatus](#)。如需詳細資訊，請參閱此頁面 AWS CLI 的範例。

列出 FHIR 匯出任務

使用 `ListFHIRExportJobs` 列出 HealthLake 資料存放區的 FHIR 匯出任務。下列功能表提供 AWS 管理主控台 和 SDKs 的 AWS CLI 和 AWS 程式碼範例程序。如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [ListFHIRExportJobs](#)。

注意

HealthLake 支援用於醫療保健資料交換的 [FHIR R4 規格](#)。因此，所有運作狀態資料都會以 FHIR R4 格式匯出。

列出 FHIR 匯出任務

根據您的存取偏好設定選擇選單 AWS HealthLake。

AWS CLI 和 SDKs

CLI

AWS CLI

列出所有 FHIR 匯出任務

下列 `list-fhir-export-jobs` 範例示範如何使用命令來檢視與帳戶相關聯的匯出任務清單。

```
aws healthlake list-fhir-export-jobs \  
  --datastore-id (Data store ID) \  
  --submitted-before (DATE like 2024-10-13T19:00:00Z) \  
  --submitted-after (DATE like 2020-10-13T19:00:00Z) \  
  --job-name "FHIR-EXPORT" \  
  --job-status SUBMITTED \  
  --max-results (Integer between 1 and 500)
```

輸出：

```
{  
  "ExportJobPropertiesList": [  
    {  
      "ExportJobProperties": {
```

```

        "OutputDataConfig": {
            "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
            "S3Configuration": {
                "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                "KmsKeyId": "(KmsKey Id)"
            }
        },
        "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role
Name)",
        "JobStatus": "COMPLETED",
        "JobId": "c145fbb27b192af392f8ce6e7838e34f",
        "JobName": "FHIR-EXPORT",
        "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
        "EndTime": "2024-11-20T11:34:01.636000-05:00",
        "DatastoreId": "(Data store ID)"
    }
}
]
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRExportJobs](#)。

Python

適用於 Python 的 SDK (Boto3)

```

@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_fhir_export_jobs(
    self,
    datastore_id: str,

```

```
    job_name: str = None,
    job_status: str = None,
    submitted_before: datetime = None,
    submitted_after: datetime = None,
) -> list[dict[str, any]]:
    """
    Lists HealthLake export jobs satisfying the conditions.
    :param datastore_id: The data store ID.
    :param job_name: The export job name.
    :param job_status: The export job status.
    :param submitted_before: The export job submitted before the specified
date.
    :param submitted_after: The export job submitted after the specified
date.
    :return: A list of export jobs.
    """
    try:
        parameters = {"DatastoreId": datastore_id}
        if job_name is not None:
            parameters["JobName"] = job_name
        if job_status is not None:
            parameters["JobStatus"] = job_status
        if submitted_before is not None:
            parameters["SubmittedBefore"] = submitted_before
        if submitted_after is not None:
            parameters["SubmittedAfter"] = submitted_after
        next_token = None
        jobs = []
        # Loop through paginated results.
        while True:
            if next_token is not None:
                parameters["NextToken"] = next_token
            response =
self.health_lake_client.list_fhir_export_jobs(**parameters)
            jobs.extend(response["ExportJobPropertiesList"])
            if "NextToken" in response:
                next_token = response["NextToken"]
            else:
                break
        return jobs
    except ClientError as err:
        logger.exception(
            "Couldn't list export jobs. Here's why %s",
            err.response["Error"]["Message"],
```

```
)
raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRExportJobs](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  IF iv_submitted_after IS NOT INITIAL.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
      iv_submittedafter = iv_submitted_after
    ).
  ELSE.
    oo_result = lo_hll->listfhirexportjobs(
      iv_datastoreid = iv_datastore_id
    ).
  ENDIF.
  DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
  DATA(lv_job_count) = lines( lt_export_jobs ).
  MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
```

```
DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIRExportJobs](#)。

可用性範例

找不到所需的內容嗎？使用此頁面右側的提供意見回饋連結來請求程式碼範例。

AWS 主控台

Note

FHIR 匯出任務資訊不適用於 HealthLake 主控台。反之，使用 AWS CLI 搭配 `ListFHIRExportJobs` 列出所有 FHIR 匯出任務。如需詳細資訊，請參閱此頁面 AWS CLI 的範例。

HealthLake AWS SDKs的程式碼範例

下列程式碼範例示範如何使用 HealthLake 搭配 AWS 軟體開發套件 (SDK)。

Actions 是大型程式的程式碼摘錄，必須在內容中執行。雖然動作會告訴您如何呼叫個別服務函數，但您可以在其相關情境中查看內容中的動作。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

程式碼範例

- [HealthLake AWS SDKs的基本範例](#)
 - [HealthLake AWS SDKs的動作](#)
 - [CreateFHIRDatastore 搭配 AWS SDK 或 CLI 使用](#)
 - [DeleteFHIRDatastore 搭配 AWS SDK 或 CLI 使用](#)
 - [DescribeFHIRDatastore 搭配 AWS SDK 或 CLI 使用](#)
 - [DescribeFHIRExportJob 搭配 AWS SDK 或 CLI 使用](#)
 - [DescribeFHIRImportJob 搭配 AWS SDK 或 CLI 使用](#)
 - [ListFHIRDatastores 搭配 AWS SDK 或 CLI 使用](#)
 - [ListFHIRExportJobs 搭配 AWS SDK 或 CLI 使用](#)
 - [ListFHIRImportJobs 搭配 AWS SDK 或 CLI 使用](#)
 - [ListTagsForResource 搭配 AWS SDK 或 CLI 使用](#)
 - [StartFHIRExportJob 搭配 AWS SDK 或 CLI 使用](#)
 - [StartFHIRImportJob 搭配 AWS SDK 或 CLI 使用](#)
 - [TagResource 搭配 AWS SDK 或 CLI 使用](#)
 - [UntagResource 搭配 AWS SDK 或 CLI 使用](#)

HealthLake AWS SDKs的基本範例

下列程式碼範例示範如何 AWS HealthLake 搭配 AWS SDKs 使用的基本概念。

範例

- [HealthLake AWS SDKs的動作](#)

基本概念


```
--preload-data-config PreloadDataType="SYNTHEA" \
--sse-configuration '{ "KmsEncryptionConfig": { "CmkType":
"CUSTOMER_MANAGED_KMS_KEY", "KmsKeyId": "arn:aws:kms:us-east-1:your-account-
id:key/your-key-id" } }' \
--identity-provider-configuration file://
identity_provider_configuration.json
```

identity_provider_configuration.json 的內容：

```
{
  "AuthorizationStrategy": "SMART_ON_FHIR_V1",
  "FineGrainedAuthorizationEnabled": true,
  "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-id:function:your-
lambda-name",
  "Metadata": "{\"issuer\": \"https://ehr.example.com\", \"jwks_uri\":
\"https://ehr.example.com/.well-known/jwks.json\", \"authorization_endpoint
\": \"https://ehr.example.com/auth/authorize\", \"token_endpoint\": \"https://
ehr.token.com/auth/token\", \"token_endpoint_auth_methods_supported\":
[\"client_secret_basic\", \"foo\"], \"grant_types_supported\": [\"client_credential
\", \"foo\"], \"registration_endpoint\": \"https://ehr.example.com/auth/
register\", \"scopes_supported\": [\"openid\", \"profile\", \"launch\"],
\"response_types_supported\": [\"code\"], \"management_endpoint\": \"https://
ehr.example.com/user/manage\", \"introspection_endpoint\": \"https://
ehr.example.com/user/introspect\", \"revocation_endpoint\": \"https://
ehr.example.com/user/revoke\", \"code_challenge_methods_supported\": [\"S256\"],
\"capabilities\": [\"launch-ehr\", \"sso-openid-connect\", \"client-public\"]}"
}
```

輸出：

```
{
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
(Data store ID)/r4/",
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/
(Data store ID)",
  "DatastoreStatus": "CREATING",
  "DatastoreId": "(Data store ID)"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [CreateFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def create_fhir_datastore(
    self,
    datastore_name: str,
    sse_configuration: dict[str, any] = None,
    identity_provider_configuration: dict[str, any] = None,
) -> dict[str, str]:
    """
    Creates a new HealthLake data store.
    When creating a SMART on FHIR data store, the following parameters are
    required:
    - sse_configuration: The server-side encryption configuration for a SMART
    on FHIR-enabled data store.
    - identity_provider_configuration: The identity provider configuration
    for a SMART on FHIR-enabled data store.

    :param datastore_name: The name of the data store.
    :param sse_configuration: The server-side encryption configuration for a
    SMART on FHIR-enabled data store.
    :param identity_provider_configuration: The identity provider
    configuration for a SMART on FHIR-enabled data store.
    :return: A dictionary containing the data store information.
    """
    try:
        parameters = {"DatastoreName": datastore_name,
"DatastoreTypeVersion": "R4"}
        if (
            sse_configuration is not None
```

```
        and identity_provider_configuration is not None
    ):
        # Creating a SMART on FHIR-enabled data store
        parameters["SseConfiguration"] = sse_configuration
        parameters[
            "IdentityProviderConfiguration"
        ] = identity_provider_configuration

    response =
self.health_lake_client.create_fhir_datastore(**parameters)
    return response
except ClientError as err:
    logger.exception(
        "Couldn't create data store %s. Here's why %s",
        datastore_name,
        err.response["Error"]["Message"],
    )
    raise
```

下列程式碼示範啟用 FHIR 的 HealthLake 資料存放區上的 SMART 參數範例。

```
sse_configuration = {
    "KmsEncryptionConfig": {"CmkType": "AWS_OWNED_KMS_KEY"}
}
# TODO: Update the metadata to match your environment.
metadata = {
    "issuer": "https://ehr.example.com",
    "jwks_uri": "https://ehr.example.com/.well-known/jwks.json",
    "authorization_endpoint": "https://ehr.example.com/auth/
authorize",
    "token_endpoint": "https://ehr.token.com/auth/token",
    "token_endpoint_auth_methods_supported": [
        "client_secret_basic",
        "foo",
    ],
    "grant_types_supported": ["client_credential", "foo"],
    "registration_endpoint": "https://ehr.example.com/auth/register",
    "scopes_supported": ["openId", "profile", "launch"],
    "response_types_supported": ["code"],
    "management_endpoint": "https://ehr.example.com/user/manage",
```

```
introspection_endpoint": "https://ehr.example.com/user/
introspect",
    "revocation_endpoint": "https://ehr.example.com/user/revoke",
    "code_challenge_methods_supported": ["S256"],
    "capabilities": [
        "launch-ehr",
        "sso-openid-connect",
        "client-public",
    ],
}
# TODO: Update the IdpLambdaArn.
identity_provider_configuration = {
    "AuthorizationStrategy": "SMART_ON_FHIR_V1",
    "FineGrainedAuthorizationEnabled": True,
    "IdpLambdaArn": "arn:aws:lambda:your-region:your-account-
id:function:your-lambda-name",
    "Metadata": json.dumps(metadata),
}
data_store = self.create_fhir_datastore(
    datastore_name, sse_configuration,
    identity_provider_configuration
)
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [CreateFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_name = 'MyHealthLakeDataStore'  
  oo_result = lo_hll->createfhirdatastore(  
    iv_datastorename = iv_datastore_name  
    iv_datastoretypeversion = 'R4'  
  ).  
  MESSAGE 'Data store created successfully.' TYPE 'I'.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
  CATCH /aws1/cx_hllinternalserverex INTO DATA(lo_internal_ex).  
    lv_error = |Internal server error: { lo_internal_ex->av_err_code }-  
{ lo_internal_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_internal_ex.  
  CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).  
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-  
{ lo_throttling_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_throttling_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [CreateFHIRDatastore](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

DeleteFHIRDatastore 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 DeleteFHIRDatastore。

CLI

AWS CLI

刪除 FHIR 資料存放區

下列delete-fhir-datastore範例示範如何刪除資料存放區及其在 AWS HealthLake 中的所有內容。

```
aws healthlake delete-fhir-datastore \  
  --datastore-id (Data store ID)
```

輸出：

```
{  
  "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/  
(Data store ID)/r4/",  
  "DatastoreArn": "arn:aws:healthlake:us-east-1:(AWS Account ID):datastore/  
(Data store ID)",  
  "DatastoreStatus": "DELETING",  
  "DatastoreId": "(Data store ID)"  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DeleteFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.  
    """  
    health_lake_client = boto3.client("healthlake")  
    return cls(health_lake_client)  
  
def delete_fhir_datastore(self, datastore_id: str) -> None:  
    """  
    Deletes a HealthLake data store.  
    :param datastore_id: The data store ID.  
    """
```

```

try:

self.health_lake_client.delete_fhir_datastore(DatastoreId=datastore_id)
except ClientError as err:
    logger.exception(
        "Couldn't delete data store with ID %s. Here's why %s",
        datastore_id,
        err.response["Error"]["Message"],
    )
    raise

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DeleteFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
  oo_result = lo_hll->deletefhirdatastore(
    iv_datastoreid = iv_datastore_id
  ).
  MESSAGE 'Data store deleted successfully.' TYPE 'I'.
  CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
  DATA(lv_error) = |Access denied: { lo_access_ex->av_err_code }-
{ lo_access_ex->av_err_msg }|.

```

```

    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
  CATCH /aws1/cx_hllconflictexception INTO DATA(lo_conflict_ex).
    lv_error = |Conflict error: { lo_conflict_ex->av_err_code }-
  { lo_conflict_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_conflict_ex.
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
  { lo_notfound_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_notfound_ex.
ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DeleteFHIRDatastore](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

DescribeFHIRDatastore 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 DescribeFHIRDatastore。

CLI

AWS CLI

描述 FHIR 資料存放區

下列 describe-fhir-datastore 範例示範如何尋找資料存放區 in AWS HealthLake 的屬性。

```

aws healthlake describe-fhir-datastore \
  --datastore-id "1f2f459836ac6c513ce899f9e4f66a59"

```

輸出：

```

{
  "DatastoreProperties": {

```

```

    "PreloadDataConfig": {
      "PreloadDataType": "SYNTHEA"
    },
    "SseConfiguration": {
      "KmsEncryptionConfig": {
        "CmkType": "CUSTOMER_MANAGED_KMS_KEY",
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      }
    },
    "DatastoreName": "Demo",
    "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/<Data store ID>",
    "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/<Data store ID>/r4/",
    "DatastoreStatus": "ACTIVE",
    "DatastoreTypeVersion": "R4",
    "CreatedAt": 1603761064.881,
    "DatastoreId": "<Data store ID>",
    "IdentityProviderConfiguration": {
      "AuthorizationStrategy": "AWS_AUTH",
      "FineGrainedAuthorizationEnabled": false
    }
  }
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRDatastore](#)。

Python

適用於 Python 的 SDK (Boto3)

```

@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")

```

```
return cls(health_lake_client)

def describe_fhir_datastore(self, datastore_id: str) -> dict[str, any]:
    """
    Describes a HealthLake data store.
    :param datastore_id: The data store ID.
    :return: The data store description.
    """
    try:
        response = self.health_lake_client.describe_fhir_datastore(
            DatastoreId=datastore_id
        )
        return response["DatastoreProperties"]
    except ClientError as err:
        logger.exception(
            "Couldn't describe data store with ID %s. Here's why %s",
            datastore_id,
            err.response["Error"]["Message"],
        )
        raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRDatastore](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  oo_result = lo_hll->describe.fhir.datastore(  
    iv_datastoreid = iv_datastore_id  
  ).  
  DATA(lo_datastore_properties) = oo_result->get_datastoreproperties( ).  
  IF lo_datastore_properties IS BOUND.  
    DATA(lv_datastore_name) = lo_datastore_properties->  
>get_datastorename( ).  
    DATA(lv_datastore_status) = lo_datastore_properties->  
>get_datastorestatus( ).  
    MESSAGE 'Data store described successfully.' TYPE 'I'.  
  ENDIF.  
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    lv_error = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRDatastore](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

DescribeFHIRExportJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 DescribeFHIRExportJob。

CLI

AWS CLI

描述 FHIR 匯出任務

下列describe-fhir-export-job範例顯示如何在 AWS HealthLake 中尋找 FHIR 匯出任務的屬性。

```
aws healthlake describe-fhir-export-job \  
  --datastore-id (Data store ID) \  
  --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

輸出：

```
{  
  "ExportJobProperties": {  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "IN_PROGRESS",  
    "JobId": "9009813e9d69ba7cf79bcb3468780f16",  
    "SubmitTime": "2024-11-20T11:31:46.672000-05:00",  
    "EndTime": "2024-11-20T11:34:01.636000-05:00",  
    "OutputDataConfig": {  
      "S3Configuration": {  
        "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",  
        "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-  
b56c-4216-a250-f4c43ef46e83"  
      }  
    },  
    "DatastoreId": "(Data store ID)"  
  }  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRExportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.
```

```
        :return: An instance of HealthLakeWrapper initialized with the default
HealthLake client.
        """
        health_lake_client = boto3.client("healthlake")
        return cls(health_lake_client)

def describe_fhir_export_job(
    self, datastore_id: str, job_id: str
) -> dict[str, any]:
    """
    Describes a HealthLake export job.
    :param datastore_id: The data store ID.
    :param job_id: The export job ID.
    :return: The export job description.
    """
    try:
        response = self.health_lake_client.describe_fhir_export_job(
            DatastoreId=datastore_id, JobId=job_id
        )
        return response["ExportJobProperties"]
    except ClientError as err:
        logger.exception(
            "Couldn't describe export job with ID %s. Here's why %s",
            job_id,
            err.response["Error"]["Message"],
        )
        raise
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRExportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  oo_result = lo_hll->describefhirexportjob(  
    iv_datastoreid = iv_datastore_id  
    iv_jobid = iv_job_id  
  ).  
  DATA(lo_export_job_properties) = oo_result->get_exportjobproperties( ).  
  IF lo_export_job_properties IS BOUND.  
    DATA(lv_job_status) = lo_export_job_properties->get_jobstatus( ).  
    MESSAGE |Export job status: { lv_job_status }.| TYPE 'I'.  
  ENDIF.  
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-  
  { lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    lv_error = |Validation error: { lo_validation_ex->av_err_code }-  
  { lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRExportJob](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

DescribeFHIRImportJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 DescribeFHIRImportJob。

CLI

AWS CLI

描述 FHIR 匯入任務

下列 describe-fhir-import-job 範例顯示如何使用 AWS HealthLake 學習 FHIR 匯入任務的屬性。

```
aws healthlake describe-fhir-import-job \  
  --datastore-id (Data store ID) \  
  --job-id c145fbb27b192af392f8ce6e7838e34f
```

輸出：

```
{  
  "ImportJobProperties": {  
    "InputDataConfig": {  
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"  
      { "arrayitem2": 2 }  
    },  
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)",  
    "JobStatus": "COMPLETED",  
    "JobId": "c145fbb27b192af392f8ce6e7838e34f",  
    "SubmitTime": 1606272542.161,  
    "EndTime": 1606272609.497,  
    "DatastoreId": "(Data store ID)"  
  }  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [DescribeFHIRImportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":
```

```
"""
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def describe_fhir_import_job(
    self, datastore_id: str, job_id: str
) -> dict[str, any]:
    """
    Describes a HealthLake import job.
    :param datastore_id: The data store ID.
    :param job_id: The import job ID.
    :return: The import job description.
    """
    try:
        response = self.health_lake_client.describe_fhir_import_job(
            DatastoreId=datastore_id, JobId=job_id
        )
        return response["ImportJobProperties"]
    except ClientError as err:
        logger.exception(
            "Couldn't describe import job with ID %s. Here's why %s",
            job_id,
            err.response["Error"]["Message"],
        )
        raise
```


- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [DescribeFHIRImportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

 Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  " iv_job_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
  oo_result = lo_hll->describefhirimportjob(  
    iv_datastoreid = iv_datastore_id  
    iv_jobid = iv_job_id  
  ).  
  DATA(lo_import_job_properties) = oo_result->get_importjobproperties( ).  
  IF lo_import_job_properties IS BOUND.  
    DATA(lv_job_status) = lo_import_job_properties->get_jobstatus( ).  
    MESSAGE |Import job status: { lv_job_status }.| TYPE 'I'.  
  ENDIF.  
  CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    DATA(lv_error) = |Resource not found: { lo_notfound_ex->av_err_code }-  
  { lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
  CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    lv_error = |Validation error: { lo_validation_ex->av_err_code }-  
  { lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [DescribeFHIRImportJob](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

ListFHIRDatastores 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListFHIRDatastores。

CLI

AWS CLI

列出 FHIR 資料存放區

下列list-fhir-datastores範例顯示如何使用 命令，以及使用者如何根據 AWS HealthLake 中的資料存放區狀態篩選結果。

```
aws healthlake list-fhir-datastores \  
  --filter DatastoreStatus=ACTIVE
```

輸出：

```
{  
  "DatastorePropertiesList": [  
    {  
      "PreloadDataConfig": {  
        "PreloadDataType": "SYNTHEA"  
      },  
      "SseConfiguration": {  
        "KmsEncryptionConfig": {  
          "CmkType": "CUSTOMER_MANAGED_KMS_KEY",  
          "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
        }  
      },  
      "DatastoreName": "Demo",  
      "DatastoreArn": "arn:aws:healthlake:us-east-1:<AWS Account ID>:datastore/  
<Data store ID>",  
      "DatastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/  
datastore/<Data store ID>/r4/",  
      "DatastoreStatus": "ACTIVE",  
      "DatastoreTypeVersion": "R4",  
      "CreatedAt": 1603761064.881,  
      "DatastoreId": "<Data store ID>",  
      "IdentityProviderConfiguration": {  
        "AuthorizationStrategy": "AWS_AUTH",  
        "FineGrainedAuthorizationEnabled": false  
      }  
    }  
  ]  
}
```

```

    }
  }
]
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRDatastores](#)。

Python

適用於 Python 的 SDK (Boto3)

```

@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_fhir_datastores(self) -> list[dict[str, any]]:
    """
    Lists all HealthLake data stores.
    :return: A list of data store descriptions.
    """
    try:
        next_token = None
        datastores = []

        # Loop through paginated results.
        while True:
            parameters = {}
            if next_token is not None:
                parameters["NextToken"] = next_token
            response =
self.health_lake_client.list_fhir_datastores(**parameters)
            datastores.extend(response["DatastorePropertiesList"])
            if "NextToken" in response:

```

```

        next_token = response["NextToken"]
    else:
        break

    return datastores
except ClientError as err:
    logger.exception(
        "Couldn't list data stores. Here's why %s", err.response["Error"]
["Message"])
    )
    raise

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRDatastores](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
  oo_result = lo_hll->listfhirdatastores( ).
  DATA(lt_datastores) = oo_result->get_datastorepropertieslist( ).
  DATA(lv_datastore_count) = lines( lt_datastores ).
  MESSAGE |Found { lv_datastore_count } data store(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
  DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
  MESSAGE lv_error TYPE 'I'.

```

```

        RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
        lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_throttling_ex.
    ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIREdatastores](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

ListFHIRExportJobs 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListFHIRExportJobs。

CLI

AWS CLI

列出所有 FHIR 匯出任務

下列 list-fhir-export-jobs 範例示範如何使用命令來檢視與帳戶相關聯的匯出任務清單。

```

aws healthlake list-fhir-export-jobs \
  --datastore-id (Data store ID) \
  --submitted-before (DATE like 2024-10-13T19:00:00Z) \
  --submitted-after (DATE like 2020-10-13T19:00:00Z) \
  --job-name "FHIR-EXPORT" \
  --job-status SUBMITTED \
  --max-results (Integer between 1 and 500)

```

輸出：

```

{
  "ExportJobPropertiesList": [
    {

```

```

        "ExportJobProperties": {
            "OutputDataConfig": {
                "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                "S3Configuration": {
                    "S3Uri": "s3://(Bucket Name)/(Prefix Name)/",
                    "KmsKeyId": "(KmsKey Id)"
                }
            },
            "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role
Name)",
            "JobStatus": "COMPLETED",
            "JobId": "c145fbb27b192af392f8ce6e7838e34f",
            "JobName": "FHIR-EXPORT",
            "SubmitTime": "2024-11-20T11:31:46.672000-05:00",
            "EndTime": "2024-11-20T11:34:01.636000-05:00",
            "DatastoreId": "(Data store ID)"
        }
    }
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRExportJobs](#)。

Python

適用於 Python 的 SDK (Boto3)

```

    @classmethod
    def from_client(cls) -> "HealthLakeWrapper":
        """
        Creates a HealthLakeWrapper instance with a default AWS HealthLake
        client.

        :return: An instance of HealthLakeWrapper initialized with the default
        HealthLake client.
        """
        health_lake_client = boto3.client("healthlake")
        return cls(health_lake_client)

    def list_fhir_export_jobs(
        self,

```

```

    datastore_id: str,
    job_name: str = None,
    job_status: str = None,
    submitted_before: datetime = None,
    submitted_after: datetime = None,
) -> list[dict[str, any]]:
    """
    Lists HealthLake export jobs satisfying the conditions.
    :param datastore_id: The data store ID.
    :param job_name: The export job name.
    :param job_status: The export job status.
    :param submitted_before: The export job submitted before the specified
date.
    :param submitted_after: The export job submitted after the specified
date.
    :return: A list of export jobs.
    """
    try:
        parameters = {"DatastoreId": datastore_id}
        if job_name is not None:
            parameters["JobName"] = job_name
        if job_status is not None:
            parameters["JobStatus"] = job_status
        if submitted_before is not None:
            parameters["SubmittedBefore"] = submitted_before
        if submitted_after is not None:
            parameters["SubmittedAfter"] = submitted_after
        next_token = None
        jobs = []
        # Loop through paginated results.
        while True:
            if next_token is not None:
                parameters["NextToken"] = next_token
            response =
self.health_lake_client.list_fhir_export_jobs(**parameters)
            jobs.extend(response["ExportJobPropertiesList"])
            if "NextToken" in response:
                next_token = response["NextToken"]
            else:
                break
        return jobs
    except ClientError as err:
        logger.exception(
            "Couldn't list export jobs. Here's why %s",

```

```

        err.response["Error"]["Message"],
    )
    raise

```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRExportJobs](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```

TRY.
    " iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    IF iv_submitted_after IS NOT INITIAL.
        oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
            iv_submittedafter = iv_submitted_after
        ).
    ELSE.
        oo_result = lo_hll->listfhirexportjobs(
            iv_datastoreid = iv_datastore_id
        ).
    ENDIF.
    DATA(lt_export_jobs) = oo_result->get_exportjobpropertieslist( ).
    DATA(lv_job_count) = lines( lt_export_jobs ).
    MESSAGE |Found { lv_job_count } export job(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).

```

```

        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIRExportJobs](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

ListFHIRImportJobs 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListFHIRImportJobs。

CLI

AWS CLI

列出所有 FHIR 匯入任務

下列 list-fhir-import-jobs 範例示範如何使用命令來檢視與帳戶相關聯的所有匯入任務的清單。

```

aws healthlake list-fhir-import-jobs \
  --datastore-id (Data store ID) \
  --submitted-before (DATE like 2024-10-13T19:00:00Z) \
  --submitted-after (DATE like 2020-10-13T19:00:00Z) \
  --job-name "FHIR-IMPORT" \
  --job-status SUBMITTED \
  --max-results (Integer between 1 and 500)

```

輸出：

```
{
```

```

"ImportJobPropertiesList": [
  {
    "JobId": "c0fddbf76f238297632d4aebdbfc9ddf",
    "JobStatus": "COMPLETED",
    "SubmitTime": "2024-11-20T10:08:46.813000-05:00",
    "EndTime": "2024-11-20T10:10:09.093000-05:00",
    "DatastoreId": "(Data store ID)",
    "InputDataConfig": {
      "S3Uri": "s3://(Bucket Name)/(Prefix Name)/"
    },
    "JobOutputDataConfig": {
      "S3Configuration": {
        "S3Uri": "s3://(Bucket Name)/
import/6407b9ae4c2def3cb6f1a46a0c599ec0-FHIR_IMPORT-
c0fddbf76f238297632d4aebdbfc9ddf/",
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/b7f645cb-
e564-4981-8672-9e012d1ff1a0"
      }
    },
    "JobProgressReport": {
      "TotalNumberOfScannedFiles": 1,
      "TotalSizeOfScannedFilesInMB": 0.001798,
      "TotalNumberOfImportedFiles": 1,
      "TotalNumberOfResourcesScanned": 1,
      "TotalNumberOfResourcesImported": 1,
      "TotalNumberOfResourcesWithCustomerError": 0,
      "TotalNumberOfFilesReadWithCustomerError": 0,
      "Throughput": 0.0
    },
    "DataAccessRoleArn": "arn:aws:iam::(AWS Account ID):role/(Role Name)"
  }
]
}

```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListFHIRImportJobs](#)。

Python

適用於 Python 的 SDK (Boto3)

```

@classmethod
def from_client(cls) -> "HealthLakeWrapper":

```

```
"""
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def list_fhir_import_jobs(
    self,
    datastore_id: str,
    job_name: str = None,
    job_status: str = None,
    submitted_before: datetime = None,
    submitted_after: datetime = None,
) -> list[dict[str, any]]:
    """
    Lists HealthLake import jobs satisfying the conditions.
    :param datastore_id: The data store ID.
    :param job_name: The import job name.
    :param job_status: The import job status.
    :param submitted_before: The import job submitted before the specified
    date.
    :param submitted_after: The import job submitted after the specified
    date.
    :return: A list of import jobs.
    """
    try:
        parameters = {"DatastoreId": datastore_id}
        if job_name is not None:
            parameters["JobName"] = job_name
        if job_status is not None:
            parameters["JobStatus"] = job_status
        if submitted_before is not None:
            parameters["SubmittedBefore"] = submitted_before
        if submitted_after is not None:
            parameters["SubmittedAfter"] = submitted_after
        next_token = None
        jobs = []
        # Loop through paginated results.
        while True:
```

```
        if next_token is not None:
            parameters["NextToken"] = next_token
        response =
self.health_lake_client.list_fhir_import_jobs(**parameters)
        jobs.extend(response["ImportJobPropertiesList"])
        if "NextToken" in response:
            next_token = response["NextToken"]
        else:
            break
    return jobs
except ClientError as err:
    logger.exception(
        "Couldn't list import jobs. Here's why %s",
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListFHIRImportJobs](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

TRY.

```
" iv_datastore_id = 'a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
IF iv_submitted_after IS NOT INITIAL.
```

```
oo_result = lo_hll->listfhirimportjobs(
    iv_datastoreid = iv_datastore_id
    iv_submittedafter = iv_submitted_after
).
ELSE.
oo_result = lo_hll->listfhirimportjobs(
    iv_datastoreid = iv_datastore_id
).
ENDIF.
DATA(lt_import_jobs) = oo_result->get_importjobpropertieslist( ).
DATA(lv_job_count) = lines( lt_import_jobs ).
MESSAGE |Found { lv_job_count } import job(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_validation_ex.
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
MESSAGE lv_error TYPE 'I'.
RAISE EXCEPTION lo_notfound_ex.
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListFHIRImportJobs](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

ListTagsForResource 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 ListTagsForResource。

CLI

AWS CLI

列出資料存放區的標籤

下列 list-tags-for-resource 範例列出與指定資料存放區相關聯的標籤：

```
aws healthlake list-tags-for-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/  
  fhir/0725c83f4307f263e16fd56b6d8ebdbe"
```

輸出：

```
{  
  "tags": {  
    "key": "value",  
    "key1": "value1"  
  }  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [ListTagsForResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.  
    """  
    health_lake_client = boto3.client("healthlake")  
    return cls(health_lake_client)  
  
def list_tags_for_resource(self, resource_arn: str) -> dict[str, str]:  
    """  
    Lists the tags for a HealthLake resource.  
    :param resource_arn: The resource ARN.  
    :return: The tags for the resource.  
    """  
    try:  
        response = self.health_lake_client.list_tags_for_resource(  
            ResourceARN=resource_arn
```

```
    )
    return response["Tags"]
except ClientError as err:
    logger.exception(
        "Couldn't list tags for resource %s. Here's why %s",
        resource_arn,
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [ListTagsForResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
    DATA(lo_result) = lo_hll->listtagsforresource(
        iv_resourcearn = iv_resource_arn
    ).
    ot_tags = lo_result->get_tags( ).
    DATA(lv_tag_count) = lines( ot_tags ).
    MESSAGE |Found { lv_tag_count } tag(s).| TYPE 'I'.
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
```

```

        DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).
        lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-
{ lo_notfound_ex->av_err_msg }|.
        MESSAGE lv_error TYPE 'I'.
        RAISE EXCEPTION lo_notfound_ex.
    ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [ListTagsForResource](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

StartFHIRExportJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 StartFHIRExportJob。

CLI

AWS CLI

啟動 FHIR 匯出任務

下列 start-fhir-export-job 範例顯示如何使用 AWS HealthLake 啟動 FHIR 匯出任務。

```

aws healthlake start-fhir-export-job \
  --output-data-config '{"S3Configuration": {"S3Uri": "s3://(Bucket Name)/
(Prefix Name)/", "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-
b56c-4216-a250-f4c43ef46e83"}}' \
  --datastore-id (Data store ID) \
  --data-access-role-arn arn:aws:iam::(AWS Account ID):role/(Role Name)

```

輸出：

```

{
  "DatastoreId": "(Data store ID)",
  "JobStatus": "SUBMITTED",

```

```
"JobId": "9b9a51943afaedd0a8c0c26c49135a31"
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [StartFHIRExportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def start_fhir_export_job(
    self,
    job_name: str,
    datastore_id: str,
    output_s3_uri: str,
    kms_key_id: str,
    data_access_role_arn: str,
) -> dict[str, str]:
    """
    Starts a HealthLake export job.
    :param job_name: The export job name.
    :param datastore_id: The data store ID.
    :param output_s3_uri: The output S3 URI.
    :param kms_key_id: The KMS key ID associated with the output S3 bucket.
    :param data_access_role_arn: The data access role ARN.
    :return: The export job.
    """
    try:
        response = self.health_lake_client.start_fhir_export_job(
            OutputDataConfig={
```

```
        "S3Configuration": {"S3Uri": output_s3_uri, "KmsKeyId":
kms_key_id}
    },
    DataAccessRoleArn=data_access_role_arn,
    DatastoreId=datastore_id,
    JobName=job_name,
)

return response
except ClientError as err:
    logger.exception(
        "Couldn't start export job. Here's why %s",
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [StartFHIRExportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_job_name = 'MyExportJob'
    " iv_output_s3_uri = 's3://my-bucket/export/output/'
```

```

    " iv_kms_key_id = 'arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'
    " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/
HealthLakeExportRole'
    oo_result = lo_hll->startfhirexportjob(
      iv_jobname = iv_job_name
      io_outputdataconfig = NEW /aws1/cl_hlloutputdataconfig(
        io_s3configuration = NEW /aws1/cl_hlls3configuration(
          iv_s3uri = iv_output_s3_uri
          iv_kmskeyid = iv_kms_key_id
        )
      )
      iv_dataaccessrolearn = iv_data_access_role_arn
      iv_datastoreid = iv_datastore_id
    ).
    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Export job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
  ENDRTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [StartFHIRExportJob](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

StartFHIRImportJob 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 StartFHIRImportJob。

CLI

AWS CLI

啟動 FHIR 匯入任務

下列start-fhir-import-job範例示範如何使用 AWS HealthLake 啟動 FHIR 匯入任務。

```
aws healthlake start-fhir-import-job \  
  --input-data-config S3Uri="s3://(Bucket Name)/(Prefix Name)/" \  
  --job-output-data-config '{"S3Configuration": {"S3Uri": "s3://(Bucket Name)/  
(Prefix Name)/", "KmsKeyId": "arn:aws:kms:us-east-1:012345678910:key/d330e7fc-  
b56c-4216-a250-f4c43ef46e83"}}' \  
  --datastore-id (Data store ID) \  
  --data-access-role-arn "arn:aws:iam::(AWS Account ID):role/(Role Name)"
```

輸出：

```
{  
  "DatastoreId": "(Data store ID)",  
  "JobStatus": "SUBMITTED",  
  "JobId": "c145fbb27b192af392f8ce6e7838e34f"  
}
```

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [StartFHIRImportJob](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.
```

```
"""
health_lake_client = boto3.client("healthlake")
return cls(health_lake_client)

def start_fhir_import_job(
    self,
    job_name: str,
    datastore_id: str,
    input_s3_uri: str,
    job_output_s3_uri: str,
    kms_key_id: str,
    data_access_role_arn: str,
) -> dict[str, str]:
    """
    Starts a HealthLake import job.
    :param job_name: The import job name.
    :param datastore_id: The data store ID.
    :param input_s3_uri: The input S3 URI.
    :param job_output_s3_uri: The job output S3 URI.
    :param kms_key_id: The KMS key ID associated with the output S3 bucket.
    :param data_access_role_arn: The data access role ARN.
    :return: The import job.
    """
    try:
        response = self.health_lake_client.start_fhir_import_job(
            JobName=job_name,
            InputDataConfig={"S3Uri": input_s3_uri},
            JobOutputDataConfig={
                "S3Configuration": {
                    "S3Uri": job_output_s3_uri,
                    "KmsKeyId": kms_key_id,
                }
            },
            DataAccessRoleArn=data_access_role_arn,
            DatastoreId=datastore_id,
        )
        return response
    except ClientError as err:
        logger.exception(
            "Couldn't start import job. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [StartFHIRImportJob](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
  " iv_job_name = 'MyImportJob'  
  " iv_input_s3_uri = 's3://my-bucket/import/data.ndjson'  
  " iv_job_output_s3_uri = 's3://my-bucket/import/output/'  
  " iv_kms_key_id = 'arn:aws:kms:us-  
east-1:123456789012:key/12345678-1234-1234-1234-123456789012'  
  " iv_data_access_role_arn = 'arn:aws:iam::123456789012:role/  
HealthLakeImportRole'  
  oo_result = lo_hll->startfhirimportjob(  
    iv_jobname = iv_job_name  
    io_inputdataconfig = NEW /aws1/cl_hllinputdataconfig( iv_s3uri =  
iv_input_s3_uri )  
    io_joboutputdataconfig = NEW /aws1/cl_hlloutputdataconfig(  
      io_s3configuration = NEW /aws1/cl_hlls3configuration(  
        iv_s3uri = iv_job_output_s3_uri  
        iv_kmskeyid = iv_kms_key_id  
      )  
    )  
  )  
  iv_dataaccessrolearn = iv_data_access_role_arn
```

```

        iv_datastoreid = iv_datastore_id
    ).
    DATA(lv_job_id) = oo_result->get_jobid( ).
    MESSAGE |Import job started with ID { lv_job_id }.| TYPE 'I'.
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-
{ lo_validation_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_validation_ex.
    CATCH /aws1/cx_hllthrottlingex INTO DATA(lo_throttling_ex).
    lv_error = |Throttling error: { lo_throttling_ex->av_err_code }-
{ lo_throttling_ex->av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_throttling_ex.
    CATCH /aws1/cx_hllaccessdeniedex INTO DATA(lo_access_ex).
    lv_error = |Access denied: { lo_access_ex->av_err_code }-{ lo_access_ex-
>av_err_msg }|.
    MESSAGE lv_error TYPE 'I'.
    RAISE EXCEPTION lo_access_ex.
ENDTRY.

```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [StartFHIRImportJob](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

TagResource 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 TagResource。

CLI

AWS CLI

將標籤新增至資料存放區

以下 tag-resource 範例示範如何將標籤新增至資料存放區。

```

aws healthlake tag-resource \
  --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \

```

```
--tags '[{"Key": "key1", "Value": "value1"}]'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [TagResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod
def from_client(cls) -> "HealthLakeWrapper":
    """
    Creates a HealthLakeWrapper instance with a default AWS HealthLake
    client.

    :return: An instance of HealthLakeWrapper initialized with the default
    HealthLake client.
    """
    health_lake_client = boto3.client("healthlake")
    return cls(health_lake_client)

def tag_resource(self, resource_arn: str, tags: list[dict[str, str]]) ->
None:
    """
    Tags a HealthLake resource.
    :param resource_arn: The resource ARN.
    :param tags: The tags to add to the resource.
    """
    try:
        self.health_lake_client.tag_resource(ResourceARN=resource_arn,
        Tags=tags)
    except ClientError as err:
        logger.exception(
            "Couldn't tag resource %s. Here's why %s",
            resource_arn,
            err.response["Error"]["Message"],
        )
        raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [TagResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.  
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/  
fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'  
    lo_hll->tagresource(  
        iv_resourcearn = iv_resource_arn  
        it_tags = it_tags  
    ).  
    MESSAGE 'Resource tagged successfully.' TYPE 'I'.  
    CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
    DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_validation_ex.  
    CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
    lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
    MESSAGE lv_error TYPE 'I'.  
    RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [TagResource](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

UntagResource 搭配 AWS SDK 或 CLI 使用

下列程式碼範例示範如何使用 UntagResource。

CLI

AWS CLI

從資料存放區移除標籤。

下列 untag-resource 範例示範如何從資料存放區移除標籤。

```
aws healthlake untag-resource \  
  --resource-arn "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/  
b91723d65c6fdeb1d26543a49d2ed1fa" \  
  --tag-keys '["key1"]'
```

此命令不會產生輸出。

- 如需 API 詳細資訊，請參閱《AWS CLI 命令參考》中的 [UntagResource](#)。

Python

適用於 Python 的 SDK (Boto3)

```
@classmethod  
def from_client(cls) -> "HealthLakeWrapper":  
    """  
    Creates a HealthLakeWrapper instance with a default AWS HealthLake  
    client.  
  
    :return: An instance of HealthLakeWrapper initialized with the default  
    HealthLake client.  
    """  
    health_lake_client = boto3.client("healthlake")  
    return cls(health_lake_client)  
  
def untag_resource(self, resource_arn: str, tag_keys: list[str]) -> None:
```

```
"""
Untags a HealthLake resource.
:param resource_arn: The resource ARN.
:param tag_keys: The tag keys to remove from the resource.
"""
try:
    self.health_lake_client.untag_resource(
        ResourceARN=resource_arn, TagKeys=tag_keys
    )
except ClientError as err:
    logger.exception(
        "Couldn't untag resource %s. Here's why %s",
        resource_arn,
        err.response["Error"]["Message"],
    )
    raise
```

- 如需 API 詳細資訊，請參閱《AWS SDK for Python (Boto3) API 參考》中的 [UntagResource](#)。

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

SAP ABAP

適用於 SAP ABAP 的開發套件

Note

GitHub 上提供更多範例。尋找完整範例，並了解如何在 [AWS 程式碼範例儲存庫](#) 中設定和執行。

```
TRY.
    " iv_resource_arn = 'arn:aws:healthlake:us-east-1:123456789012:datastore/
    fhir/a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6'
```

```
lo_hll->untagresource(  
    iv_resourcearn = iv_resource_arn  
    it_tagkeys = it_tag_keys  
).  
MESSAGE 'Resource untagged successfully.' TYPE 'I'.  
CATCH /aws1/cx_hllvalidationex INTO DATA(lo_validation_ex).  
DATA(lv_error) = |Validation error: { lo_validation_ex->av_err_code }-  
{ lo_validation_ex->av_err_msg }|.  
MESSAGE lv_error TYPE 'I'.  
RAISE EXCEPTION lo_validation_ex.  
CATCH /aws1/cx_hllresourcenotfoundex INTO DATA(lo_notfound_ex).  
lv_error = |Resource not found: { lo_notfound_ex->av_err_code }-  
{ lo_notfound_ex->av_err_msg }|.  
MESSAGE lv_error TYPE 'I'.  
RAISE EXCEPTION lo_notfound_ex.  
ENDTRY.
```

- 如需 API 詳細資訊，請參閱《適用於 AWS SAP ABAP 的 SDK API 參考》中的 [UntagResource](#)。

如需 AWS SDK 開發人員指南和程式碼範例的完整清單，請參閱 [搭配 AWS SDK 使用 HealthLake](#)。此主題也包含有關入門的資訊和舊版 SDK 的詳細資訊。

整合 AWS HealthLake

下列 AWS 服務會直接與 整合 AWS HealthLake ，以啟用整合的自然語言處理、SQL 查詢和資料倉儲。

- Amazon Comprehend Medical 是一種符合 HIPAA 資格的自然語言處理服務 (NLP)，使用機器學習程式庫從 HealthLake 中的非結構化醫療文字擷取有意義的運作狀態資料。如需詳細資訊，請參閱 [Amazon Comprehend Medical 開發人員指南](#)。
- Amazon Athena 是一種互動式查詢服務，可讓您使用標準 SQL 直接在 Amazon Simple Storage Service (Amazon S3) 儲存貯體中分析 HealthLake 資料。如需詳細資訊，請參閱 [Amazon Athena 開發人員指南](#)。

主題

- [HealthLake 的整合式自然語言處理 \(NLP\)](#)
- [使用 Amazon Athena 查詢 HealthLake 資料](#)

HealthLake 的整合式自然語言處理 (NLP)

AWS HealthLake 提供整合的自然語言處理 (NLP) 程式庫，用於剖析、識別和映射存放在 FHIR [DocumentReference](#) 資源類型中的非結構化資料。

Important (重要)

適用於 HealthLake 的整合式 NLP 預設為關閉。若要將其開啟，請使用 [提交支援案例AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。

HealthLake 整合 NLP 的運作方式是呼叫 Amazon Comprehend Medical DetectEntities-V2、InferICD10-CM和 InferRxNorm API 動作。動作會將結果附加為DocumentReference資源的延伸。當 Amazon Comprehend Medical API 動作偵測到 SIGN、SYMPTOM和/或 的特徵DIAGNOSIS時，它們會產生 FHIR [Linkage](#) 資源。新的 Condition和 Observation 資源是從以 SIGN、SYMPTOM或 特性識別的實體建立，DIAGNOSIS並使用 Linkage 資源連結到來源文件。

Note

雖然 HealthLake 整合 NLP 產生的 FHIR 資源支援 GET 請求，但 FHIR API search 功能則不支援。若要進一步了解如何使用 HealthLake 與 Athena 的整合來搜尋 NLP 延伸模組，請參閱 [SQL 索引和查詢](#)。

內容

- [HealthLake 整合式 NLP 程式庫](#)
- [使用 FHIR REST API 互動](#)
- [HealthLake 整合式 NLP 的搜尋參數](#)
- [HealthLake 整合 NLP 範例請求](#)

HealthLake 整合式 NLP 程式庫

HealthLake 會使用 Amazon Comprehend Medical 程式庫推斷 DocumentReference 資源類型中找到的資料。Amazon Comprehend Medical API 操作 DetectEntities-V2、InferICD10-CM 和 InferRxNorm 會將醫療情況 InferRxNorm 偵測為特徵。每個操作都提供不同的洞見。

⚠ 語言支援

Amazon Comprehend Medical API 操作只會偵測英文文字的醫療實體。

- DetectEntities-V2：檢查各種醫療實體的臨床文字，並傳回其特定資訊，例如實體類別、位置和可信度分數。
- InferICD10-CM：將病患記錄中的醫療狀況偵測為實體，並將這些實體連結至 CDC 國家健康統計中心 (WHO) 授權的 ICD-10-CM 知識庫中的標準化概念識別符。
- InferRxNorm：將藥物偵測為患者記錄中列出的實體，並將其連結至國家醫學圖書館 RxNorm 資料庫中的標準化概念識別符。

每個 API 操作支援的特徵為 SIGN、SYMPTOM 和 DIAGNOSIS。如果偵測到特徵，則會將其新增為 HealthLake 資料存放區中不同位置的 FHIR 相容延伸。

新增擴充功能的位置。

- DocumentReference : Amazon Comprehend Medical API 操作的結果會新增為 DocumentReference 資源類型中找到extension的每個文件的。延伸模組中的結果分為兩個群組。您可以根據它們的，在結果中找到它們URL。
 - <http://healthlake.amazonaws.com/system-generated-resources/>
 - 這些是由 HealthLake 建立或新增至 的資源類型。
 - <http://healthlake.amazonaws.com/aws-cm/>
 - 將 Amazon Comprehend Medical API 操作的原始輸出新增至 HealthLake 資料存放區的位置。
- Linkage : 此資源類型是由於整合的 NLP 而新增或建立。特定 上的GET請求會Linkage傳回連結資源的清單。若要識別 HealthLake Linkage 是否已新增，請尋找新增的"tag": [{"display": "SYSTEM_GENERATED"}]鍵/值對。若要進一步了解 Linkage 的 FHIR 規格，請參閱 FHIR R4 文件[Linkage](#)中的。
- Amazon Comprehend Medical 操作所產生的 FHIR 資源類型。
 - Observation : 包含 Amazon Comprehend Medical API 動作的結果，DetectEntities-V2以及特徵為 SIGN或 InferICD10-CM時的結果SYMPTOM。
 - Condition : 包含 Amazon Comprehend Medical API 動作的結果，DetectEntities-V2以及特徵為 InferICD10-CM時的結果DIAGNOSIS。
 - MedicationStatement : 包含來自 Amazon Comprehend Medical API 動作 的結果InferRxNorm。

使用 FHIR REST API 互動

根據預設，Amazon Comprehend Medical API 操作在提出GET請求時不會傳回偵測到的特徵。若要查看整合 NLP 操作的結果，您必須為下列 FHIR 資源類型指定已知ID的。

- Linkage
- Observation
- Condition
- MedicationStatement

在指定 ID已知包含 Amazon Comprehend Medical API 操作結果的GET請求中，可使用DocumentReference資源類型外的 HealthLake 整合 NLP 動作結果。

HealthLake 整合式 NLP 的搜尋參數

下表列出 HealthLake 整合式 NLP 的可搜尋屬性。

HealthLake NLP 的搜尋參數

搜尋參數	尋找的相符項目
detectEntities-entity-category	DetectEntities 子延伸中的實體類別位於 CM AWS 延伸模組中
detectEntities-entity-text	DetectEntities 子延伸中的實體文字位於 CM AWS 延伸模組中
detectEntities-entity-type	CM 延伸模組中 DetectEntities AWS 子延伸中的實體類型
detectEntities-entity-score	CM 延伸模組中 DetectEntities AWS 子延伸中的實體分數
infer-icd10cm-entity-text	CM 延伸模組 InferICD10CM AWS 子延伸中的實體文字
infer-icd10cm-entity-score	CM 延伸模組 InferICD10CM AWS 子延伸中的實體分數
infer-icd10cm-entity-concept-code	CM 延伸模組 InferICD10CM AWS 子延伸中的實體概念碼
infer-icd10cm-entity-concept-description	CM 延伸模組 InferICD10CM 子延伸中的 AWS 實體概念描述
infer-icd10cm-entity-concept-score	CM 延伸模組 InferICD10CM AWS 子延伸中的實體概念分數
infer-rxnorm-entity-score	CM 延伸模組內 InferRxNorm AWS 子延伸中的實體分數
infer-rxnorm-entity-text	CM 延伸模組 InferRxNorm AWS 子延伸中的實體文字

搜尋參數	尋找的相符項目
infer-rxnorm-entity-concept-code	InferRxNorm 子延伸中的實體概念程式碼位於 CM AWS 延伸模組中
infer-rxnorm-entity-concept-description	CM 延伸模組 InferRxNorm 子延伸中的 AWS 實體概念描述
infer-rxnorm-entity-concept-score	CM 延伸模組 InferRxNorm AWS 子延伸中的實體概念分數

HealthLake 提供特殊搜尋，以符合 EntityText 和 EntityCategory 屬於相同實體的條件。下表說明 HealthLake 支援的特殊搜尋參數。

搜尋參數

搜尋參數	傳回相符項目
detectEntities-entity-text-category	如果在 DetectEntities 子延伸中至少有一個符合 entityText 和 entityCategory 的實體。
detectEntities-entity-type-score	如果在 DetectEntities 子延伸中至少有一個符合 entityType 和 entityScore 的實體。
detectEntities-entity-text-score	如果在 DetectEntities 子延伸中至少有一個符合 entityText 和 entityScore 的實體。
detectEntities-entity-text-type	如果在 DetectEntities 子延伸中至少有一個符合 entityText 和 entityType 的實體。
detectEntities-entity-category-score	如果至少有一個實體同時符合 entityCategory 和 entityScore。
infer-icd10cm-entity-text-concept-code	如果 InferICD10CM 子延伸中至少有一個符合 entityText 的實體，且該實體至少有一個符合該程式碼的 conceptCode。

搜尋參數	傳回相符項目
infer-icd10cm-entity-text-concept-score	如果 InferICD10CM 子延伸中至少有一個符合 entityText 的實體，且該實體至少有一個符合分數的 conceptScore。
infer-icd10cm-entity-concept-description-concept-score	如果 InferICD10CM 子延伸中的實體中至少有一個概念符合概念描述和 conceptScore。
infer-rxnorm-entity-text-concept-code	如果 InferRxNorm 子延伸中至少有一個符合 entityText 的實體，且該實體至少有一個符合該程式碼的 conceptCode。
infer-rxnorm-entity-text-concept-score	如果 InferRxNorm 子延伸中至少有一個符合 entityText 的實體，且該實體至少有一個符合分數的 conceptScore。
infer-rxnorm-entity-concept-description-concept-score	如果 InferRxNorm 子延伸中的實體中至少有一個概念符合概念描述和 conceptScore。

HealthLake 整合 NLP 範例請求

範例 1：擷取至 HealthLake 資料存放區的 **Patient** 記錄

以下是以與醫療專業人員的 Patient 相遇為基礎的臨床備註範例。

合成資料

下列範例中的文字是合成內容，不包含受保護的健康資訊 (PHI)。

1991-08-31

Chief Complaint

- Headache
- Sinus Pain
- Nasal Congestion
- Sore Throat
- Pain with Bright Lights

```
- Nasal Discharge
- Cough

# History of Present Illness
Jerónimo599 is a 4 month-old non-hispanic white male.

# Social History
Patient has never smoked.

Patient comes from a middle socioeconomic background.

Patient currently has Aetna.

# Allergies
No Known Allergies.

# Medications
No Active Medications.

# Assessment and Plan
Patient is presenting with bee venom (substance), mold (organism), house dust mite (organism), animal dander (substance), grass pollen (substance), tree pollen (substance), lisinopril, sulfamethoxazole / trimethoprim, fish (substance).

## Plan

The patient was prescribed the following medications:
- astemizole 10 mg oral tablet
- nda020800 0.3 ml epinephrine 1 mg/ml auto-injector
The patient was placed on a careplan:
- self-care interventions (procedure)
```

提醒您，此資訊在 DocumentReference 資源中以 base64 格式編碼。當本文件擷取至 HealthLake 且 Amazon Comprehend Medical API 操作完成時，若要查看結果，您可以從 DocumentReference 資源類型的 GET 請求開始。

```
GET https://https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/DocumentReference
```

當 Amazon Comprehend Medical API 操作成功時，請在 extension 連結至下列項目的 內尋找這些鍵值對 "url": "http://healthlake.amazonaws.com/aws-cm/"

```
{
  "url": "http://healthlake.amazonaws.com/aws-cm/status/",
  "valueString": "SUCCESS"
},
{
  "url": "http://healthlake.amazonaws.com/aws-cm/message/",
  "valueString": "The AWS HealthLake integrated medical NLP operation was successful."
}
```

下列索引標籤顯示如何根據資源類型，在您的 HealthLake 資料存放區中報告擷取的醫療記錄。

DocumentReference

若要查看單一 DocumentReference 資源類型的結果，請提出 GET 請求，其中提供特定資源 id 的。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/
r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/DocumentReference/0e938f03-da7f-4178-acd8-
eea9586c46ed
```

成功時，您會收到 200 HTTP 回應代碼和下列 JSON 回應（為了清楚起見，已截斷）。

以下是 `http://healthlake.amazonaws.com/system-generated-resources/` 部分。您可以看到 Linkage/`e366d29f-2c22-4c19-866e-09603937935a` 已新增新的。您也可以查看 HealthLake 已將推論型問題清單新增至特定 Condition Observation 和資源類型的位置。

若要查看這些資源類型的修改方式，請選擇相關索引標籤。

```
{
  "extension": [
    {
      "url": "http://healthlake.amazonaws.com/linkage",
      "valueReference": {
        "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
      }
    },
    {
      "url": "http://healthlake.amazonaws.com/nlp-entity",
      "valueReference": {
        "reference": "Observation/c6e0a3ff-7a17-4d8b-bfd0-d02d7da090c5"
      }
    }
  ],
}
```

```
{
  "url": "http://healthlake.amazonaws.com/nlp-entity",
  "valueReference": {
    "reference": "Condition/0854e1f3-894d-448e-a8d9-3af5b9902baf"
  }
},
"url": "http://healthlake.amazonaws.com/system-generated-resources/"
}
```

Linkage

若要查看單一Linkage資源類型的結果，請提出 GET 請求，其中提供特定資源ID的。

```
GET https://https://healthlake.region.amazonaws.com/datastore/datastoreId/
r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/Linkage/e366d29f-2c22-4c19-866e-09603937935a
```

成功時，您會收到 200 HTTP 回應代碼，以及下列截斷的 JSON 回應。

回應包含 item 元素。其中，索引鍵/值對 "type": "source" 會指出用來修改的特定 DocumentReference 項目，Condition 並 Observations 列在索引 "type": "alternate" 鍵/值對下。

您也會看到 meta 元素和對應的鍵值對 "tag": [{"display": "SYSTEM_GENERATED"}]，表示這些資源是由 HealthLake 建立。

```
{
  "resourceType": "Linkage",
  "id": "e366d29f-2c22-4c19-866e-09603937935a",
  "active": true,
  "item": [
    {
      "type": "alternate",
      "resource": {
        "reference": "Observation/c6e0a3ff-7a17-4d8b-bfd0-d02d7da090c5",
        "type": "Observation"
      }
    },
    {
      "type": "alternate",
      "resource": {
```

```

    "reference": "Condition/9d5c1ef6-f822-4faf-b55f-7c70f2a4aa8d",
    "type": "Condition"
  },
  {
    "type": "source",
    "resource": {
      "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed",
      "type": "DocumentReference"
    }
  }
],
"meta": {
  "lastUpdated": "2022-10-21T19:38:31.327Z",
  "tag": [{
    "display": "SYSTEM_GENERATED"
  }]
}
}

```

Resource type: Observation

若要查看單一Observation資源類型的結果，請提出 GET 請求，其中提供特定資源ID的。

```

GET https://https://healthlake.region.amazonaws.com/
datastore/datastoreId/r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/
Observation/e366d29f-2c22-4c19-866e-09603937935a

```

Amazon Comprehend Medical API 操作的結果會修改為下列元素：code、meta和modifierExtension。

code

類型的元素CodeableConcept。若要進一步了解，請參閱 FHIR R4 文件[CodeableConcept](#)中的。

HealthLake 會附加下列三個鍵值對。

- "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"：其中 URL 是指特定的 Amazon Comprehend Medical API 操作。在此情況下，InferICD10CM。
- "code": "A52.06"：其中 A52.06是 ICD-10-CM 程式碼，用於從疾病控制中心識別知識庫中找到的概念。

- "display": "Other syphilitic heart involvement": 其中 "Other syphilitic heart involvement" 是上下文中 ICD-10-CM 程式碼的長描述。

下列截斷的 JSON 回應僅包含 code 元素。

```
"code": {
  "coding":
  [
    {
      "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
      "code": "A52.06",
      "display": "Other syphilitic heart involvement"
    }
  ],
  "text": "Other syphilitic heart involvement"
}
```

若要了解模型對指派的 ICD-10-CM 程式碼是否正確的可信度，請使用 modifierExtension 元素。

meta

meta 元素包含中繼資料，指出 code 元素是否包含 Amazon Comprehend Medical API 操作所新增的詳細資訊。

下列截斷的 JSON 回應僅包含 meta 元素。

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.879Z",
  "tag": [{
    "display": "SYSTEM_GENERATED"
  }]
}
```

modifierExtension

modifierExtension 元素包含有關 code 元素中指定代碼之可信度層級的更多詳細資訊。它還具有索引鍵/值對，提供用於產生結果和相關 Linkage 資源類型的原始 DocumentReference 的連結。

對於每個新增的 coding 元素，您會看到 entity-score 和 entity-Concept-Score 新增至 modifierExtension。對於鍵/值對中的每個值，您會看到分數。對於 entity-score，此分數是

Amazon Comprehend Medical 對偵測準確性的可信度。對於 `entity-Concept-Score`，此分數是 Amazon Comprehend Medical 對實體準確連結到 ICD-10-CM 概念的可信度。

下列截斷的 JSON 回應僅包含 `modifierExtension` 元素。

```
"modifierExtension": [{
  "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-score",
  "valueDecimal": 0.45005733
},
{
  "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-Concept-Score",
  "valueDecimal": 0.1111792
},
{
  "url": "http://healthlake.amazonaws.com/system-generated-linkage",
  "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
  }
},
{
  "url": "http://healthlake.amazonaws.com/source-document-reference",
  "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
  }
}
]
```

完整 JSON 回應

```
{
  "subject": {
    "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
  "resourceType": "Observation",
  "status": "unknown",
  "code": {
    "coding": [{
      "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
      "code": "A52.06",
      "display": "Other syphilitic heart involvement"
    }],
  }
}
```

```

    "text": "Other syphilitic heart involvement"
  },
  "meta": {
    "lastUpdated": "2022-10-21T19:38:30.879Z",
    "tag": [{
      "display": "SYSTEM_GENERATED"
    }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-score",
    "valueDecimal": 0.45005733
  },
  {
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-Concept-Score",
    "valueDecimal": 0.1111792
  },
  {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
      "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
      "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
  }
],
  "id": "7e88c7c5-21a5-4dd7-8fc2-a02474fba583"
}

```

Condition

若要查看單一Condition資源類型的結果，請提出 GET 請求，其中提供特定資源ID的。

```

GET https://healthlake.region.amazonaws.com/datastore/datastoreId/
r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/Condition/b06d343d-
ddb8-4f36-82cb-853fcd434dfd

```

Amazon Comprehend Medical API 操作的結果會修改為下列元素：code、meta和 modifierExtension。

code

類型的元素 CodeableConcept。若要進一步了解，請參閱 FHIR R4 文件 [CodeableConcept](#) 中的。

HealthLake 會附加下列三個鍵值對。

- "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/"：其中 URL 是指特定的 Amazon Comprehend Medical API 操作。在此情況下，InferICD10CM。
- "code": "I70.0"：其中 A52.06 是 ICD-10-CM 程式碼，用於從疾病控制中心識別知識庫中找到的概念。
- "display": "Atherosclerosis of aorta"：其中 "Other syphilitic heart involvement" 是本體中 ICD-10-CM 程式碼的長描述。

下列截斷的 JSON 回應僅包含 code 元素。

```
"code": {
  "coding":
  [
    {
      "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
      "code": "I70.0",
      "display": "Atherosclerosis of aorta"
    }
  ],
  "text": "Atherosclerosis of aorta"
}
```

若要了解模型對指派的 ICD-10-CM 程式碼是否正確的可信度，請使用 modifierExtension 元素。

meta

meta 元素包含中繼資料，指出 code 元素是否包含 Amazon Comprehend Medical API 操作所新增的詳細資訊。

下列截斷的 JSON 回應僅包含 meta 元素。

```
"meta": {
  "lastUpdated": "2022-10-21T19:38:30.877Z",
  "tag": [{
    "display": "SYSTEM_GENERATED"
  }]
}
```

modifierExtension

modifierExtension 元素包含有關code元素中指定代碼之可信度層級的更多詳細資訊。它還具有索引鍵/值對，提供用於產生結果和相關 Linkage 資源類型的原始 DocumentReference 的連結。

對於每個新增的coding元素，您會看到 entity-score和 entity-Concept-Score 新增至 modifierExtension。對於鍵/值對中的每個值，您會看到分數。對於 entity-score，此分數是 Amazon Comprehend Medical 對偵測準確性的可信度。對於 entity-Concept-Score，此分數是 Amazon Comprehend Medical 對實體準確連結到 ICD-10-CM 概念的可信度。

下列截斷的 JSON 回應僅包含 modifierExtension元素。

```
"modifierExtension": [{
  "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-score",
  "valueDecimal": 0.94417894
},
{
  "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-Concept-Score",
  "valueDecimal": 0.8458298
},
{
  "url": "http://healthlake.amazonaws.com/system-generated-linkage",
  "valueReference": {
    "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
  }
},
{
  "url": "http://healthlake.amazonaws.com/source-document-reference",
  "valueReference": {
    "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
  }
}
]
```

完整 JSON 回應

```
{
  "subject": {
    "reference": "Patient/0679b7b7-937d-488a-b48d-6315b8e7003b"
  },
  "resourceType": "Condition",
  "code": {
    "coding": [{
      "system": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/",
      "code": "I70.0",
      "display": "Atherosclerosis of aorta"
    }],
    "text": "Atherosclerosis of aorta"
  },
  "meta": {
    "lastUpdated": "2022-10-21T19:38:30.877Z",
    "tag": [{
      "display": "SYSTEM_GENERATED"
    }]
  },
  "modifierExtension": [{
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-score",
    "valueDecimal": 0.94417894
  },
  {
    "url": "http://healthlake.amazonaws.com/aws-cm/infer-icd10/aws-cm-icd10-entity-Concept-Score",
    "valueDecimal": 0.8458298
  },
  {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
      "reference": "Linkage/e366d29f-2c22-4c19-866e-09603937935a"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/source-document-reference",
    "valueReference": {
      "reference": "DocumentReference/0e938f03-da7f-4178-acd8-eea9586c46ed"
    }
  }
],
```

```
"id": "b06d343d-ddb8-4f36-82cb-853fcd434dfd"
}
```

範例 2：DocumentReference 包含 MedicationStatement 資源類型的

以下是根據病患與醫療專業人員接觸的臨床備註範例。

⚠ 合成資料

此範例中的文字是合成內容，不包含受保護的健康資訊 (PHI)。

```
Tom is not prescribed Advil
```

下列索引標籤顯示如何根據資源類型，在您的 HealthLake 資料存放區中報告擷取的醫療記錄。

DocumentReference

若要查看單一 DocumentReference 資源類型的結果，請提出 GET 請求，其中提供特定資源 ID 的。

```
GET https://https://healthlake.region.amazonaws.com/datastore/datastoreId/
r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/DocumentReference/c549125d-a218-421f-
b8bf-23614c5e796c
```

成功時，您會收到 200 HTTP 回應代碼和下列截斷的 JSON 回應。

鍵/值對 "url": "http://healthlake.amazonaws.com/system-generated-resources/" 表示其中的資源類型 extension 已由 Amazon Comprehend Medical API 操作新增。您可以查看新的 Linkage 資源類型和多個 MedicationStatement 資源。

```
"extension": [{
  "extension": [{
    "url": "http://healthlake.amazonaws.com/linkage",
    "valueReference": {
      "reference": "Linkage/394bb244-177b-4409-8657-26b20ed56dd7"
    }
  }],
  {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
```

```

    "valueReference": {
      "reference": "MedicationStatement/cbf6af10-b0b9-451c-bdde-99611e3498a8"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
      "reference": "MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
      "reference": "MedicationStatement/4a01f6c8-5f3a-4122-80ab-405312f96aa2"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
      "reference": "MedicationStatement/fbfb77d8-70cf-4579-b4c0-d6fe3c01656b"
    }
  },
  {
    "url": "http://healthlake.amazonaws.com/nlp-entity",
    "valueReference": {
      "reference": "MedicationStatement/1340c9ce-9c48-4bf9-9b2f-d0ab027f5e0b"
    }
  }
],
"url": "http://healthlake.amazonaws.com/system-generated-resources/"
}

```

Linkage

若要查看單一Linkage資源類型的結果，請提出 GET 請求，其中提供特定資源ID的。

```
GET https://https://healthlake.region.amazonaws.com/datastore/datastoreId/
r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/Linkage/394bb244-177b-4409-8657-26b20ed56dd7
```

成功時，您會收到 200 HTTP 回應代碼和下列 JSON 回應。

回應包含 item 元素。其中，鍵/值對 "type": "source" 會指出用來修改 MedicationStatement 資源類型的特定 DocumentReference 項目。

您也可以看到 meta 元素和對應的鍵/值對 "tag": [{"display": "SYSTEM_GENERATED"}]，表示這些資源是由 HealthLake 建立。

```
{
  "resourceType": "Linkage",
  "id": "394bb244-177b-4409-8657-26b20ed56dd7",
  "active": true,
  "item": [{
    "type": "alternate",
    "resource": {
      "reference": "MedicationStatement/cbf6af10-b0b9-451c-bdde-99611e3498a8",
      "type": "MedicationStatement"
    }
  },
  {
    "type": "alternate",
    "resource": {
      "reference": "MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7",
      "type": "MedicationStatement"
    }
  },
  {
    "type": "alternate",
    "resource": {
      "reference": "MedicationStatement/4a01f6c8-5f3a-4122-80ab-405312f96aa2",
      "type": "MedicationStatement"
    }
  },
  {
    "type": "alternate",
    "resource": {
      "reference": "MedicationStatement/fbfb77d8-70cf-4579-b4c0-d6fe3c01656b",
      "type": "MedicationStatement"
    }
  },
  {
    "type": "alternate",
    "resource": {
      "reference": "MedicationStatement/1340c9ce-9c48-4bf9-9b2f-d0ab027f5e0b",
      "type": "MedicationStatement"
    }
  }
],
{
```

```

    "type": "source",
    "resource": {
      "reference": "DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c",
      "type": "DocumentReference"
    }
  ],
  "meta": {
    "lastUpdated": "2022-10-24T20:05:03.501Z",
    "tag": [{
      "display": "SYSTEM_GENERATED"
    }]
  }
}

```

MedicationStatement

若要查看單一MedicationStatement資源類型的結果，請提出 GET 請求，其中提供特定資源ID的。

```

GET https://https://healthlake.region.amazonaws.com/
datastore/datastoreId/r4/eeb8005725ae22b35b4edbd68cf2dfd/r4/
MedicationStatement/9a89b0d3-6681-45ca-9926-27951edce5c7

```

MedicationStatement 資源類型是找到 Amazon Comprehend Medical InferRxNorm API 操作結果的位置。結果會修改為下列元素：medicationCodeableConcept、meta和modifierExtension。

medicationCodeableConcept

類型的元素CodeableConcept。若要進一步了解，請參閱 FHIR R4 文件[CodeableConcept](#)中的。

HealthLake 會附加下列三個鍵值對。

- "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/": 其中 URL 是指特定的 Amazon Comprehend Medical API 操作。在此情況下，InferRxNorm。
- "code": "731533": 其中 731533 是 RxNorm 概念 ID，也稱為 RxCUI。
- "display": "ibuprofen 200 MG Oral Capsule [Advil]": 其中 ibuprofen 200 MG Oral Capsule [Advil]是 RxNorm 概念的描述。

下列截斷的 JSON 回應僅包含 MedicationStatement 元素。

```
"medicationCodeableConcept": {
  "coding": [
    {
      "system": "http://healthlake.amazonaws.com/aws-cm/infer-rxnorm/",
      "code": "731533",
      "display": "ibuprofen 200 MG Oral Capsule [Advil]"
    }
  ]
}
```

meta

meta 元素包含中繼資料，指出 code 元素是否包含 Amazon Comprehend Medical API 操作所新增的詳細資訊。

下列截斷的 JSON 回應僅包含 meta 元素。

```
"meta": {
  "lastUpdated": "2022-10-24T20:05:02.800Z",
  "tag": [
    {
      "display": "SYSTEM_GENERATED"
    }
  ]
}
```

modifierExtension

modifierExtension 元素包含鍵/值對，提供用來 DocumentReference 產生結果和相關 Linkage 資源類型的原始連結。

```
"modifierExtension": [
  {
    "url": "http://healthlake.amazonaws.com/system-generated-linkage",
    "valueReference": {
      "reference": "Linkage/394bb244-177b-4409-8657-26b20ed56dd7"
    }
  },
  {
```

```
"url": "http://healthlake.amazonaws.com/source-document-reference",
"valueReference": {
  "reference": "DocumentReference/c549125d-a218-421f-b8bf-23614c5e796c"
}
}
]
```

使用 Amazon Athena 查詢 HealthLake 資料

在 HealthLake 匯入任務期間，巢狀 FHIR JSON 資料會經歷 ETL 程序，並以 [Apache Iceberg 開放資料表格式](#) 存放，其中每個 FHIR 資源類型都表示為 Athena 中的個別資料表。這可讓使用者使用 SQL 查詢 FHIR 資料，但不必先匯出資料。這很有價值，因為它可讓臨床醫生和科學家查詢 FHIR 資料，以驗證他們的決策或推進他們的研究。如需 Apache Iceberg 資料表如何在 Athena 中運作的詳細資訊，請參閱 [《Athena 使用者指南》中的查詢 Apache Iceberg 資料表](#)。

Note

HealthLake 支援 Athena 中 HealthLake 資料的 FHIR R4 read 互動。如需詳細資訊，請參閱 [閱讀取 FHIR 資源](#)。

本節中的主題說明如何將 HealthLake 資料存放區連線至 Athena、如何使用 SQL 查詢，以及如何將結果與其他 AWS 服務連線以進行進一步分析。

主題

- [Amazon Athena 入門](#)
- [使用 SQL 查詢 HealthLake 資料](#)
- [具有複雜篩選的範例 SQL 查詢](#)

Amazon Athena 入門

若要將 HealthLake 與 Amazon Athena 整合，您必須設定許可。若要這樣做，您將建立 Athena 使用者、群組或角色，並授予他們 HealthLake 資料存放區中 FHIR 資源的存取權。

- [授予使用者、群組或角色對 HealthLake 資料存放區的存取權 \(AWS Lake Formation 主控台\)](#)
- [設定 Athena 帳戶](#)

授予使用者、群組或角色對 HealthLake 資料存放區的存取權 (AWS Lake Formation 主控台)

角色：HealthLake 管理員

HealthLake 管理員角色是 AWS Lake Formation 中的資料湖管理員。他們授予 Lake Formation 中 HealthLake 資料存放區的存取權。

對於每個建立的資料存放區，AWS Lake Formation 主控台中會顯示兩個項目。一個項目是資源連結。資源連結名稱一律以斜體顯示。每個資源連結都會顯示其連結共用資源的名稱和擁有者。對於所有 HealthLake 資料存放區，共用資源擁有者是 HealthLake 服務帳戶。另一個項目是 HealthLake 服務帳戶中的 HealthLake 資料存放區。此程序中的步驟會使用資源連結的資料存放區。

若要進一步了解資源連結，請參閱 [Lake Formation 開發人員指南中的資源連結如何在 Lake Formation 中運作](#)。AWS

若要讓使用者、群組或角色能夠查詢 Athena 中的資料，您必須授予資源資料庫的描述許可。然後，您必須在資料表上授予選取和描述。

步驟 1：授予 HealthLake 資料存放區資源連結資料庫的 DESCRIBE 許可

1. 開啟 AWS Lake Formation 主控台：<https://console.aws.amazon.com/lakeformation/>
2. 在主要導覽列中，選擇資料庫。
3. 在資料庫頁面上，選擇斜體資料存放區名稱旁的選項按鈕。
4. 選擇動作 (▼)。
5. 選擇 Grant (授予)。
6. 在授予資料許可頁面的委託人下，選擇 IAM 使用者或角色。
7. 在 IAM 使用者或角色下，使用向下箭頭 (▼)，或搜尋您想要在 Athena 中查詢的 IAM 使用者、角色或群組。
8. 在 LF 標籤或目錄資源卡下，選擇具名資料目錄資源選項。
9. 在資料庫下，使用向下箭頭 (▼) 選擇您要共用存取權的 HealthLake 資料存放區資料庫。
10. 在資源連結許可卡片的資源連結許可下，選擇描述。

當授予成功時，授予許可成功橫幅隨即出現。若要檢視您剛授予的許可，請選擇資料湖許可。在資料表中尋找使用者、群組和角色。在許可欄下，您會看到描述清單。

現在，您必須使用目標上的授予來授予資料庫中所有資料表的選取和描述。

步驟 2：授予 HealthLake 資料存放區資源連結中所有資料表的存取權

1. 開啟 AWS Lake Formation 主控台：<https://console.aws.amazon.com/lakeformation/>
2. 在主要導覽列中，選擇資料庫。
3. 在資料庫頁面上，選擇斜體資料存放區名稱旁的選項按鈕。
4. 選擇動作 (▼)。
5. 選擇對目標授予。
6. 在授予資料許可頁面的委託人下，選擇 IAM 使用者或角色。
7. 在 IAM 使用者或角色下，使用向下箭頭 (▼) 或搜尋您想要在 Athena 中查詢的 IAM 使用者、群組或角色。
8. 在 LF 標籤或目錄資源卡片下，選擇具名資料目錄資源選項。
9. 在資料庫下，使用向下箭頭 (▼) 選擇您要授予存取權的 HealthLake 資料存放區資料庫。
10. 在資料表下，選擇所有資料表以與 HealthLake 使用者共用所有資料表。
11. 在資料表許可卡的資料表許可下，選擇描述和選取。
12. 選擇 Grant (授予)。

選擇授予後，會出現授予許可成功橫幅。指定的使用者現在可以在 Athena 中的 HealthLake 資料存放區上進行查詢。

Athena 入門

HealthLake 使用者

HealthLake 使用者將使用 Athena 主控台 AWS CLI，或 AWS SDKs 來查詢 HealthLake 管理員與其共用的 HealthLake 資料存放區。

若要使用 Athena 查詢資料存放區，您必須執行下列三件事。

- 透過 Lake Formation 授予 IAM 使用者或角色對 HealthLake 資料存放區的存取權。如需詳細資訊，請參閱 [授予使用者、群組或角色對 HealthLake 資料存放區的存取權 \(AWS Lake Formation 主控台\)](#)。
- 為您的 HealthLake 資料存放區建立工作群組。

- 指定 Amazon S3 儲存貯體來存放查詢結果。

若要開始使用 Athena，請將 AmazonAthenaFullAccess 和 AmazonS3FullAccess AWS 受管政策新增至您的使用者、群組或角色。使用 AWS 受管政策是開始使用新服務的好方法。請記住，AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為它們可供所有 AWS 客戶使用。設定 IAM 政策的許可時，請僅授予執行任務所需的許可。若要進一步了解 IAM 並套用最低權限，請參閱《IAM 使用者指南》中的[套用最低權限許可](#)。

Important

若要查詢 Athena 中的 HealthLake 資料存放區，您必須使用 Athena 引擎第 3 版。

工作群組是資源，因此您可以使用 IAM 型政策來控制對特定工作群組的存取。若要進一步了解，請參閱《Athena 使用者指南》中的[使用工作群組來控制查詢存取和成本](#)。

若要進一步了解設定工作群組，請參閱《Athena 使用者指南<https://docs.aws.amazon.com/athena/latest/ug/workgroups-procedure.html>》中的。

Note

Amazon S3 儲存貯體所在的區域，且 Athena 主控台必須相符。

您必須指定 Amazon S3 中的查詢結果儲存貯體位置，然後才能執行查詢，否則您必須使用已指定儲存貯體，且其組態可覆寫用戶端設定的工作群組。每個執行的查詢都會自動儲存輸出檔案。

如需在 Athena 主控台中指定查詢結果位置的詳細資訊，請參閱《Amazon Athena [使用者指南](#)》中的[使用 Athena 主控台指定查詢結果位置](#)。

若要查看如何在 Athena 中查詢 HealthLake 資料存放區的範例，請參閱[使用 SQL 查詢 HealthLake 資料](#)。

使用 SQL 查詢 HealthLake 資料

當您將 FHIR 資料匯入 HealthLake 資料存放區時，巢狀 JSON FHIR 資料會同時進行 ETL 程序，並以 Apache Iceberg 開放資料表格式儲存在 Amazon S3 中。HealthLake 資料存放區中的每個 FHIR 資源類型都會轉換為資料表，可使用 Amazon Athena 來查詢。資料表可以個別查詢或使用 SQL 型查詢做為群組。由於資料存放區的結構，您的資料會以多種不同的資料類型匯入 Athena。若要進一步了解如

何建立可存取這些資料類型的 SQL 查詢，請參閱《Amazon Athena 使用者指南》中的[查詢具有複雜類型和巢狀結構的陣列](#)。

Note

本主題中的所有範例都使用使用 Synthea 建立的虛構資料。若要進一步了解如何建立預先載入 Synthea 資料的資料存放區，請參閱 [建立 HealthLake 資料存放區](#)。

對於資源類型中的每個元素，FHIR 規格會定義基數。元素的基數會定義此元素可顯示多少次的下限和上限。建構 SQL 查詢時，您必須將此納入考量。例如，讓我們看看[資源類型：病患](#)中的一些元素。

- 元素：名稱 FHIR 規格會將基數設定為 0..*。

元素會擷取為陣列。

```
[{
  id = null,
  extension = null,
  use = official,
  _use = null,
  text = null,
  _text = null,
  family = Wolf938,
  _family = null,
  given = [Noel608],
  _given = null,
  prefix = null,
  _prefix = null,
  suffix = null,
  _suffix = null,
  period = null
}]
```

在 Athena 中，若要查看如何擷取資源類型，請在資料表和檢視下搜尋它。若要存取此陣列中的元素，您可以使用點表示法。以下是存取 given 和 值的簡單範例 family。

```
SELECT
  name[1].given as FirstName,
  name[1].family as LastName
FROM Patient
```

- 元素：MaritalStatus FHIR 規格會將基數設定為 0..1。

此元素會擷取為 JSON。

```
{
  id = null,
  extension = null,
  coding = [
    {
      id = null,
      extension = null,
      system = http://terminology.hl7.org/CodeSystem/v3-MaritalStatus,
      _system = null,
      version = null,
      _version = null,
      code = S,
      _code = null,
      display = Never Married,
      _display = null,
      userSelected = null,
      _userSelected = null
    }
  ],
  text = Never Married,
  _text = null
}
```

在 Athena 中，若要查看如何擷取資源類型，請在資料表和檢視下搜尋它。若要存取 JSON 中的鍵/值對，您可以使用點表示法。由於這不是陣列，因此不需要陣列索引。以下是存取值的簡單範例text。

```
SELECT
    maritalstatus.text as MaritalStatus
FROM Patient
```

若要進一步了解如何存取和搜尋 JSON，請參閱 Athena 使用者指南中的[查詢 JSON](#)。

Athena Data Manipulation Language (DML) 查詢陳述式是以 Trino 為基礎。Athena 不支援 Trino 的所有功能，而且有很大的差異。若要進一步了解，請參閱《Amazon Athena 使用者指南》中的[DML 查詢、函數和運算子](#)。

此外，Athena 支援您在建立 HealthLake 資料存放區查詢時可能遇到的多種資料類型。若要進一步了解 Athena 中的資料類型，請參閱 [《Amazon Athena 使用者指南》中的 Amazon Athena 中的資料類型](#)。Amazon Athena

若要進一步了解 SQL 查詢如何在 Athena 中運作，請參閱 [《Amazon Athena 使用者指南》中的 Amazon Athena 的 SQL 參考](#)。Amazon Athena

每個索引標籤顯示如何使用 Athena 搜尋指定資源類型和相關聯元素的範例。

Element: Extension

元素extension用於在資料存放區中建立自訂欄位。

此範例說明如何存取 Patient 資源類型中找到的 extension元素功能。

當您的 HealthLake 資料存放區匯入 Athena 時，資源類型的元素會以不同的方式剖析。由於的結構element是可變的，因此無法在結構描述中完整指定。為了處理該變異性，陣列內的元素會以字串形式傳遞。

在的資料表描述中Patient，您可以看到extension描述為的元素array<string>，這表示您可以使用索引值來存取陣列的元素。不過，若要存取字串的元素，您必須使用 json_extract。

以下是在患者資料表中找到的 extension元素的單一項目。

```
[{
  "valueString": "Kerry175 Cummerata161",
  "url": "http://hl7.org/fhir/StructureDefinition/patient-mothersMaidenName"
},
{
  "valueAddress": {
    "country": "DE",
    "city": "Hamburg",
    "state": "Hamburg"
  },
  "url": "http://hl7.org/fhir/StructureDefinition/patient-birthPlace"
},
{
  "valueDecimal": 0.0,
  "url": "http://synthetichealth.github.io/synthea/disability-adjusted-life-years"
},
{
  "valueDecimal": 5.0,
  "url": "http://synthetichealth.github.io/synthea/quality-adjusted-life-years"
}
```

```
}  
]
```

即使這是有效的 JSON，Athena 仍會將其視為字串。

此 SQL 查詢範例示範如何建立包含 patient-mothersMaidenName 和 patient-birthPlace 元素的資料表。若要存取這些元素，您需要使用不同的陣列索引和 json_extract。

```
SELECT  
  extension[1],  
  json_extract(extension[1], '$.valueString') AS MothersMaidenName,  
  extension[2],  
  json_extract(extension[2], '$.valueAddress.city') AS birthPlace  
FROM patient
```

若要進一步了解涉及 JSON 的查詢，請參閱《Amazon Athena 使用者指南》中的[從 JSON 擷取資料](#)。

Element: birthDate (Age)

年齡不是 FHIR 中病患資源類型的元素。以下是根據年齡篩選的搜尋的兩個範例。

由於年齡不是元素，因此我們會將 birthDate 用於 SQL 查詢。若要查看元素如何導入 FHIR，請在資料表和檢視下搜尋資料表名稱。您可以看到它是類型字串。

範例 1：計算存留期的值

在此範例 SQL 查詢中，我們使用內建的 SQL 工具和 current_dateyear 來擷取這些元件。然後，我們將它們減去，以傳回患者的實際年齡作為稱為的資料欄 age。

```
SELECT  
  (year(current_date) - year(date(birthdate))) as age  
FROM patient
```

範例 2：篩選之前出生且 2019-01-01 為的患者 male。

SQL 查詢說明如何使用 CAST 函數將 birthDate 元素轉換為類型 DATE，以及如何根據 WHERE 子句中的兩個條件進行篩選。由於元素預設會擷取為類型字串，因此我們必須 CAST 將其擷取為類型 DATE。然後，您可以使用 < 運算子將其與不同的日期進行比較 2019-01-01。透過使用 AND，您可以將第二個條件新增至 WHERE 子句。

```
SELECT birthdate
```

```
FROM patient
-- we convert birthdate (varchar) to date > cast that as date too
WHERE CAST(birthdate AS DATE) < CAST('2019-01-01' AS DATE) AND gender = 'male'
```

Resource type: Location

此範例顯示搜尋城市名稱為 Attleboro 的位置資源類型內的位置。

```
SELECT *
FROM Location
WHERE address.city='ATTLEBORO'
LIMIT 10;
```

Element: Age

```
SELECT birthdate
FROM patient
-- we convert birthdate (varchar) to date > cast that as date too
WHERE CAST(birthdate AS DATE) < CAST('2019-01-01' AS DATE) AND gender = 'male'
```

Resource type: Condition

資源類型條件會存放與問題相關的診斷資料，這些問題已上升到關注層級。HealthLake 的整合醫療自然語言處理 (NLP) 會根據 DocumentReference Condition 資源類型中找到的詳細資訊產生新資源。產生新資源時，HealthLake 會將標籤附加SYSTEM_GENERATED至 meta 元素。此 SQL 查詢範例示範如何搜尋條件資料表並傳回結果，SYSTEM_GENERATED 其中的結果已移除。

若要進一步了解 HealthLake 的整合自然語言處理 (NLP)，請參閱 [HealthLake 的整合式自然語言處理 \(NLP\)](#)。

```
SELECT *
FROM condition
WHERE meta.tag[1] is NULL
```

您也可以在指定的字串元素內搜尋，進一步篩選查詢。modifierextension 元素包含有關哪些 DocumentReference 資源用於產生一組條件的詳細資訊。同樣地，您必須使用 json_extract 來存取做為字串帶入 Athena 的巢狀 JSON 元素。

此範例 SQL 查詢示範如何搜尋根據特定 Condition 產生的所有 DocumentReference。使用將 JSON 元素 CAST 設定為字串，以便您可以使用 LIKE 進行比較。

```
SELECT
    meta.tag[1].display as SystemGenerated,
    json_extract(modifierextension[4], '$.valueReference.reference') as
    DocumentReference
FROM condition
WHERE meta.tag[1].display = 'SYSTEM_GENERATED'

AND CAST(json_extract(modifierextension[4], '$.valueReference.reference') as
    VARCHAR) LIKE '%DocumentReference/67aa0278-8111-40d0-8adc-43055eb9d18d%'
```

Resource type: Observation

資源類型 Observation 會存放對病患、裝置或其他主體所做的測量和簡單聲明。HealthLake 的整合自然語言處理 (NLP) 會根據在 Observation 資源中找到的詳細資訊產生新 DocumentReference 資源。此範例 SQL 查詢包含 WHERE meta.tag[1] is NULL 註解，這表示會包含 SYSTEM_GENERATED 結果。

```
SELECT valueCodeableConcept.coding[1].code
FROM Observation
WHERE valueCodeableConcept.coding[1].code = '266919005'
-- WHERE meta.tag[1] is NULL
```

此欄已匯入為 [struct](#)。因此，您可以使用點符號存取其中的元素。

Resource type: MedicationStatement

MedicationStatement 是一種 FHIR 資源類型，可用來存放患者已使用、正在使用或未來將使用之藥物的詳細資訊。HealthLake 的整合醫療自然語言處理 (NLP) 會根據 DocumentReference 資源類型中找到的文件產生新的 MedicationStatement 資源。產生新資源時，HealthLake 會將標籤附加 SYSTEM_GENERATED 至 meta 元素。此範例 SQL 查詢示範如何建立查詢，使用其識別符根據單一病患進行篩選，並尋找 HealthLake 整合 NLP 已新增的資源。

```
SELECT *
FROM medicationstatement
WHERE meta.tag[1].display = 'SYSTEM_GENERATED' AND subject.reference =
    'Patient/0679b7b7-937d-488a-b48d-6315b8e7003b';
```

若要進一步了解 HealthLake 的整合自然語言處理 (NLP)，請參閱 [HealthLake 的整合式自然語言處理 \(NLP\)](#)。

具有複雜篩選的範例 SQL 查詢

下列範例示範如何使用具有複雜篩選的 Amazon Athena SQL 查詢，從 HealthLake 資料存放區尋找 FHIR 資料。

Example 根據人口統計資料建立篩選條件

建立病患群組時，識別正確的病患人口統計特性非常重要。此範例查詢示範如何使用 Trino 點表示法和 `json_extract` 來篩選 HealthLake 資料存放區中的資料。

```
SELECT
  id
  , CONCAT(name[1].family, ' ', name[1].given[1]) as name
  , (year(current_date) - year(date(birthdate))) as age
  , gender as gender
  , json_extract(extension[1], '$.valueString') as MothersMaidenName
  , json_extract(extension[2], '$.valueAddress.city') as birthPlace
  , maritalstatus.coding[1].display as maritalstatus
  , address[1].line[1] as addressline
  , address[1].city as city
  , address[1].district as district
  , address[1].state as state
  , address[1].postalcode as postalcode
  , address[1].country as country
  , json_extract(address[1].extension[1], '$.extension[0].valueDecimal') as latitude
  , json_extract(address[1].extension[1], '$.extension[1].valueDecimal') as longitude
  , telecom[1].value as telNumber
  , deceasedboolean as deceasedIndicator
  , deceaseddatetime
FROM database.patient;
```

您可以使用 Athena 主控台進一步排序和下載結果。

Example 為病患及其相關條件建立篩選條件

下列範例查詢示範如何尋找和排序 HealthLake 資料存放區中找到之病患的所有相關條件。

```
SELECT
  patient.id as patientId
  , condition.id as conditionId
  , CONCAT(name[1].family, ' ', name[1].given[1]) as name
  , condition.meta.tag[1].display
```

```

, json_extract(condition.modifierextension[1], '$.valueDecimal') AS confidenceScore
, category[1].coding[1].code as categoryCode
, category[1].coding[1].display as categoryDescription
, code.coding[1].code as diagnosisCode
, code.coding[1].display as diagnosisDescription
, onsetdatetime
, severity.coding[1].code as severityCode
, severity.coding[1].display as severityDescription
, verificationstatus.coding[1].display as verificationStatus
, clinicalstatus.coding[1].display as clinicalStatus
, encounter.reference as encounterId
, encounter.type as encountertype
FROM database.patient, condition
WHERE CONCAT('Patient/', patient.id) = condition.subject.reference
ORDER BY name;

```

您可以使用 Athena 主控台進一步排序結果，或下載結果以進行進一步分析。

Example 為患者及其相關觀察建立篩選條件

下列範例查詢示範如何尋找和排序 HealthLake 資料存放區中找到之病患的所有相關觀察。

```

SELECT
  patient.id as patientId
  , observation.id as observationId
  , CONCAT(name[1].family, ' ', name[1].given[1]) as name
  , meta.tag[1].display
  , json_extract(modifierextension[1], '$.valueDecimal') AS confidenceScore
  , status
  , category[1].coding[1].code as categoryCode
  , category[1].coding[1].display as categoryDescription
  , code.coding[1].code as observationCode
  , code.coding[1].display as observationDescription
  , effectivedatetime
  , CASE
    WHEN valuequantity.value IS NOT NULL THEN CONCAT(CAST(valuequantity.value AS
    VARCHAR),' ',valuequantity.unit)
      WHEN valueCodeableConcept.coding [ 1 ].code IS NOT NULL THEN
    CAST(valueCodeableConcept.coding [ 1 ].code AS VARCHAR)
      WHEN valuestring IS NOT NULL THEN CAST(valuestring AS VARCHAR)
      WHEN valueboolean IS NOT NULL THEN CAST(valueboolean AS VARCHAR)
      WHEN valueinteger IS NOT NULL THEN CAST(valueinteger AS VARCHAR)
      WHEN valueratio IS NOT NULL THEN CONCAT(CAST(valueratio.numerator.value AS
    VARCHAR),'/',CAST(valueratio.denominator.value AS VARCHAR))
  
```

```

    WHEN valuerange IS NOT NULL THEN CONCAT(CAST(valuerange.low.value AS
VARCHAR), '-', CAST(valuerange.high.value AS VARCHAR))
    WHEN valueSampledData IS NOT NULL THEN CAST(valueSampledData.data AS VARCHAR)
    WHEN valueTime IS NOT NULL THEN CAST(valueTime AS VARCHAR)
    WHEN valueDateTime IS NOT NULL THEN CAST(valueDateTime AS VARCHAR)
    WHEN valuePeriod IS NOT NULL THEN valuePeriod.start
    WHEN component[1] IS NOT NULL THEN CONCAT(CAST(component[2].valuequantity.value
AS VARCHAR), ' ', CAST(component[2].valuequantity.unit AS VARCHAR),
 '/', CAST(component[1].valuequantity.value AS VARCHAR), '
 ', CAST(component[1].valuequantity.unit AS VARCHAR))
    END AS observationvalue
, encounter.reference as encounterId
, encounter.type as encountertype
FROM database.patient, observation
WHERE CONCAT('Patient/', patient.id) = observation.subject.reference
ORDER BY name;

```

Example 為病患及其相關程序建立篩選條件

將程序連接到病患是醫療保健的重要層面。下列 SQL 範例查詢示範如何使用 FHIR Procedure Patient 和資源類型來完成此操作。下列 SQL 查詢將傳回 HealthLake 資料存放區中找到的所有患者及其相關程序。

```

SELECT
patient.id as patientId
, PROCEDURE.id as procedureId
, CONCAT(name[1].family, ' ', name[1].given[1]) as name
, status
, category.coding[1].code as categoryCode
, category.coding[1].display as categoryDescription
, code.coding[1].code as procedureCode
, code.coding[1].display as procedureDescription
, performeddatetime
, performer[1]
, encounter.reference as encounterId
, encounter.type as encountertype
FROM database.patient, procedure
WHERE CONCAT('Patient/', patient.id) = procedure.subject.reference
ORDER BY name;

```

您可以使用 Athena 主控台下載結果以進行進一步分析，或對其進行排序，以進一步了解結果。

Example 為病患及其相關處方建立篩選條件

查看病患目前正在使用之藥物的目前清單非常重要。使用 Athena，您可以撰寫 SQL 查詢，使用 HealthLake 資料存放區中找到的 MedicationRequest Patient 和資源類型。

下列 SQL 查詢會聯結匯入 Athena 的 Patient 和 MedicationRequest 資料表。它也會使用點表示法，將處方組織到其個別項目中。

```
SELECT
  patient.id as patientId
  , medicationrequest.id as medicationrequestid
  , CONCAT(name[1].family, ' ', name[1].given[1]) as name
  , status
  , statusreason.coding[1].code as categoryCode
  , statusreason.coding[1].display as categoryDescription
  , category[1].coding[1].code as categoryCode
  , category[1].coding[1].display as categoryDescription
  , priority
  , donotperform
  , encounter.reference as encounterId
  , encounter.type as encountertype
  , medicationcodeableconcept.coding[1].code as medicationCode
  , medicationcodeableconcept.coding[1].display as medicationDescription
  , dosageinstruction[1].text as dosage
FROM database.patient, medicationrequest
WHERE CONCAT('Patient/', patient.id ) = medicationrequest.subject.reference
ORDER BY name
```

您可以使用 Athena 主控台來排序結果，或下載結果以進行進一步分析。

Example 查看 MedicationStatement 資源類型中找到的藥品

下列範例查詢說明如何使用 SQL 組織匯入 Athena 的巢狀 JSON。查詢使用 FHIR meta 元素來指出 HealthLake 整合自然語言處理 (NLP) 新增藥物的時間。它也會使用 json_extract 來搜尋 JSON 字串陣列內的資料。如需詳細資訊，請參閱 [自然語言處理](#)。

```
SELECT
  medicationcodeableconcept.coding[1].code as medicationCode
  , medicationcodeableconcept.coding[1].display as medicationDescription
  , meta.tag[1].display
  , json_extract(modifierextension[1], '$.valueDecimal') AS confidenceScore
FROM medicationstatement;
```

您可以使用 Athena 主控台下載或排序這些結果。

Example 針對特定疾病類型進行篩選

此範例示範如何找到一組 18 到 75 歲且已診斷出糖尿病的患者。

```
SELECT patient.id as patientId,
       condition.id as conditionId,
       CONCAT(name [ 1 ].family, ' ', name [ 1 ].given [ 1 ]) as name,
       (year(current_date) - year(date(birthdate))) AS age,
       CASE
         WHEN condition.encounter.reference IS NOT NULL THEN condition.encounter.reference
         WHEN observation.encounter.reference IS NOT NULL THEN observation.encounter.reference
       END as encounterId,
       CASE
         WHEN condition.encounter.type IS NOT NULL THEN observation.encounter.type
         WHEN observation.encounter.type IS NOT NULL THEN observation.encounter.type
       END AS encountertype,
       condition.code.coding [ 1 ].code as diagnosisCode,
       condition.code.coding [ 1 ].display as diagnosisDescription,
       observation.category [ 1 ].coding [ 1 ].code as categoryCode,
       observation.category [ 1 ].coding [ 1 ].display as categoryDescription,
       observation.code.coding [ 1 ].code as observationCode,
       observation.code.coding [ 1 ].display as observationDescription,
       effectivedatetime AS observationDateTime,
       CASE
         WHEN valuequantity.value IS NOT NULL THEN CONCAT(CAST(valuequantity.value AS
           VARCHAR),' ',valuequantity.unit)
         WHEN valueCodeableConcept.coding [ 1 ].code IS NOT NULL THEN
           CAST(valueCodeableConcept.coding [ 1 ].code AS VARCHAR)
         WHEN valuestring IS NOT NULL THEN CAST(valuestring AS VARCHAR)
         WHEN valueboolean IS NOT NULL THEN CAST(valueboolean AS VARCHAR)
         WHEN valueinteger IS NOT NULL THEN CAST(valueinteger AS VARCHAR)
         WHEN valueratio IS NOT NULL THEN CONCAT(CAST(valueratio.numerator.value AS
           VARCHAR),'/',CAST(valueratio.denominator.value AS VARCHAR))
         WHEN valuerange IS NOT NULL THEN CONCAT(CAST(valuerange.low.value AS
           VARCHAR),'-',CAST(valuerange.high.value AS VARCHAR))
         WHEN valueSampledData IS NOT NULL THEN CAST(valueSampledData.data AS VARCHAR)
         WHEN valueTime IS NOT NULL THEN CAST(valueTime AS VARCHAR)
         WHEN valueDateTime IS NOT NULL THEN CAST(valueDateTime AS VARCHAR)
         WHEN valuePeriod IS NOT NULL THEN valuePeriod.start
         WHEN component[1] IS NOT NULL THEN CONCAT(CAST(component[2].valuequantity.value
           AS VARCHAR),' ',CAST(component[2].valuequantity.unit AS VARCHAR),
```

```
'/', CAST(component[1].valuequantity.value AS VARCHAR),'  
' ,CAST(component[1].valuequantity.unit AS VARCHAR))  
    END AS observationvalue,  
CASE  
    WHEN condition.meta.tag [ 1 ].display = 'SYSTEM GENERATED' THEN 'YES'  
    WHEN condition.meta.tag [ 1 ].display IS NULL THEN 'NO'  
    WHEN observation.meta.tag [ 1 ].display = 'SYSTEM GENERATED' THEN 'YES'  
    WHEN observation.meta.tag [ 1 ].display IS NULL THEN 'NO'  
    END AS IsSystemGenerated,  
CAST(  
    json_extract(  
        condition.modifierextension [ 1 ],  
        '$.valueDecimal'  
    ) AS int  
    ) AS confidenceScore  
FROM database.patient,  
    database.condition,  
    database.observation  
WHERE CONCAT('Patient/', patient.id) = condition.subject.reference  
    AND CONCAT('Patient/', patient.id) = observation.subject.reference  
    AND (year(current_date) - year(date(birthdate))) >= 18  
    AND (year(current_date) - year(date(birthdate))) <= 75  
    AND condition.code.coding [ 1 ].display like ('%diabetes%');
```

現在您可以使用 Athena 主控台來排序結果，或下載結果以進行進一步分析。

監控 AWS HealthLake

監控和記錄是維護的安全性、可靠性、可用性和效能的重要部分 AWS HealthLake。AWS 提供下列服務來監看 HealthLake、在發生錯誤時回報，並適時採取自動動作。

- AWS CloudTrail 會擷取您 AWS 帳戶或代表您的帳戶發出的 API 呼叫和相關事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。您可以識別呼叫的使用者和帳戶 AWS、進行呼叫的來源 IP 地址，以及呼叫的時間。如需詳細資訊，請參閱「[AWS CloudTrail 使用者指南](#)」。
- Amazon CloudWatch AWS 會即時監控您的 AWS 資源和您在 上執行的應用程式。您可以收集和追蹤指標、建立自訂儀板表，以及設定警示，在特定指標達到您指定的閾值時通知您或採取動作。例如，您可以讓 CloudWatch 追蹤 CPU 使用量或其他 Amazon EC2 執行個體指標，並在需要時自動啟動新的執行個體。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。
- Amazon EventBridge 為無伺服器事件匯流排服務，可讓您輕鬆將應用程式與來自各種來源的資料互相連線。EventBridge 可從自己的應用程式、軟體即服務 (SaaS) 應用程式和 AWS 服務提供即時資料串流，然後將該資料路由到 Lambda 等目標。這可讓您監控在服務中發生的事件，並建置事件導向的架構。如需詳細資訊，請參閱 [Amazon EventBridge 使用者指南](#)。

主題

- [使用 記錄 HealthLake API 呼叫 AWS CloudTrail](#)
- [使用 Amazon CloudWatch 監控 HealthLake 指標](#)
- [使用 Amazon EventBridge 監控 HealthLake 事件](#)

使用 記錄 HealthLake API 呼叫 AWS CloudTrail

AWS HealthLake 已與 服務整合 AWS CloudTrail，此服務提供使用者、角色或 HealthLake 中 AWS 服務所採取動作的記錄。CloudTrail 會將 HealthLake 的所有 API 呼叫擷取為事件。擷取的呼叫包括來自 HealthLake 主控台的呼叫，以及對 HealthLake API 操作的程式碼呼叫。如果您建立線索，則可以將 CloudTrail 事件持續交付至 Amazon S3 儲存貯體，包括 HealthLake 的事件。即使您未設定追蹤，依然可以透過 CloudTrail 主控台的事件歷史記錄檢視最新事件。您可以使用 CloudTrail 所收集的資訊，判斷向 HealthLake 提出的請求、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱[AWS CloudTrail 《使用者指南》](#)。

AWS HealthLake CloudTrail 中的資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。在 HealthLake 中發生活動時，該活動會與事件歷史記錄中的其他服務 AWS 事件一起記錄在 CloudTrail 事件中。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱[使用 CloudTrail 事件歷史記錄檢視事件](#)。

若要持續記錄您 AWS 帳戶中的事件，包括 HealthLake 的事件，請建立追蹤。線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有 AWS 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案](#)，以及[從多個帳戶接收 CloudTrail 日誌檔案](#)

CloudTrail 會記錄所有 HealthLake 動作，並記載於 [HealthLake API 參考](#) 和本開發人員指南中，以了解使用 FHIR REST API 執行的動作。例如，對下列動作的呼叫會在 CloudTrail 日誌檔案中產生項目：

- DescribeFHIRImportJob
- DescribeFHIRExportJob
- StartFHIRImportJob
- ListFHIRImportJobs
- StartFHIRExportJob
- ListFHIRExportJobs
- CreateFHIRDatastore
- ListFHIRDatastores
- DeleteFHIRDatastore
- DescribeFHIRDatastore
- UpdateResource
- CreateResource
- DeleteResource
- ReadResource

- GetCapabilities
- SearchWithGet
- SearchWithPost

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 是否使用根或 AWS Identity and Access Management (IAM) 使用者登入資料提出請求。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

了解 AWS HealthLake 日誌檔案項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

以下範例顯示的是展示 CreateFHIRDatastore 動作的 CloudTrail 日誌項目。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ARO0A2B3ZH0ADD20J4AHJX:git
full_access_iam_role580074395690222150",
    "arn": "arn:aws:sts::691207106566:assumed-role/
colossusfrontend_full_access_iam_role/_iam_role580074395690222150",
    "accountId": "AccountID",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ARO0A2B3ZH0ADD20J4AHJX",
        "arn": "arn:aws:iam::691207106566:role/full_access_iam_role",
        "accountId": "AccountID",
        "userName": "full_access_iam_role"
      },
      "webIdFederationData": {
```

```
    },
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2020-11-20T00:08:15Z"
    }
  },
  "eventTime": "2020-11-20T00:08:16Z",
  "eventSource": "healthlake.amazonaws.com",
  "eventName": "CreateFHIRDatastore",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "3.213.247.1",
  "userAgent": "Coral/Netty4",
  "requestParameters": {
    "datastoreName":
"testCreateFHIRDatastore_GBYAZFCLLBLELBSUT0YYFQZRLBLQJNFOYQVRPZB0JAIUUAHICAEAGIWLNVQEYAMSXVWMBLXC",
    "datastoreTypeVersion": "R4",
    "clientToken": "d737ffe0-14dd-44cc-9f0a-fdf59b26c66b"
  },
  "responseElements": {
    "datastoreId": "datastoreID",
    "datastoreArn": "arn:aws:healthlake:us-
east-1:691207106566:datastore/55576c487ff4975262b10d1d65eb4509",
    "datastoreStatus": "CREATING",
    "datastoreEndpoint": "datastore_endpoint/"
  },
  "requestID": "68e62bdd-d2d4-44c1-af69-e6f055a69f99",
  "eventID": "7ef483dc-5dca-469e-823a-7d9e3a7fe924",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "eventCategory": "Management",
  "recipientAccountId": "691207106566"
}
```

使用 Amazon CloudWatch 監控 HealthLake 指標

您可以使用 Amazon CloudWatch 監控 HealthLake，它會收集原始資料並將其處理為可讀且近乎即時的指標。這些統計資料會保留 15 個月，因此您可以使用該歷史資訊，並更清楚 Web 應用程式或服務的效能。您也可以設定留意特定閾值的警示，當滿足這些閾值時傳送通知或採取動作。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

Note

系統會報告所有[原生 HealthLake 動作](#)的指標。

下表列出向 CloudWatch 報告的 HealthLake 指標和維度。每個都會以使用者指定資料範圍的頻率計數顯示。

下列 HealthLake 指標會報告給 CloudWatch。

向 CloudWatch 報告的 HealthLake 指標

指標	Description
呼叫計數	<p>呼叫 APIs 的次數。您可以針對帳戶或指定的資料存放區回報此情況。</p> <p>單位：Count</p> <p>有效統計資料：總和、計數</p> <p>維度：操作、資料存放區 ID、資料存放區類型</p>
成功的請求	<p>成功的 API 請求數量。</p> <p>單位：Count</p> <p>有效的統計資訊：總和、平均</p> <p>維度：操作、資料存放區 ID、資料存放區類型</p>
使用者錯誤	<p>由於使用者錯誤而失敗的請求數量。</p> <p>單位：Count</p> <p>有效的統計資訊：總和、平均</p> <p>維度：操作、資料存放區 ID、資料存放區類型</p>
伺服器錯誤	<p>由於伺服器錯誤而失敗的請求數量。</p> <p>單位：Count</p>

指標	Description
	有效的統計資訊：總和、平均 維度：操作、資料存放區 ID、資料存放區類型
調節的請求	已調節的請求數量。此指標不包含在使用者或伺服器錯誤計數中。 單位：Count 有效的統計資訊：總和、平均 維度：操作、資料存放區 ID、資料存放區類型
延遲	處理使用者請求所需的時間，以毫秒為單位。 單位：毫秒 有效統計資訊：下限、上限、平均數 維度：操作、資料存放區 ID、資料存放區類型

下列 HealthLake 維度會報告給 CloudWatch。

向 CloudWatch 報告的 HealthLake 維度

維度	Description
作業	請求中使用的 API 操作
DataStoreID	請求中使用的資料存放區 ID
DataStoreType	請求中使用的資料存放區類型

您可以使用 AWS 管理主控台、AWS CLI、或 CloudWatch API 取得 HealthLake 的指標。您可以透過其中一個 Amazon AWS 軟體開發套件 (SDKs) 或 CloudWatch API 工具來使用 CloudWatch API。HealthLake 主控台會根據來自 CloudWatch API 的原始資料顯示圖形。

您必須擁有適當的 CloudWatch 許可，才能使用 CloudWatch 監控 HealthLake。如需詳細資訊，請參閱《[Amazon CloudWatch 使用者指南](#)》中的 [Amazon CloudWatch 的身分和存取管理](#)。Amazon CloudWatch

檢視 HealthLake 指標

若要檢視指標 (CloudWatch 主控台)

1. 登入 AWS 管理主控台 並開啟 [CloudWatch 主控台](#)。
2. 選擇指標，選擇所有指標，然後選擇 AWS/HealthLake。
3. 選擇維度、選擇指標名稱，再選擇 Add to graph (新增至圖形)。
4. 選擇日期範圍的值。所選日期範圍的指標計數會顯示在圖形中。

使用 CloudWatch 建立警示

CloudWatch 警示會監看指定期間內的單一指標，並執行一或多個動作：傳送通知至 Amazon Simple Notification Service (SNS) 主題或 Auto Scaling 政策。動作是根據指標在您指定的數個期間內相對於指定閾值的值。當警示變更狀態時，CloudWatch 也可以傳送 SNS 訊息給您。

Note

CloudWatch 警示只有在狀態變更且在您指定的期間持續存在時，才會叫用動作。

若要檢視指標 (CloudWatch 主控台)

1. 登入 [CloudWatch 主控台](#)。
2. 選擇 Alarms (警示)，然後選擇 Create Alarm (建立警示)。
3. 選擇 AWS/HealthLake，然後選擇指標。
4. 對於 Time Range (時間範圍)，選擇要監控的時間範圍，然後選擇 Next (下一步)。
5. 輸入 Name (名稱) 和 Description (描述)。
6. 針對隨時，選擇 \geq ，然後輸入最大值。
7. 如果您希望 CloudWatch 在達到警示狀態時傳送電子郵件，請在動作區段中，針對每當此警示，選擇狀態為 ALARM。針對傳送通知至，選擇郵寄清單，或選擇新增清單並建立新的郵寄清單。
8. 在 Alarm Preview (警示預覽) 區段中預覽警示。如果警示符合您的要求，選擇 Create Alarm (建立警示)。

使用 Amazon EventBridge 監控 HealthLake 事件

Amazon EventBridge 是一種無伺服器服務，該服務使用事件將應用程式元件連接在一起，讓您更輕鬆地建置可擴展的事件驅動型應用程式。EventBridge 的基礎是建立將事件路由到目標的規則。AWS HealthLake 提供對 EventBridge 的持久狀態變更交付。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 [什麼是 Amazon EventBridge？](#)

Note

若要了解如何將 HealthLake 事件傳送至 Amazon EventBridge，請參閱 AWS for Industries 部落格中的 [的 Amazon EventBridge 整合 AWS HealthLake](#)。

主題

- [傳送至 EventBridge 的 HealthLake 事件](#)
- [HealthLake 事件結構](#)

傳送至 EventBridge 的 HealthLake 事件

下表列出傳送至 EventBridge 以進行處理的所有 HealthLake 事件。

HealthLake 事件類型	State
資料存放區事件	
建立資料存放區	CREATING
資料存放區作用中	ACTIVE
刪除資料存放區	DELETING
已刪除資料存放區	DELETED
資料存放區建立失敗	CREATE_FAILED

如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [datastoreStatus](#)。

匯入任務事件

HealthLake 事件類型	State
匯入任務已提交	SUBMITTED
匯入進行中的任務	IN_PROGRESS
匯入任務已完成，但發生錯誤	COMPLETED_WITH_ERRORS
匯入任務已完成	COMPLETED
匯入任務失敗	FAILED

如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [jobStatus](#)。

匯出任務事件

匯出任務已提交	SUBMITTED
匯出進行中的任務	IN_PROGRESS
匯出任務已完成，但發生錯誤	COMPLETED_WITH_ERRORS
匯出任務已完成	COMPLETED
匯出任務失敗	FAILED

如需詳細資訊，請參閱 AWS HealthLake API 參考中的 [jobStatus](#)。

HealthLake 事件結構

HealthLake 事件是具有 JSON 結構的物件，也包含中繼資料詳細資訊。您可以使用中繼資料做為輸入，以重新建立事件或了解詳細資訊。所有相關聯的中繼資料欄位都會列在下表程式碼範例下的資料表中。如需詳細資訊，請參閱《Amazon EventBridge 使用者指南》中的 [AWS 服務事件中繼資料](#)。

Note

若要了解如何將 HealthLake 事件傳送至 Amazon EventBridge，請參閱 AWS for Industries 部落格中的 [的 Amazon EventBridge 整合 AWS HealthLake](#)。

資料存放區事件

Data Store Creating

狀態 - **CREATING**

```
{
  "version": "0",
  "id": "514ad836-bb8a-4523-a10b-fa2756c3bdb0",
  "detail-type": "Data Store Creating",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T08:58:12Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "datastoreName": "your-data-store-name",
    "datastoreTypeVersion": "R4",
    "datastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
    eeb8005725ae22b35b4edbd68cf2dfd/r4/"
  }
}
```

Data Store Active

狀態 - **ACTIVE**

```
{
  "version": "0",
  "id": "d57105bc-0d2d-4009-b34d-453e2567c599",
  "detail-type": "Data Store Active",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T09:16:51Z",
  "region": "us-east-1",
  "resources":
  [
```

```

    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "datastoreName": "your-data-store-name",
    "datastoreTypeVersion": "R4",
    "datastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
    eeb8005725ae22b35b4edbd68cf2dfd/r4/"
  }
}

```

Data Store Deleting

狀態 - **DELETING**

```

{
  "version": "0",
  "id": "d135ee1f-e14a-4730-8766-7b98f822c94a",
  "detail-type": "Data Store Deleting",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T12:44:47Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "datastoreName": "your-data-store-name",
    "datastoreTypeVersion": "R4",
    "datastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
    eeb8005725ae22b35b4edbd68cf2dfd/r4/"
  }
}

```

Data Store Deleted

狀態 - **DELETED**

```

{
  "version": "0",
  "id": "6d880b86-e115-4947-81a9-494db704571a",
  "detail-type": "Data Store Deleted",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-05-12T12:58:03Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "datastoreName": "your-data-store-name",
    "datastoreTypeVersion": "R4",
    "datastoreEndpoint": "https://healthlake.us-east-1.amazonaws.com/datastore/
    eeb8005725ae22b35b4edbd68cf2dfd/r4/"
  }
}

```

資料存放區事件 - 中繼資料描述

名稱	類型	Description
version	string	EventBridge 事件結構描述版本。
id	string	為每個事件產生的第 4 版 UUID。
detail-type	string	正在傳送的事件類型。
source	string	識別產生事件的服務。
account	string	資料存放區擁有者的 12 位數 AWS 帳戶 ID。
time	string	事件發生的時間。

名稱	類型	Description
region	string	識別資料存放區的 AWS 區域。
resources	陣列 (字串)	包含資料存放區 ARN 的 JSON 陣列。
detail	object	包含事件相關資訊的 JSON 物件。
detail.datastoreId	string	與狀態變更事件相關聯的資料存放區 ID。
detail.datastoreName	string	資料存放區名稱。
detail.datastoreTypeVersion	string	資料存放區 FHIR 版本。
detail.datastoreEndpoint	string	資料存放區端點。

匯入任務事件

Import Job Submitted

狀態 - **SUBMITTED**

```
{
  "version": "0",
  "id": "25e606f7-800c-da41-45df-0e68587250c9",
  "detail-type": "Import Job Submitted",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T01:50:51Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd6c68cf2dfd"
  ],
}
```

```

    "detail":
    {
      "jobId": "08c60716d6321710893ff88410e902c2",
      "submitTime": "2023-12-08T01:50:50.986Z",
      "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
      "inputDataConfig":
      {
        "s3Uri": "s3://amzn-s3-demo-source-bucket/input/"
      }
    }
  }
}

```

Import Job In Progress

狀態 - IN_PROGRESS

```

{
  "version": "0",
  "id": "cc886b49-2737-19c4-7c4e-84ac9429ab73",
  "detail-type": "Import Job In Progress",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T01:51:23Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "jobId": "08c60716d6321710893ff88410e902c2",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "inputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/input/"
    }
  }
}

```

Import Job Completed

狀態 - COMPLETED

```
{
  "version": "0",
  "id": "36c865ef-da41-76ef-c882-3ba8dad8656b",
  "detail-type": "Import Job Completed",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
eeb8005725ae22b35b4eddbc68cf2dfd"
  ],
  "detail":
  {
    "jobId": "08c60716d6321710893ff88410e902c2",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4eddbc68cf2dfd",
    "inputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/input/"
    }
  }
}
```

Import Job Completed With Errors

狀態 - **COMPLETED_WITH_ERRORS**

```
{
  "version": "0",
  "id": "b61387d7-bffe-4f01-8291-65dc4be52cc1",
  "detail-type": "Import Job Completed With Errors",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
eeb8005725ae22b35b4eddbc68cf2dfd"
  ],
  "detail":
```

```
{
  "jobId": "08c60716d6321710893ff88410e902c2",
  "submitTime": "2023-12-08T01:50:50.986Z",
  "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
  "inputDataConfig":
  {
    "s3Uri": "s3://amzn-s3-demo-source-bucket/input/"
  }
}
```

Import Job Failed

狀態 - **FAILED**

```
{
  "version": "0",
  "id": "c4d65575-d1a7-4040-9c6c-c225bf6723c5",
  "detail-type": "Import Job Failed",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "jobId": "08c60716d6321710893ff88410e902c2",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "inputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/input/"
    }
  }
}
```

匯入任務事件 - 中繼資料描述

名稱	類型	Description
version	string	EventBridge 事件結構描述版本。
id	string	為每個事件產生的第 4 版 UUID。
detail-type	string	正在傳送的事件類型。
source	string	識別產生事件的服務。
account	string	資料存放區擁有者的 12 位數 AWS 帳戶 ID。
time	string	事件發生的時間。
region	string	識別資料存放區的 AWS 區域。
resources	陣列 (字串)	包含資料存放區 ARN 的 JSON 陣列。
detail	object	包含事件相關資訊的 JSON 物件。
detail.jobId	string	與狀態變更事件相關聯的匯入任務 ID。
detail.submitTime	string	匯入任務提交的時間。
detail.datastoreId	string	產生狀態變更事件的資料存放區。
detail.inputDataConfig	string	Amazon S3 儲存貯體的輸入字首路徑，其中包含要匯入的 FHIR 檔案。

匯出任務事件

Export Job Submitted

狀態 - **SUBMITTED**

```
{
  "version": "0",
  "id": "f8af7d04-2221-4f02-a01a-6fc3ae403bab",
  "detail-type": "Export Job Submitted",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T01:50:51Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4edbd68cf2dfd"
  ],
  "detail":
  {
    "jobId": "45e899e545bf774710388260fc60b143",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
    "outputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/output/"
    }
  }
}
```

Export Job In Progress

狀態 - **IN_PROGRESS**

```
{
  "version": "0",
  "id": "7bb7e39c-707d-4a83-8532-cee015299100",
  "detail-type": "Export Job In Progress",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T01:51:23Z",
  "region": "us-east-1",
```

```
"resources":
[
  "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
eeb8005725ae22b35b4eddbc68cf2dfd"
],
"detail":
{
  "jobId": "45e899e545bf774710388260fc60b143",
  "submitTime": "2023-12-08T01:50:50.986Z",
  "datastoreId": "eeb8005725ae22b35b4eddbc68cf2dfd",
  "outputDataConfig":
  {
    "s3Uri": "s3://amzn-s3-demo-source-bucket/output/"
  }
}
}
```

Export Job Completed

狀態 - COMPLETED

```
{
  "version": "0",
  "id": "d7629aa7-e63a-4b84-858c-96a62b57ebc8",
  "detail-type": "Export Job Completed",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
eeb8005725ae22b35b4eddbc68cf2dfd"
  ],
  "detail":
  {
    "jobId": "45e899e545bf774710388260fc60b143",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4eddbc68cf2dfd",
    "outputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/output/"
    }
  }
}
```

```
}
```

Export Job Completed With Errors

狀態 - **COMPLETED_WITH_ERRORS**

```
{
  "version": "0",
  "id": "5fa50bc5-50e3-4bc4-b66a-1b1d2f7b07a7",
  "detail-type": "Export Job Completed With Errors",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
  "resources":
  [
    "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
    eeb8005725ae22b35b4eddbc68cf2dfd"
  ],
  "detail":
  {
    "jobId": "45e899e545bf774710388260fc60b143",
    "submitTime": "2023-12-08T01:50:50.986Z",
    "datastoreId": "eeb8005725ae22b35b4eddbc68cf2dfd",
    "outputDataConfig":
    {
      "s3Uri": "s3://amzn-s3-demo-source-bucket/output/"
    }
  }
}
```

Export Job Failed

狀態 - **FAILED**

```
{
  "version": "0",
  "id": "49fce45e-7e02-4846-8582-e7f19ca039cb",
  "detail-type": "Export Job Failed",
  "source": "aws.healthlake",
  "account": "123456789012",
  "time": "2023-12-08T02:14:42Z",
  "region": "us-east-1",
```

```

    "resources":
    [
        "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/
eeb8005725ae22b35b4edbd68cf2dfd"
    ],
    "detail":
    {
        "jobId": "45e899e545bf774710388260fc60b143",
        "submitTime": "2023-12-08T01:50:50.986Z",
        "datastoreId": "eeb8005725ae22b35b4edbd68cf2dfd",
        "outputDataConfig":
        {
            "s3Uri": "s3://amzn-s3-demo-source-bucket/output/"
        }
    }
}

```

匯出任務事件 - 中繼資料描述

名稱	類型	Description
version	string	EventBridge 事件結構描述版本。
id	string	為每個事件產生的第 4 版 UUID。
detail-type	string	正在傳送的事件類型。
source	string	識別產生事件的服務。
account	string	資料存放區擁有者的 12 位數 AWS 帳戶 ID。
time	string	事件發生的時間。
region	string	識別資料存放區的 AWS 區域。
resources	陣列 (字串)	包含資料存放區 ARN 的 JSON 陣列。

名稱	類型	Description
detail	object	包含事件相關資訊的 JSON 物件。
detail.jobId	string	與狀態變更事件相關聯的匯出任務 ID。
detail.submitTime	string	匯出任務提交的時間。
detail.datastoreId	string	產生狀態變更事件的資料存放區。
detail.outputDataConfig	string	Amazon S3 儲存貯體的輸出字首路徑，其中包含要匯出的 FHIR 檔案。

安全 in AWS HealthLake

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，該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。[共同責任模式](#)將其描述為雲端的安全性，和雲端中的安全性：

- 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。在[AWS 合規計劃](#)合規中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 HealthLake 的合規計劃，請參閱[合規計劃的 AWS 服務範圍](#)。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 HealthLake 時套用共同責任模型。下列主題說明如何設定 HealthLake 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 HealthLake 資源。

主題

- [資料保護 in AWS HealthLake](#)
- [REST for AWS HealthLake 加密](#)
- [for AWS HealthLake 傳輸中的加密](#)
- [適用於 AWS HealthLake 的身分和存取管理](#)
- [AWS HealthLake 的合規驗證](#)
- [AWS HealthLake 中的基礎設施安全](#)
- [使用 建立 AWS HealthLake 資源 AWS CloudFormation](#)
- [AWS HealthLake 和界面 VPC 端點 \(AWS PrivateLink\)](#)
- [AWS HealthLake 中的安全最佳實務](#)
- [彈性 in AWS HealthLake](#)

資料保護 in AWS HealthLake

[共同責任模型](#)適用於 AWS HealthLake AWS 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](#)。如需歐洲資料保護的相關資訊，請參閱「[一般資料保護法規 \(GDPR\) 中心](#)」。

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 HealthLake 或使用主控台、API AWS CLI、AWS SDKs 的其他 AWS 服務時。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。

REST for AWS HealthLake 加密

HealthLake 預設提供加密，使用服務擁有的 AWS Key Management Service (AWS KMS) 金鑰來保護靜態敏感客戶資料。客戶管理的 KMS 金鑰也受到支援，而且從資料存放區匯入和匯出檔案都需要。若要進一步了解客戶受管 KMS 金鑰，請參閱[Amazon Key Management Service](#)。建立資料存放區時，客戶可以選擇 AWS 擁有的 KMS 金鑰或客戶管理的 KMS 金鑰。建立資料存放區之後，就無法變更加密組態。如果資料存放區使用 AWS 擁有的 KMS 金鑰，則會將其表示為 `AWS_OWNED_KMS_KEY`，而且您不會看到用於靜態加密的特定金鑰。

AWS 擁有的 KMS 金鑰

HealthLake 預設會使用這些金鑰自動加密潛在的敏感資訊，例如個人身分識別或靜態私有健康資訊 (PHI) 資料。AWS 擁有的 KMS 金鑰不會存放在您的帳戶中。它們是 AWS 擁有和管理用於多個 AWS 帳戶的 KMS 金鑰集合的一部分。AWS 服務可以使用 AWS 擁有的 KMS 金鑰來保護您的資料。您無法檢視、管理、使用 AWS 擁有的 KMS 金鑰或稽核其使用方式。不過，您不需要進行任何工作或變更任何程式來保護加密資料的金鑰。

如果您使用 AWS 擁有的 KMS 金鑰，您不需要支付每月費用或使用費，而且這些金鑰不會計入您帳戶的 AWS KMS 配額。如需詳細資訊，請參閱 [AWS 擁有的金鑰](#)。

客戶受管的 KMS 金鑰

HealthLake 支援使用您建立、擁有和管理的對稱客戶受管 KMS 金鑰，以透過現有 AWS 擁有的加密新增第二層加密。您可以完全控制此層加密，因此能執行以下任務：

- 建立和維護金鑰政策、IAM 政策和授予
- 輪換金鑰密碼編譯資料
- 啟用和停用金鑰政策
- 新增 標籤
- 建立金鑰別名
- 安排金鑰供刪除

您也可以使用 CloudTrail 來追蹤 HealthLake AWS KMS 代表您傳送給的請求。AWS KMS 需支付額外費用。如需詳細資訊，請參閱 [客戶擁有的金鑰](#)。

建立客戶自管金鑰

您可以使用 AWS 管理主控台或 AWS KMS APIs 來建立對稱客戶受管金鑰。

請遵循 AWS Key Management Service 開發人員指南中 [建立對稱客戶受管金鑰](#) 的步驟。

金鑰政策會控制客戶受管金鑰的存取權限。每個客戶受管金鑰都必須只有一個金鑰政策，其中包含決定誰可以使用金鑰及其使用方式的陳述式。在建立客戶受管金鑰時，可以指定金鑰政策。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的 [管理對客戶受管金鑰的存取](#)。

若要將客戶受管金鑰與 HealthLake 資源搭配使用，必須在金鑰政策中允許 [kms:CreateGrant](#) 操作。這會將授予新增至客戶受管金鑰，以控制對指定 KMS 金鑰的存取，讓使用者存取 HealthLake 所需的 [kms:grant](#) 操作。如需詳細資訊，請參閱 [使用授予](#)。

若要將客戶受管 KMS 金鑰與 HealthLake 資源搭配使用，必須在金鑰政策中允許下列 API 操作：

- kms:CreateGrant 會將授予新增至特定客戶受管 KMS 金鑰，以允許存取授予操作。
- kms:DescribeKey 提供驗證金鑰所需的客戶受管金鑰詳細資訊。這是所有操作的必要項目。
- kms:GenerateDataKey 提供存取權，以加密所有寫入操作的靜態資源。
- kms:Decrypt 可讓您存取加密資源的讀取或搜尋操作。

以下是政策陳述式範例，允許使用者建立資料存放區 in AWS HealthLake 並與其互動，該資料存放區由該金鑰加密：

```
"Statement": [
  {
    "Sid": "Allow access to create data stores and do CRUD/search in AWS
HealthLake",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:HealthLakeFullAccessRole"
    },
    "Action": [
      "kms:DescribeKey",
      "kms:CreateGrant",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:ViaService": "healthlake.amazonaws.com",
        "kms:CallerAccount": "111122223333"
      }
    }
  }
]
```

使用客戶受管 KMS 金鑰所需的 IAM 許可

使用客戶受管 KMS 金鑰建立啟用 AWS KMS 加密的資料存放區時，建立 HealthLake 資料存放區的使用者或角色需要金鑰政策和 IAM 政策的許可。

您可以使用 [kms:ViaService 條件金鑰](#)，將 KMS 金鑰的使用限制為僅來自 HealthLake 的請求。

如需金鑰政策的詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[啟用 IAM 政策](#)。

建立儲存庫的 IAM 使用者、IAM 角色或 AWS 帳戶必須具有

kms:CreateGrant、kms:GenerateDataKey 和 kms:DescribeKey 許可，以及必要的 HealthLake 許可。

HealthLake 如何在 AWS KMS 中使用授予

HealthLake 需要[授予](#)才能使用客戶受管 KMS 金鑰。當您建立使用客戶受管 KMS 金鑰加密的資料存放區時，HealthLake 會透過傳送 [CreateGrant](#) 請求至 AWS KMS 來代表您建立授予。AWS KMS 中的授予用於授予 HealthLake 存取客戶帳戶中 KMS 金鑰的權限。

HealthLake 代表您建立的授予不應被撤銷或淘汰。如果您撤銷或淘汰授予 HealthLake 在帳戶中使用 AWS KMS 金鑰許可的授予，HealthLake 無法存取此資料、加密推送到資料存放區的新 FHIR 資源，或在提取時解密這些資源。當您撤銷或淘汰 HealthLake 的授予時，變更會立即發生。若要撤銷存取權，您應該刪除資料存放區，而不是撤銷授予。刪除資料存放區時，HealthLake 會代表您淘汰授予。

監控 HealthLake 的加密金鑰

您可以使用 CloudTrail 來追蹤 HealthLake 在使用客戶受管 KMS 金鑰時代表您傳送給 AWS KMS 的請求。CloudTrail 日誌中的日誌項目會在 userAgent 欄位中顯示 healthlake.amazonaws.com，以清楚區分 HealthLake 提出的請求。

下列範例是 CreateGrant、GenerateDataKey、Decrypt 和 DescribeKey 的 CloudTrail 事件，用於監控 HealthLake 呼叫 AWS KMS 的操作，以存取客戶受管金鑰加密的資料。

以下說明如何使用 CreateGrant 來允許 HealthLake 存取客戶提供的 KMS 金鑰，讓 HealthLake 能夠使用該 KMS 金鑰來加密所有靜態客戶資料。

使用者不需要建立自己的授予。HealthLake 會透過傳送 CreateGrant 請求至 AWS KMS 來代表您建立授予。中的授予 AWS KMS 用於授予 HealthLake 存取客戶帳戶中 AWS KMS 金鑰的權限。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLEROLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLEKEYID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLEROLE",
        "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Sampleuser01"
      }
    }
  }
}
```

```
    },
    "webIdFederationData": {},
    "attributes": {
      "creationDate": "2021-06-30T19:33:37Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "healthlake.amazonaws.com"
},
"eventTime": "2021-06-30T20:31:15Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-east-1",
"sourceIPAddress": "healthlake.amazonaws.com",
"userAgent": "healthlake.amazonaws.com",
"requestParameters": {
  "operations": [
    "CreateGrant",
    "Decrypt",
    "DescribeKey",
    "Encrypt",
    "GenerateDataKey",
    "GenerateDataKeyWithoutPlaintext",
    "ReEncryptFrom",
    "ReEncryptTo",
    "RetireGrant"
  ],
  "granteePrincipal": "healthlake.us-east-1.amazonaws.com",
  "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN",
  "retiringPrincipal": "healthlake.us-east-1.amazonaws.com"
},
"responseElements": {
  "grantId": "EXAMPLE_ID_01"
},
"requestID": "EXAMPLE_ID_02",
"eventID": "EXAMPLE_ID_03",
"readOnly": false,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
  }
],
```

```

"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

下列範例示範如何使用 `GenerateDataKey`，確保使用者在儲存資料之前擁有加密資料的必要許可。

```

{
"eventVersion": "1.08",
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "EXAMPLEUSER",
  "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
  "accountId": "111122223333",
  "accessKeyId": "EXAMPLEKEYID",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "EXAMPLEROLE",
      "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
      "accountId": "111122223333",
      "userName": "Sampleuser01"
    },
    "webIdFederationData": {},
    "attributes": {
      "creationDate": "2021-06-30T21:17:06Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "healthlake.amazonaws.com"
},
"eventTime": "2021-06-30T21:17:37Z",
"eventSource": "kms.amazonaws.com",
"eventName": "GenerateDataKey",
"awsRegion": "us-east-1",
"sourceIPAddress": "healthlake.amazonaws.com",
"userAgent": "healthlake.amazonaws.com",
"requestParameters": {
  "keySpec": "AES_256",
  "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
}
}

```

```

    },
    "responseElements": null,
    "requestID": "EXAMPLE_ID_01",
    "eventID": "EXAMPLE_ID_02",
    "readOnly": true,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

下列範例顯示 HealthLake 如何呼叫 Decrypt 操作，以使用存放的加密資料金鑰來存取加密的資料。

```

    {
      "eventVersion": "1.08",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLEUSER",
        "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLEKEYID",
        "sessionContext": {
          "sessionIssuer": {
            "type": "Role",
            "principalId": "EXAMPLEROLE",
            "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
            "accountId": "111122223333",
            "userName": "Sampleuser01"
          },
          "webIdFederationData": {},
          "attributes": {
            "creationDate": "2021-06-30T21:17:06Z",
            "mfaAuthenticated": "false"
          }
        }
      }
    }

```

```

    },
    "invokedBy": "healthlake.amazonaws.com"
  },
  "eventTime": "2021-06-30T21:21:59Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "healthlake.amazonaws.com",
  "userAgent": "healthlake.amazonaws.com",
  "requestParameters": {
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
  },
  "responseElements": null,
  "requestID": "EXAMPLE_ID_01",
  "eventID": "EXAMPLE_ID_02",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

下列範例顯示 HealthLake 如何使用 DescribeKey 操作來驗證 AWS KMS 客戶擁有的 AWS KMS 金鑰是否處於可用狀態，並在使用者無法運作時進行故障診斷。

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EXAMPLEUSER",
    "arn": "arn:aws:sts::111122223333:assumed-role/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "EXAMPLEKEYID",

```

```
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EXAMPLEROLE",
        "arn": "arn:aws:iam::111122223333:role/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Sampleuser01"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2021-07-01T18:36:14Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "healthlake.amazonaws.com"
  },
  "eventTime": "2021-07-01T18:36:36Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "healthlake.amazonaws.com",
  "userAgent": "healthlake.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
  },
  "responseElements": null,
  "requestID": "EXAMPLE_ID_01",
  "eventID": "EXAMPLE_ID_02",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-east-1:111122223333:key/EXAMPLE_KEY_ARN"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}
```

進一步了解

下列資源提供有關靜態資料加密的詳細資訊。

如需 [AWS Key Management Service 基本概念](#) 的詳細資訊，請參閱 AWS KMS 文件。

如需 AWS KMS 文件中 [安全最佳實務](#) 的詳細資訊。

for AWS HealthLake 傳輸中的加密

AWS HealthLake 使用 TLS 1.2 透過公有端點和後端服務加密傳輸中的資料。

適用於 AWS HealthLake 的身分和存取管理

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 HealthLake 資源。IAM 是您可以免費使用 AWS 服務的。

主題

- [目標對象](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [如何使用 IAM 搭配 AWS HealthLake](#)
- [AWS HealthLake 的身分型政策範例](#)
- [AWS HealthLake 的受管政策](#)
- [疑難排解 AWS HealthLake 身分和存取](#)

目標對象

使用方式 AWS Identity and Access Management (IAM) 會根據您的角色而有所不同：

- 服務使用者 — 若無法存取某些功能，請向管理員申請所需許可 (請參閱 [疑難排解 AWS HealthLake 身分和存取](#))
- 服務管理員 — 負責設定使用者存取權並提交相關許可請求 (請參閱 [如何使用 IAM 搭配 AWS HealthLake](#))
- IAM 管理員 — 撰寫政策以管理存取控制 (請參閱 [AWS HealthLake 的身分型政策範例](#))

使用身分驗證

身分驗證是您 AWS 使用身分憑證登入的方式。您必須以 AWS 帳戶根使用者、IAM 使用者或擔任 IAM 角色身分進行身分驗證。

您可以使用身分來源的登入資料，例如 AWS IAM Identity Center (IAM Identity Center)、單一登入身分驗證或 Google/Facebook 登入資料，以聯合身分的形式登入。如需有關登入的詳細資訊，請參閱《AWS 登入 使用者指南》中的[如何登入您的 AWS 帳戶](#)。

對於程式設計存取，AWS 提供 SDK 和 CLI 以密碼編譯方式簽署請求。如需詳細資訊，請參閱《IAM 使用者指南》中的[API 請求的AWS 第 4 版簽署程序](#)。

AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個名為 AWS 帳戶 theroot 使用者的登入身分開始，該身分可完整存取所有 AWS 服務和資源。強烈建議不要使用根使用者來執行日常任務。有關需要根使用者憑證的任務，請參閱《IAM 使用者指南》中的[需要根使用者憑證的任務](#)。

聯合身分

最佳實務是要求人類使用者使用聯合身分提供者，以 AWS 服務使用臨時憑證存取。

聯合身分是您企業目錄、Web 身分提供者的使用者，或使用身分來源的 AWS 服務憑證存取 Directory Service。聯合身分會擔任角色，而該角色會提供臨時憑證。

若需集中化管理存取權限，建議使用 AWS IAM Identity Center。如需詳細資訊，請參閱 AWS IAM Identity Center 使用者指南中的[什麼是 IAM Identity Center ?](#)。

IAM 使用者和群組

IAM 使用者https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html是一種身分具備單人或應用程式的特定許可權。建議以臨時憑證取代具備長期憑證的 IAM 使用者。如需詳細資訊，請參閱《IAM 使用者指南》中的[要求人類使用者使用聯合身分提供者來 AWS 使用臨時憑證存取](#)。

[IAM 群組](#)會指定 IAM 使用者集合，使管理大量使用者的許可權更加輕鬆。如需詳細資訊，請參閱《IAM 使用者指南》中的[IAM 使用者的使用案例](#)。

IAM 角色

IAM 角色https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html的身分具有特定許可權，其可以提供臨時憑證。您可以透過[從使用者切換到 IAM 角色 \(主控台\)](#)或呼叫 AWS CLI 或 AWS API 操作來擔任角色。如需詳細資訊，請參閱《IAM 使用者指南》中的[擔任角色的方法](#)。

IAM 角色適用於聯合身分使用者存取、臨時 IAM 使用者許可、跨帳戶存取權與跨服務存取，以及在 Amazon EC2 執行的應用程式。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](#)。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策定義與身分或資源相關聯的許可。當委託人提出請求時 AWS，會評估這些政策。大多數政策會以 JSON 文件 AWS 的形式存放在中。如需進一步了解 JSON 政策文件，請參閱《IAM 使用者指南》中的 [JSON 政策概觀](#)。

管理員會使用政策，透過定義哪些主體可在哪些條件下對哪些資源執行動作，以指定可存取的範圍。

預設情況下，使用者和角色沒有許可。IAM 管理員會建立 IAM 政策並將其新增至角色，供使用者後續擔任。IAM 政策定義動作的許可，無論採用何種方式執行。

身分型政策

身分型政策是附加至身分 (使用者、使用者群組或角色) 的 JSON 許可政策文件。這類政策控制身分可對哪些資源執行哪些動作，以及適用的條件。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](#)。

身分型政策可分為內嵌政策 (直接內嵌於單一身分) 與受管政策 (可附加至多個身分的獨立政策)。如需了解如何在受管政策及內嵌政策之間做選擇，請參閱《IAM 使用者指南》中的 [在受管政策與內嵌政策之間選擇](#)。

資源型政策

資源型政策是附加到資源的 JSON 政策文件。範例包括 IAM 角色信任政策與 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。您必須在資源型政策中 [指定主體](#)。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

其他政策類型

AWS 支援其他政策類型，可設定更多常見政策類型授予的最大許可：

- 許可界限 — 設定身分型政策可授與 IAM 實體的最大許可。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 實體許可界限](#)。

- 服務控制政策 (SCP) — 為 AWS Organizations 中的組織或組織單位指定最大許可。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[服務控制政策](#)。
- 資源控制政策 (RCP) — 設定您帳戶中資源可用許可的上限。如需詳細資訊，請參閱《AWS Organizations 使用者指南》中的[資源控制政策 \(RCP\)](#)。
- 工作階段政策 — 在以程式設計方式為角色或聯合身分使用者建立臨時工作階段時，以參數形式傳遞的進階政策。如需詳細資訊，請參《IAM 使用者指南》中的[工作階段政策](#)。

多種政策類型

當多種類型的政策適用於請求時，產生的許可會更複雜而無法理解。若要了解如何 AWS 決定是否在涉及多個政策類型時允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

如何使用 IAM 搭配 AWS HealthLake

在您使用 IAM 管理 HealthLake 的存取權之前，請先了解哪些 IAM 功能可與 HealthLake 搭配使用。

您可以搭配 AWS HealthLake 使用的 IAM 功能

IAM 功能	HealthLake 支援
身分型政策	是
資源型政策	否
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
主體許可	是
服務角色	是
服務連結角色	否

若要全面了解 HealthLake 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](#)。

AWS HealthLake 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](#)。

AWS HealthLake 的身分型政策範例

若要檢視 HealthLake 身分型政策的範例，請參閱 [AWS HealthLake 的身分型政策範例](#)。

AWS HealthLake 內的資源型政策

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](#)。

AWS HealthLake 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 HealthLake 動作清單，請參閱服務授權參考中的[AWS HealthLake 定義的動作](#)。

HealthLake 中的政策動作在動作之前使用下列字首：

```
healthlake
```

若要在單一陳述式中指定多個動作，請以逗號分隔每個動作。

```
"Action": [  
  "healthlake:action1",  
  "healthlake:action2"  
]
```

若要檢視 HealthLake 身分型政策的範例，請參閱 [AWS HealthLake 的身分型政策範例](#)。

AWS HealthLake 的政策資源

支援政策資源：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。

Resource JSON 政策元素可指定要套用動作的物件。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。若動作不支援資源層級許可，使用萬用字元 (*) 表示該陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 HealthLake 資源類型及其 ARNs 的清單，請參閱《服務授權參考》中的[AWS HealthLake 定義的資源](#)。若要了解您可以使用哪些動作來指定每個資源的 ARN，請參閱[AWS HealthLake 定義的動作](#)。

若要檢視 HealthLake 身分型政策的範例，請參閱 [AWS HealthLake 的身分型政策範例](#)。

AWS HealthLake 的政策條件索引鍵

支援服務特定政策條件金鑰：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。

Condition 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的[AWS 全域條件內容索引鍵](#)。

若要查看 HealthLake 條件索引鍵的清單，請參閱《服務授權參考》中的[AWS HealthLake 的條件索引鍵](#)。若要了解您可以使用條件金鑰的動作和資源，請參閱[AWS HealthLake 定義的動作](#)。

若要檢視 HealthLake 身分型政策的範例，請參閱 [AWS HealthLake 的身分型政策範例](#)。

存取控制清單 (ACLs) in AWS HealthLake

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。

使用 AWS HealthLake 的屬性型存取控制 (ABAC)

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的[條件元素](#)中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的[使用 ABAC 授權定義許可](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的[使用屬性型存取控制 \(ABAC\)](#)。

搭配 AWS HealthLake 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，當您使用聯合或切換角色時，會自動建立。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](#)。

AWS HealthLake 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，並結合 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](#)。

AWS HealthLake 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](#)。

如需完整存取 AWS HealthLake 所需的服務角色和內嵌政策的相關資訊，請參閱 [設定 AWS HealthLake](#)。

Warning

變更服務角色的許可可能會中斷 HealthLake 功能。只有在 HealthLake 提供指引時，才能編輯服務角色。

AWS HealthLake 的服務連結角色

支援服務連結角色：否

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 AWS 帳戶中，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 HealthLake 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 HealthLake 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 HealthLake 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 HealthLake 的動作、資源和條件索引鍵](#)。

主題

- [政策最佳實務](#)
- [使用 AWS HealthLake 主控台](#)
- [在 中存取 an AWS HealthLake 資料存放區 Amazon Athena](#)
- [允許使用者檢視自己的許可](#)

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 HealthLake 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 AWS HealthLake 主控台

若要存取 AWS HealthLake 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 HealthLake 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。

若要完整存取 HealthLake，請將下列政策連接至 IAM 使用者或角色：

AmazonHealthLakeFullAccess 和 AWSLakeFormationDataAdmin。您也需要連接 HealthLake 內嵌政策，即服務角色。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](#)。如需建立所需服務角色的內嵌政策的相關資訊，請參閱 [設定 AWS HealthLake](#)。您還必須使用 AWS Lake Formation 主控台或 CLI 將 HealthLake 管理員指派為 AWS Lake Formation Data Lake 管理員。如需詳細資訊，請參閱 [設定 AWS HealthLake](#)。

在中存取 an AWS HealthLake 資料存放區 Amazon Athena

如果您想要為使用者和角色提供中 HealthLake 資料存放區的存取權 Amazon Athena，請將下列 IAM 政策連接至角色或使用者：AmazonAthenaFullAccess 和 AmazonS3FullAccess。Select 和 Describe 許可也需要由 AWS Lake Formation 主控台或透過 CLI 授予 AWS Lake Formation 管理員。如需詳細資訊，請參閱 [設定 AWS HealthLake](#)

允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視連接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": "*"
  }
}
```

AWS AWS HealthLake 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住，AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義特定於使用案例的[客戶管理政策](#)，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新 AWS 受管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。AWS 服務 當新的 啟動或新的 API 操作可供現有服務使用時，AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#)。

AWS 受管政策：AmazonHealthLakeFullAccess

此AmazonHealthLakeFullAccess政策提供 HealthLake 的完整存取權。將此政策連接到其使用者或角色後，使用者可以使用 HealthLake 來存取、查詢、匯入和匯出 HealthLake 中的資料。若要在 HealthLake 中執行許多常見動作，您必須將其他政策新增至使用者或角色。如需詳細資訊，請參閱 [設定 AWS HealthLake](#) 和 [HealthLake 操作和許可](#)。

您可將 AmazonHealthLakeFullAccess 政策連接到 IAM 身分。

此政策會授予管理和參與者許可，允許使用者和角色使用 HealthLake 查詢、搜尋、匯入和匯出，也可讓 HealthLake 代表具有這些許可的使用者和角色執行動作。

許可詳細資訊

此政策包含下列陳述式。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "healthlake:*",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "iam:ListRoles"
      ],
      "Resource": "*",
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "healthlake.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS 受管政策：AmazonHealthLakeReadOnlyAccess

AmazonHealthLakeReadOnlyAccess 政策會將唯讀存取和許可授予 HealthLake 和其他 AWS 服務中的相關資源。將此政策套用到您想要授予查詢和檢視 HealthLake 資料存放區功能的使用者，但不能建立或變更使用者。

您可將 AmazonHealthLakeReadOnlyAccess 政策連接到 IAM 身分。

此政策授予##許可，允許使用者和角色查詢 HealthLake。

許可詳細資訊

此政策包含下列陳述式。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "healthlake:ListFHIRDatastores",
        "healthlake:DescribeFHIRDatastore",
        "healthlake:DescribeFHIRImportJob",
        "healthlake:DescribeFHIRExportJob",
        "healthlake:GetCapabilities",
        "healthlake:ReadResource",
        "healthlake:SearchWithGet",
        "healthlake:SearchWithPost",
        "healthlake:SearchEverything"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

HealthLake 操作和許可

下表列出 HealthLake 中的典型操作，以及執行這些操作所需的許可。

HealthLake 操作	所需的許可
在 HealthLake 中建立資料存放區	AmazonHealthLakeFullAccess、AmazonLakeFormationDataAdmin、 內嵌政策和 由管理的 AWS Lake Formation 管理員許可 AWS Lake Formation
在 HealthLake 中刪除資料存放區	AmazonHealthLakeFullAccess、AmazonLakeFormationDataAdmin 內嵌政策和 由管理的 AWS Lake Formation 管理員許可 AWS Lake Formation
在 HealthLake 中列出、搜尋或查詢資料存放區	AmazonHealthLakeReadOnlyAccess
使用 查詢資料存放區 Amazon Athena	AmazonAthenaFullAccess 受管理之資料表的 AmazonS3FullAccess、AWS Lake Formation Select 和 Describe 許可 AWS Lake Formation
從 HealthLake 匯入資料	請參閱 設定匯入任務的許可 。
從 HealthLake 匯出資料	請參閱 設定匯出任務的許可 。

HealthLake 受 AWS 管政策的更新

檢視自此服務開始追蹤這些變更以來 HealthLake AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 HealthLake 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	Date
AmazonHealthLakeFullAccess	AmazonHealthLakeFullAccess 允許完整存取 HealthLake 所需的政策。	2022 年 11 月 14 日
AmazonHealthLakeReadOnlyAccess	AmazonHealthLakeReadOnlyAccess HealthLake 唯讀存取所需的政策。	2022 年 11 月 14 日
HealthLake 開始追蹤變更	HealthLake 開始追蹤其 AWS 受管政策的變更。	2022 年 11 月 14 日

疑難排解 AWS HealthLake 身分和存取

使用以下資訊來協助您診斷和修正使用 HealthLake 和 IAM 時可能遇到的常見問題。

主題

- [我無權在 AWS HealthLake 中執行動作](#)
- [我未獲得執行 iam:PassRole 的授權](#)
- [我想要允許 AWS 帳戶外的人員存取 my AWS HealthLake 資源](#)

我無權在 AWS HealthLake 中執行動作

如果 AWS 管理主控台告訴您無權執行動作，則必須聯絡您的管理員尋求協助。您的管理員是為您提供使用者名稱和密碼的人員。

以下範例錯誤會在 mateojackson IAM 使用者嘗試使用主控台檢視虛構 *my-example-widget* 資源的詳細資訊，但卻沒有虛構 healthlake:*GetWidget* 許可時發生。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
healthlake:GetWidget on resource: my-example-widget
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *my-example-widget* 動作存取 healthlake:*GetWidget* 資源。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行 iam:PassRole 動作，您的政策必須更新，以允許您將角色傳遞給 HealthLake。

有些 AWS 服務可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM marymajor 使用者嘗試使用主控台在 HealthLake 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞給服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 AWS 帳戶外的人員存取 my AWS HealthLake 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 HealthLake 是否支援這些功能，請參閱 [如何使用 IAM 搭配 AWS HealthLake](#)。
- 若要了解如何 AWS 帳戶 在您擁有的 資源之間提供存取權，請參閱 [《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 IAM 使用者中提供存取權](#)。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱 [《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有](#)。
- 如需了解如何透過聯合身分提供存取權，請參閱 [《IAM 使用者指南》中的將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 [《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取](#)。

AWS HealthLake 的合規驗證

在多個合規計畫中，第三方稽核人員會評估 AWS HealthLake 的安全性和 AWS 合規性。對於 HealthLake，這包括 HIPAA。

若要了解 是否 AWS 服務 在特定合規計畫範圍內，請參閱[AWS 服務 合規計畫範圍內](#)然後選擇您感興趣的合規計畫。如需一般資訊，請參閱[AWS 合規計畫](#)。

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 HealthLake 中的基礎設施安全

作為受管服務，AWS HealthLake 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 已發佈的 API 呼叫，透過網路存取 HealthLake。用戶端必須支援 Transport Layer Security (TLS) 1.0 或更新版本。建議使用 TLS 1.2 或更新版本。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。

使用 建立 AWS HealthLake 資源 AWS CloudFormation

AWS HealthLake 已與 整合 AWS CloudFormation，這項服務可協助您建立和設定 AWS 資源的模型，以減少建立和管理資源和基礎設施的時間。您可以建立範本，描述您想要的所有 AWS 資源，並為您 CloudFormation 佈建和設定這些資源。

使用 時 CloudFormation，您可以重複使用範本來一致且重複地設定 HealthLake 資源。描述您的資源一次，然後在多個 AWS 帳戶 和 區域中逐一佈建相同的資源。

HealthLake 和 CloudFormation 範本

若要佈建和設定 HealthLake 和相關服務的資源，您必須了解 [CloudFormation 範本](#)。範本是以 JSON 或 YAML 格式化的文本檔案。這些範本說明您要在 CloudFormation 堆疊中佈建的資源。如果您不熟悉

JSON 或 YAML，您可以使用 CloudFormation 設計工具來協助您開始使用 CloudFormation 範本。如需詳細資訊，請參閱《AWS CloudFormation 使用者指南》中的[什麼是 CloudFormation 設計器？](#)。

Note

AWS HealthLake 支援使用 建立資料存放區 CloudFormation。如需詳細資訊，包括用於佈建 HealthLake 資料存放區的 JSON 和 YAML 範本範例，請參閱AWS CloudFormation 《使用者指南》中的 [AWS HealthLake 資源類型參考](#)。

進一步了解 CloudFormation

若要進一步了解 CloudFormation，請參閱下列資源：

- [AWS CloudFormation](#)
- [AWS CloudFormation 使用者指南](#)
- [CloudFormation API 參考](#)
- [AWS CloudFormation 命令列界面使用者指南](#)

AWS HealthLake 和介面 VPC 端點 (AWS PrivateLink)

您可以建立介面 VPC 端點，在 VPC 與 AWS HealthLake 之間建立私有連線。介面 VPC 端點採用 [AWS PrivateLink](#) 技術，您可以使用這項技術來私下存取 HealthLake；APIs 無需網際網路閘道、NAT 裝置、VPN 連接或 Direct Connect 連線。VPC 中的執行個體不需要公有 IP 地址，即可與 HealthLake；APIs 通訊。VPC 與 HealthLake 之間的流量；不會離開 Amazon 網路。

每個介面端點都是由您子網路中的一或多個[彈性網路介面](#)表示。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的介面 [VPC 端點 \(AWS PrivateLink\)](#)。

HealthLake VPC 端點的考量事項

在為 HealthLake 設定介面 VPC 端點之前，請務必檢閱《Amazon VPC 使用者指南》中的[介面端點屬性和限制](#)。

HealthLake 支援從您的 VPC 呼叫其所有 API 動作。

為 HealthLake 建立介面 VPC 端點；

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 為 HealthLake ； 服務建立 VPC 端點AWS CLI。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[建立介面端點](#)。

使用下列服務名稱為 HealthLake 建立 VPC 端點：

- `com.amazonaws.region.healthlake`

如果您開啟端點的私有 DNS，您可以使用區域的預設 DNS 名稱向 HealthLake 提出 API 請求。例如 `healthlake.us-east-1.amazonaws.com`。

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[透過介面端點存取服務](#)。

為 HealthLake 建立 VPC 端點政策

您可以將端點政策連接至控制 HealthLake 存取的 VPC 端點。此政策會指定下列資訊：

- 可執行動作的主體。
- 可執行的動作。
- 可供執行動作的資源。

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的[使用 VPC 端點控制對服務的存取](#)。

範例：HealthLake 動作的 VPC 端點政策

以下是 HealthLake 端點政策的範例。連接到端點時，此政策會授予所有資源上所有主體對 HealthLake `CreateFHIRDatastore`動作的存取權。

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "healthlake:create-fhir-datastore"
      ],
      "Resource": "*"
    }
  ]
}
```

}

AWS HealthLake 中的安全最佳實務

AWS HealthLake 提供許多安全功能，供您在開發和實作自己的安全政策時加以考量。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

- 實作最低權限存取。
- 盡可能使用 Customer-Managed-Keys (CMKs) 來加密您的資料。若要進一步了解 CMKs，請參閱 [Amazon Key Management Service](#)。
- 查詢資料存放區中的 PHI 或 PII 時，請使用搜尋搭配 POST，而非使用 GET 搜尋。
- 限制對敏感和重要稽核函數的存取。
- 透過更新或大量匯入 APIs 建立資源時，請勿在任何可見欄位或邏輯 FHIR ID (LID) 中使用 PHI 或 PII，包括資料存放區和任務的名稱。
- 傳送建立、讀取、更新、刪除或搜尋請求時，請勿使用 HTTP 標頭中的 PHI。
- 啟用 AWS CloudTrail 以使用 audit AWS HealthLake，並確保沒有非預期的活動。
- 檢閱安全使用 Amazon S3 儲存貯體的[最佳實務](#)。若要進一步了解，請參閱《Amazon S3 使用者指南》中的[安全最佳實務](#)。

彈性 in AWS HealthLake

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。

如果您需要在更大的地理距離內複製資料或應用程式，請使用 AWS 本地區域。AWS 本機區域是單一資料中心，旨在補充現有的 AWS 區域。如同所有 AWS 區域，AWS 本機區域與其他 AWS 區域完全隔離。

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](#)。

AWS HealthLake 參考

下列支援參考資料適用於 FHIR、FHIR 和 上的 SMART AWS HealthLake。

Note

所有原生 HealthLake 動作和資料類型會在單獨的參考中描述。如需詳細資訊，請參閱 [AWS HealthLake API 參考](#)。

主題

- [的 FHIR 上的 SMART 支援 AWS HealthLake](#)
- [的 FHIR R4 支援 AWS HealthLake](#)
- [的合規參考 AWS HealthLake](#)
- [的支援參考 AWS HealthLake](#)

的 FHIR 上的 SMART 支援 AWS HealthLake

啟用 FHIR 的 HealthLake 資料存放區上的可替代醫療應用程式和可重複使用技術 (SMART) 允許在符合 FHIR 的應用程式上存取 SMART。HealthLake 資料是透過使用第三方授權伺服器來驗證和授權請求來存取。因此，您不是透過 管理使用者憑證 AWS Identity and Access Management，而是使用符合 FHIR 規範的授權伺服器上的 SMART。

Note

HealthLake 在 FHIR 版本 1.0 和 2.0 上支援 SMART。若要進一步了解這些架構，請參閱 FHIR R4 文件中的 [SMART 應用程式啟動](#)。

HealthLake 資料存放區支援下列 FHIR 請求上的 SMART 身分驗證和授權架構：

- OpenID (AuthN)：驗證人員或用戶端應用程式是他們聲稱是誰（或什麼）。
- OAuth 2.0 (AuthZ)：用於授權 HealthLake 資料存放區中的哪些 FHIR 資源，已驗證的請求可以讀取或寫入。這是由授權伺服器中設定的範圍所定義。

您可以使用 AWS CLI AWS SDKs 在啟用 FHIR 的資料存放區上建立 SMART。如需詳細資訊，請參閱 [建立 HealthLake 資料存放區](#)。

在啟用 FHIR 的資料存放區上建立 SMART 之後，您可以使用更新其身分提供者組態，包括授權伺服器中繼資料、權杖驗證 Lambda 函數和授權策略 UpdateFHIRDatastore。如需詳細資訊，請參閱[更新 HealthLake 資料存放區](#)。

主題

- [FHIR 上的 SMART 入門](#)
- [FHIR 上 SMART 的 HealthLake 身分驗證要求](#)
- [HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART](#)
- [使用 驗證權杖 AWS Lambda](#)
- [在啟用 FHIR 的 HealthLake 資料存放區上搭配 SMART 使用精細授權](#)
- [在 FHIR 探索文件上擷取 SMART](#)
- [在已啟用 SMART 功能的 HealthLake 資料存放區上提出 FHIR REST API 請求](#)

FHIR 上的 SMART 入門

下列主題說明如何開始使用 AWS HealthLake 的 FHIR 授權上的 SMART。其中包括您必須在 AWS 帳戶中佈建的資源、在啟用 FHIR 的 HealthLake 資料存放區上建立 SMART，以及 FHIR 用戶端應用程式如何與授權伺服器和 HealthLake 資料存放區互動的範例。

主題

- [在 FHIR 上設定適用於 SMART 的資源](#)
- [FHIR 上適用於 SMART 的用戶端應用程式工作流程](#)

在 FHIR 上設定適用於 SMART 的資源

下列步驟定義 HealthLake 如何處理 FHIR 請求上的 SMART，以及成功所需的資源。下列元素在工作流程中共同運作，以在 FHIR 請求上建立 SMART：

- 最終使用者：通常，病患或臨床醫生在 FHIR 上使用第三方 SMART 來存取 HealthLake 資料存放區中的資料。
- FHIR 上的 SMART 應用程式（稱為用戶端應用程式）：想要存取 HealthLake 資料存放區中找到之資料的應用程式。
- 授權伺服器：符合 OpenID Connect 標準的伺服器，能夠驗證使用者並發出存取權杖。
- HealthLake 資料存放區：啟用 FHIR 的 SMART HealthLake 存放區，使用 Lambda 函數回應提供承載字符的 FHIR REST 請求。

若要讓這些元素一起運作，您必須建立下列資源。

Note

我們建議您在設定授權伺服器、定義其上的必要範圍，並建立 AWS Lambda 函數來處理權杖自我檢查之後，在啟用 FHIR 的 HealthLake 資料存放區上建立您的 SMART。

1. 設定授權伺服器端點

若要在 FHIR 架構上使用 SMART，您需要設定第三方授權伺服器，以驗證對資料存放區提出的 FHIR REST 請求。如需詳細資訊，請參閱[FHIR 上 SMART 的 HealthLake 身分驗證要求](#)。

2. 在授權伺服器上定義範圍，以控制 HealthLake 資料存放區存取層級

FHIR 上的 SMART 架構使用 OAuth 範圍來判斷已驗證請求可存取的 FHIR 資源，以及存取程度。定義範圍是為最低權限設計的一種方式。如需詳細資訊，請參閱[HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART](#)。

3. 設定能夠執行權杖自我檢查的 AWS Lambda 函數

用戶端應用程式在啟用 FHIR 之 SMART 的資料存放區上傳送的 FHIR REST 請求包含 JSON Web Token (JWT)。如需詳細資訊，請參閱[解碼 JWT](#)。

4. 在啟用 FHIR 的 HealthLake 資料存放區上建立 SMART

若要在 FHIR HealthLake 資料存放區上建立 SMART，您需要提供 IdentityProviderConfiguration。如需詳細資訊，請參閱[建立 HealthLake 資料存放區](#)。

Note

建立資料存放區之後，您可以使用來更新身分提供者組態，例如輪換權杖驗證 Lambda 函數、變更授權伺服器中繼資料，或變更授權策略 UpdateFHIRDatastore。如需詳細資訊，請參閱[更新 HealthLake 資料存放區](#)。

FHIR 上適用於 SMART 的用戶端應用程式工作流程

下一節說明如何在 FHIR 上的 SMART 內容中啟動用戶端應用程式，並在 HealthLake 資料存放區上提出成功的 FHIR REST 請求。

1. 使用用戶端應用程式向 Well-Known Uniform Resource Identifier 提出 GET 請求

已啟用 SMART 的用戶端應用程式必須提出 GET 請求，才能尋找 HealthLake 資料存放區的授權端點。這是透過 Well-Known Uniform Resource Identifier (URI) 請求來完成的。如需詳細資訊，請參閱 [在 FHIR 探索文件上擷取 SMART](#)。

2. 請求存取和範圍

用戶端應用程式使用授權伺服器的授權端點，讓使用者可以登入。此程序會驗證使用者。範圍用於定義用戶端應用程式可在 HealthLake 資料存放區中存取的 FHIR 資源。如需詳細資訊，請參閱 [HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART](#)。

3. 存取權杖

現在使用者已通過身分驗證，用戶端應用程式會從授權伺服器接收 JWT 存取字符。當用戶端應用程式傳送 FHIR REST 請求給 HealthLake 時，會提供此字符。如需詳細資訊，請參閱 [字符驗證](#)。

4. 在啟用 FHIR 的 HealthLake 資料存放區上對 SMART 提出 FHIR REST API 請求

用戶端應用程式現在可以使用授權伺服器提供的存取權杖，將 FHIR REST API 請求傳送至 HealthLake 資料存放區端點。如需詳細資訊，請參閱 [在已啟用 SMART 功能的 HealthLake 資料存放區上提出 FHIR REST API 請求](#)。

5. 驗證 JWT 存取權杖

若要驗證 FHIR REST 請求中傳送的存取權杖，請使用 Lambda 函數。如需詳細資訊，請參閱 [使用驗證權杖 AWS Lambda](#)。

FHIR 上 SMART 的 HealthLake 身分驗證要求

若要存取啟用 FHIR 之 HealthLake 資料存放區上的 SMART 中的 FHIR 資源，用戶端應用程式必須由 OAuth 2.0 相容授權伺服器授權，並在 FHIR REST API 請求中呈現 OAuth Bearer 權杖。若要尋找授權伺服器的端點，請透過 Well-Known 統一資源識別符在 FHIR 探索文件上使用 HealthLake SMART。若要進一步了解此程序，請參閱 [在 FHIR 探索文件上擷取 SMART](#)。

在 FHIR HealthLake 資料存放區上建立 SMART 時，您必須在 CreateFHIRDatastore 請求的 metadata 元素中定義授權伺服器的端點和字符端點。若要進一步了解如何定義 metadata 元素，請參閱 [建立 HealthLake 資料存放區](#)。

您也可以使用更新現有資料存放區上的 metadata 元素（包括授權和字符端點）UpdateFHIRDatastore。如需詳細資訊，請參閱 [更新 HealthLake 資料存放區](#)。

使用授權伺服器端點，用戶端應用程式將使用授權服務來驗證使用者。一旦授權和驗證，授權服務會產生 JSON Web Token (JWT) 並傳遞給用戶端應用程式。此字符包含允許用戶端應用程式使用的 FHIR

資源範圍，進而限制使用者可存取的資料。或者，如果提供了啟動範圍，則回應將包含這些詳細資訊。若要進一步了解 HealthLake 支援的 FHIR 範圍上的 SMART，請參閱 [HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART](#)。

使用授權伺服器授予的 JWT，用戶端應用程式會對啟用 FHIR 的 HealthLake 資料存放區上的 SMART 進行 FHIR REST API 呼叫。若要驗證和解碼 JWT，您需要建立 Lambda 函數。HealthLake 會在收到 FHIR REST API 請求時代表您叫用此 Lambda 函數。若要查看入門 Lambda 函數範例，請參閱 [使用驗證權杖 AWS Lambda](#)。

在啟用 FHIR 的 HealthLake 資料存放區上建立 SMART 所需的授權伺服器元素

在 CreateFHIRDatastore 請求中，您需要提供授權端點和字符端點，做為 IdentityProviderConfiguration 物件中 metadata 元素的一部分。授權端點和字符端點都是必要的。若要查看在 CreateFHIRDatastore 請求中如何指定範例，請參閱 [建立 HealthLake 資料存放區](#)。

在啟用 FHIR 的 HealthLake 資料存放區上完成 SMART 上的 FHIR REST API 請求所需的宣告

您的 AWS Lambda 函數必須包含下列宣告，才能在啟用 FHIR 的 HealthLake 資料存放區上成為 SMART 上的有效 FHIR REST API 請求。

- **nbf**：[\(非之前\) 宣告](#) — "nbf" (非之前) 宣告會識別不得接受 JWT 處理的時間。處理 "nbf" 宣告時，目前日期/時間必須晚於或等於 "nbf" 宣告中列出的不早於日期/時間。我們提供的 Lambda 函數範例 `iat` 會從伺服器回應轉換為 `nbf`。
- **exp**：[\(過期時間\) 宣告](#) — 「過期時間」(過期時間) 宣告會識別 JWT 當天或之後不得接受處理的過期時間。
- **isAuthorized**：布林值設定為 `True`。指出請求已在授權伺服器上獲得授權。
- **aud**：[\(對象\) 宣告](#) — "aud" (對象) 宣告可識別 JWT 的收件人。這必須是啟用 FHIR 的 HealthLake 資料存放區端點上的 SMART。
- **scope**：這必須至少是一個 FHIR 資源相關範圍。此範圍在您的授權伺服器上定義。若要進一步了解 HealthLake 接受的 FHIR 資源相關範圍，請參閱 [HealthLake 的 FHIR 資源範圍上的 SMART](#)。

HealthLake 支援的 FHIR OAuth 2.0 範圍上的 SMART

HealthLake 使用 OAuth 2.0 做為授權通訊協定。在授權伺服器上使用此通訊協定可讓您為用戶端應用程式可存取的 FHIR 資源定義 HealthLake 資料存放區許可 (建立、讀取、更新、刪除和搜尋)。

FHIR 上的 SMART 架構定義一組可以從授權伺服器請求的範圍。例如，僅設計為允許患者檢視其實驗室結果或檢視其聯絡詳細資訊的用戶端應用程式，應僅獲授權請求read範圍。

Note

HealthLake 同時支援 FHIR V1 和 V2 上的 SMART，如下所述。建立資料存放區時，SMART on FHIR [AuthorizationStrategy](#)會設定為下列三個值之一：

- SMART_ON_FHIR_V1 – 僅支援 FHIR V1 上的 SMART，其中包含 read (讀取/搜尋) 和 write(create/update/delete)許可。
- SMART_ON_FHIR – 支援 FHIR V1 和 V2 上的 SMART，其中包括 create、read、update、delete和 search許可。
- AWS_AUTH – 預設的 AWS HealthLake 授權策略；與 FHIR 上的 SMART 無關。

獨立啟動範圍

HealthLake 支援獨立啟動模式範圍 launch/patient。

在獨立啟動模式中，用戶端應用程式會請求存取病患的臨床資料，因為用戶端應用程式不知道使用者和病患。因此，用戶端應用程式的授權請求會明確請求傳回病患範圍。身分驗證成功後，授權伺服器會發出存取字符，其中包含請求的啟動病患範圍。所需的患者內容會在授權伺服器的回應中與存取字符一起提供。

支援的啟動模式範圍

Scope (範圍)	說明
launch/patient	OAuth 2.0 授權請求中的參數，請求在授權回應中傳回該病患資料。

HealthLake 的 FHIR 資源範圍上的 SMART

HealthLake 在 FHIR 資源範圍上定義三個層級的 SMART。

- patient 範圍會授予單一病患特定資料的存取權。
- user 範圍會授予使用者可存取的特定資料的存取權。
- system 範圍會授予 HealthLake 資料存放區中所有 FHIR 資源的存取權。

下列各節列出使用 FHIR V1 上的 SMART 或 FHIR V2 上的 SMART 建構 FHIR 資源範圍的語法。

Note

建立資料存放區時，會設定 FHIR 上的 SMART 授權策略。如需詳細資訊，請參閱 AWS HealthLake API 參考 [AuthorizationStrategy](#) 中的。

HealthLake 支援的 FHIR V1 範圍上的 SMART

在 FHIR V1 上使用 SMART 時，用於建構 HealthLake 的 FHIR 資源範圍的一般語法如下。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
('patient' | 'user' | 'system') '/' (fhir-resource | '*') '.' ('read' | 'write' | '*')
```

FHIR v1 支援的授權範圍上的 SMART

範圍語法	範例範圍	結果
patient/(fhir-resource '*'). ('read' 'write' '*')	patient/AllergyIntolerance.*	病患用戶端應用程式具有所有記錄的區分的執行個體層級讀取/寫入存取權。
user/(fhir-resource '*'). ('read' 'write' '*')	user/Observation.read	使用者用戶端應用程式具有所有記錄觀察的執行個體層級讀取/寫入存取權。
system/('read' 'write' '*')	system/*.*	系統用戶端應用程式具有所有 FHIR 資源資料的讀取/寫入存取權。

HealthLake 支援的 FHIR V2 範圍上的 SMART

在 FHIR V2 上使用 SMART 時，用於建構 HealthLake 的 FHIR 資源範圍的一般語法如下。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
('patient' | 'user' | 'system') '/' (fhir-resource | '*') '.' ('c' | 'r' | 'u' | 'd' | 's')
```

Note

若要在 FHIR V2 上使用 SMART，您必須將值傳入 [permission-v2](#) 中繼資料 capabilities 字串，這是 [IdentityProviderConfiguration](#) 資料類型的成員。HealthLake 支援精細範圍。如需詳細資訊，請參閱 FHIR 美國核心實作指南中 [支援的精細範圍](#)。

FHIR V2 支援的授權範圍上的 SMART

範圍語法	範例 V1 範圍	結果
patient/Observation.rs	user/Obse rvation.read	讀取和搜尋目前病患 Observation 資源的許可。
system/*.cruds	system/*.*	系統用戶端應用程式具有所有 FHIR 資源資料的完整 create/read/update/刪除/搜尋存取權。

HealthLake 支援的 FHIR V2.2 範圍上的 SMART

V2.2 使用以搜尋參數為基礎的篩選來擴展 V2 範圍。授權伺服器現在可以發出限制特定資料特性存取的範圍，而不只是根據資源類型和 CRUDS 操作。

V2 的所有內容保持不變。V2.2 純粹是附加的：

- 現有的 V2 範圍（不含篩選條件）會繼續像以前一樣運作。
- V2 文法是以選用的 ?param=value 查詢字串延伸。
- 範圍層級 (patient/user/system)、資源類型或 CRUDS 字母沒有變更。

先決條件

在 FHIR V2.2 上啟用 SMART 之前，請確定下列事項：

- 您的資料存放區已建立，並將 `AuthorizationStrategy` 設為 `SMART_ON_FHIR` (同時支援 V1 和 V2)。使用 `SMART_ON_FHIR_V1` 或 `AWS_AUTH` 的資料存放區不符合資格。
- 您的資料存放區已包含在 `capabilities` 陣列 `permission-v2` 中。V2.2 是擴充 V2 時的附加項目，無法獨立使用。
- 您的 IDP Lambda 已設定為驗證並傳遞 V2.2 範圍格式 (包含查詢語法的範圍)。

啟用 V2.2

啟用路徑取決於您要建立新的資料存放區或更新現有的資料存放區。

新的資料存放區

建立新的資料存放區時，請將 `permission-v2.2` 新增至 `Metadata` 欄位中的 `capabilities` 陣列 [IdentityProviderConfiguration](#)：

```
"capabilities": [  
  "launch-ehr",  
  "sso-openid-connect",  
  "client-public",  
  "permission-v2",  
  "permission-v2.2"  
]
```

現有資料存放區

若要在現有資料存放區上啟用 FHIR V2.2 上的 SMART，請將 `permission-v2.2` 新增至 `Metadata` 欄位中的 `capabilities` 陣列，[IdentityProviderConfiguration](#) 並使用 `提交變更 UpdateFHIRDatastore`。如需詳細資訊，請參閱 [更新 HealthLake 資料存放區](#)。

使用要求：

- `permission-v2` 必須保留在陣列中。V2.2 擴展 V2，無法單獨使用。
- `AuthorizationStrategy` 必須是 `SMART_ON_FHIR` (不是 `SMART_ON_FHIR_V1` 或 `AWS_AUTH`)。
- 更新身分提供者組態會將其完全取代，因此請包含所有現有的欄位和功能以及 `permission-v2.2`。

變更會立即生效，沒有停機時間。若要驗證，請擷取 [探索文件](#) (需要 SigV4)：

```
GET {healthlake-endpoint}/r4/.well-known/smart-configuration
```

如果回應中的capabilities陣列包含 permission-v2.2，則 FHIR V2.2 上的 SMART 會處於作用中狀態。

延伸範圍語法

V2 文法是以選用的查詢字串延伸：

```
V2: (patient|user|system) / resource . cruds
V2.2: (patient|user|system) / resource . cruds [? param=value [& param=value ...]]
```

V2.2 範圍查詢字串元件

元件	說明
?param=value	Search-parameter 篩選條件。只有符合此條件的資源才能存取。
¶m=value	其他篩選條件。多個篩選條件是 ANDed – 所有篩選條件都必須相符。

規則：

- 篩選條件僅適用於範圍中指定的資源類型。不支援萬用字元 (*) 搭配篩選條件。
- 參數必須根據您資料存放區的搜尋組態對資源類型有效 (透過 檢查GET /r4/metadata)。
- 完整範圍字串 (例如 patient/Observation.rs?category=laboratory) 是出現在 OAuth 2.0 範圍參數和存取字符scp宣告中的常值。
- 授權請求中每個 [RFC 6749](#) 的 URL | 編碼特殊字元 (例如 → %7C)。
- 對於日期、數字和數量參數，V2.2 支援字首 [比較器](#) (例如 ?date=eq2023-01-01)。V2.2 [也支援搜尋參數修改程式](#)。

範圍範例

V2.2 範圍範例

Scope (範圍)	授予的存取
patient/DiagnosticReport.rs?category=LAB	只有 category 為 DiagnosticReport 的資源 LAB。
patient/Observation.rs?_security=http://terminology.hl7.org/CodeSystem/v3-Confidentiality R	只有具有安全標籤 Observation 的資源 Restricted。
patient/Observation.rs?date=ge2023-01-01	只有日期為 2023 年 1 月 1 日或之後 Observation 的資源。
patient/Observation.rs?category=laboratory&status=final	只有實驗室 AND 最終版 Observation 的資源。
user/Condition.rs?clinical-status=active	只有作用中 Condition 的資源。

強制執行行為

當字符包含 V2.2 範圍時，HealthLake 會套用每個操作的篩選條件：

每個操作的 V2.2 強制執行

作業	Behavior (行為)
讀取 (r)	只有在資源符合所有範圍篩選條件時，才會成功。否則會傳回 403。
搜尋 (s)	範圍篩選條件與查詢相交。只會傳回相符的資源。
建立/更新 (c/u)	資源必須符合範圍篩選條件才能寫入。否則會傳回 403。
刪除 (d)	目標資源必須符合範圍篩選條件。否則會傳回 403。

範圍優先順序

- 相同資源類型的多個 V2.2 範圍會聯集 (跨範圍的 OR)。
- 沒有篩選條件的更廣泛 V2 範圍 (例如 `patient/Observation.rs`) 會授予完整存取權，無論相同權杖中任何更窄的 V2.2 範圍為何。
- 未 `permission-v2.2` 啟用的資料存放區上的 V2.2 範圍會被無提示地忽略。

限制

V2.2 範圍篩選條件不支援下列項目：

- 複合搜尋參數 (例如 `code-value-quantity`)。
- 鏈結搜尋參數 (例如 `subject:Patient.name=Smith`)。
- `_include / _reinclude` 搜尋參數。
- `$export / $davinci-data-export` (大量資料) – 不適用 V2.2 篩選條件；大量匯出使用 V2 範圍。
- 萬用字元資源類型與篩選條件結合 (例如，`patient/*.rs?category=LAB` 無效)。使用搜尋參數篩選條件時，您必須指定明確的資源類型 (例如 `patient/Observation.rs?category=LAB`)。

疑難排解

徵狀	原因	修正
無法辨識字符範圍	資料存放區上未啟用 V2.2	透過支援票證檢查 <code>/.well-known/smart-configuration</code> 並請求啟用。
資源上存在的 403	資源不符合範圍篩選條件	根據範圍參數驗證資源值。
空的搜尋結果	範圍篩選條件比查詢更窄	結果是查詢和範圍篩選條件的交集。
InvalidScope 錯誤	範圍內的無效搜尋參數	透過 <code>/metadata CapabilityStatement</code> 確認參數。

端對端範例

案例：患者應用程式應僅顯示 2023 年以後的最終實驗室結果。

1. 具有範圍的授權伺服器問題字符：

```
patient/Observation.rs?category=laboratory&status=final&date=ge2023-01-01
```

2. 用戶端呼叫 HealthLake：

```
GET {endpoint}/r4/Observation?patient=Patient/123
```

3. HealthLake 會強制執行範圍篩選條件。回應僅包含 `category=laboratory AND status=final` 和 `date ≥ 2023-01-01` – 即使用戶端請求所有觀察 `Observation`。

使用 驗證權杖 AWS Lambda

當您在啟用 FHIR 的資料存放區上建立 HealthLake SMART 時，您必須在 `CreateFHIRDatastore` 請求中提供 AWS Lambda 函數的 ARN。Lambda 函數的 ARN 是使用 `IdpLambdaArn` 參數在 `IdentityProviderConfiguration` 物件中指定。

您必須先建立 Lambda 函數，才能建立啟用 FHIR 的 SMART 資料存放區。建立資料存放區之後，您可以使用更新以指向不同的 Lambda `IdpLambdaArn` 函數 `UpdateFHIRDatastore`。若要查看目前為資料存放區設定的 Lambda ARN，請使用 `DescribeFHIRDatastore` API 動作。如需詳細資訊，請參閱 [更新 HealthLake 資料存放區](#)。

若要讓 FHIR REST 請求在啟用 FHIR 的資料存放區上在 SMART 上成功，您的 Lambda 函數必須執行下列動作：

- 在 1 秒內將回應傳回 HealthLake 資料存放區端點。
- 解碼用戶端應用程式所傳送 REST API 請求的授權標頭中提供的存取字符。
- 指派具有足夠許可的 IAM 服務角色來執行 FHIR REST API 請求。
- 需要下列宣告才能完成 FHIR REST API 請求。如需詳細資訊，請參閱 [必要宣告](#)。
 - `nbf`
 - `exp`
 - `isAuthorized`
 - `aud`
 - `scope`

使用 Lambda 時，除了 Lambda 函數之外，還需要建立執行角色和資源型政策。Lambda 函數的執行角色是 IAM 角色，授予函數存取執行時間所需 AWS 服務和資源的許可。您提供的資源型政策必須允許 HealthLake 代表您叫用您的函數。

本主題中的各節說明來自用戶端應用程式和解碼回應的範例請求、建立 AWS Lambda 函數所需的步驟，以及如何建立 HealthLake 可採用的資源型政策。

- [第 1 部分：建立 Lambda 函數](#)
- [第 2 部分：建立 AWS Lambda 函數使用的 HealthLake 服務角色](#)
- [第 3 部分：更新 Lambda 函數的執行角色](#)
- [第 4 部分：將資源政策新增至 Lambda 函數](#)
- [第 5 部分：為您的 Lambda 函數佈建並行](#)

建立 AWS Lambda 函數

此主題中建立的 Lambda 函數會在 HealthLake 收到對啟用 FHIR 的 SMART 資料存放區的請求時觸發。來自用戶端應用程式的請求包含 REST API 呼叫，以及包含存取字符的授權標頭。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/  
Authorization: Bearer i8hweunweunweofiwweoijewiwe
```

本主題中的範例 Lambda 函數會使用 AWS Secrets Manager 來隱藏與授權伺服器相關的登入資料。我們強烈建議不要在 Lambda 函數中提供授權伺服器登入詳細資訊。

Example 驗證包含授權承載字符的 FHIR REST 請求

Lambda 函數範例示範如何驗證傳送到啟用 FHIR 之 SMART 資料存放區的 FHIR REST 請求。若要查看如何實作此 Lambda step-by-steps 說明，請參閱 [使用 建立 Lambda 函數 AWS 管理主控台](#)。

如果 FHIR REST API 請求不包含有效的資料存放區端點、存取字符和 REST 操作，Lambda 函數將會失敗。若要進一步了解必要的授權伺服器元素，請參閱 [必要宣告](#)。

```
import base64  
import boto3  
import logging  
import json  
import os  
from urllib import request, parse  
  
logger = logging.getLogger()
```

```
logger.setLevel(logging.INFO)

## Uses Secrets manager to gain access to the access key ID and secret access key for
the authorization server
client = boto3.client('secretsmanager', region_name="region-of-datastore")
response = client.get_secret_value(SecretId='name-specified-by-customer-in-
secretsmanager')
secret = json.loads(response['SecretString'])
client_id = secret['client_id']
client_secret = secret['client_secret']

unencoded_auth = f'{client_id}:{client_secret}'
headers = {
    'Authorization': f'Basic {base64.b64encode(unencoded_auth.encode()).decode()}',
    'Content-Type': 'application/x-www-form-urlencoded'
}

auth_endpoint = os.environ['auth-server-base-url'] # Base URL of the Authorization
server
user_role_arn = os.environ['iam-role-arn'] # The IAM role client application will use
to complete the HTTP request on the datastore

def lambda_handler(event, context):
    if 'datastoreEndpoint' not in event or 'operationName' not in event or
'bearerToken' not in event:
        return {}

    datastore_endpoint = event['datastoreEndpoint']
    operation_name = event['operationName']
    bearer_token = event['bearerToken']
    logger.info('Datastore Endpoint [{}], Operation Name:
[{}]' .format(datastore_endpoint, operation_name))

    ## To validate the token
    auth_response = auth_with_provider(bearer_token)
    logger.info('Auth response: [{}]' .format(auth_response))
    auth_payload = json.loads(auth_response)
    ## Required parameters needed to be sent to the datastore endpoint for the HTTP
request to go through
    auth_payload["isAuthorized"] = bool(auth_payload["active"])
    auth_payload["nbf"] = auth_payload["iat"]
    return {"authPayload": auth_payload, "iamRoleARN": user_role_arn}
```

```
## access the server
def auth_with_provider(token):
    data = {'token': token, 'token_type_hint': 'access_token'}
    req = request.Request(url=auth_endpoint + '/v1/introspect',
    data=parse.urlencode(data).encode(), headers=headers)
    with request.urlopen(req) as resp:
        return resp.read().decode()
```

使用 建立 Lambda 函數 AWS 管理主控台

下列程序假設您已在啟用 FHIR 的 SMART 資料存放區上處理 FHIR REST API 請求時，建立希望 HealthLake 擔任的服務角色。如果您尚未建立服務角色，您仍然可以建立 Lambda 函數。您必須先新增服務角色的 ARN，Lambda 函數才能運作。若要進一步了解如何建立服務角色並在 Lambda 函數中指定該角色，請參閱 [建立 HealthLake 服務角色以用於用來解碼 JWT 的 AWS Lambda 函數](#)

建立 Lambda 函數 (AWS 管理主控台)

1. 開啟 Lambda 主控台中的 [函數頁面](#)。
2. 選擇建立函數。
3. 選取從頭開始撰寫。
4. 在基本資訊下，輸入函數名稱。在執行時間下，選擇以 Python 為基礎的執行時間。
5. 執行角色 請選擇 建立具備基本 Lambda 許可的新角色。

Lambda 會建立函數和[執行角色](#)，此角色會授予函數將日誌上傳至 Amazon CloudWatch 的許可。Lambda 函數會在您叫用函數時擔任執行角色，並使用執行角色來建立 AWS SDK 的登入資料。

6. 選擇程式碼索引標籤，然後新增範例 Lambda 函數。

如果您尚未為 Lambda 函數建立要使用的服務角色，則必須先建立該角色，範例 Lambda 函數才能運作。若要進一步了解如何為 Lambda 函數建立服務角色，請參閱 [建立 HealthLake 服務角色以用於用來解碼 JWT 的 AWS Lambda 函數](#)。

```
import base64
import boto3
import logging
import json
import os
from urllib import request, parse

logger = logging.getLogger()
```

```
logger.setLevel(logging.INFO)

## Uses Secrets manager to gain access to the access key ID and secret access key
for the authorization server
client = boto3.client('secretsmanager', region_name="region-of-datastore")
response = client.get_secret_value(SecretId='name-specified-by-customer-in-
secretsmanager')
secret = json.loads(response['SecretString'])
client_id = secret['client_id']
client_secret = secret['client_secret']

unencoded_auth = f'{client_id}:{client_secret}'
headers = {
    'Authorization': f'Basic {base64.b64encode(unencoded_auth.encode()).decode()}',
    'Content-Type': 'application/x-www-form-urlencoded'
}

auth_endpoint = os.environ['auth-server-base-url'] # Base URL of the Authorization
server
user_role_arn = os.environ['iam-role-arn'] # The IAM role client application will
use to complete the HTTP request on the datastore

def lambda_handler(event, context):
    if 'datastoreEndpoint' not in event or 'operationName' not in event or
    'bearerToken' not in event:
        return {}

    datastore_endpoint = event['datastoreEndpoint']
    operation_name = event['operationName']
    bearer_token = event['bearerToken']
    logger.info('Datastore Endpoint [{}], Operation Name:
    [{}]'.format(datastore_endpoint, operation_name))

    ## To validate the token
    auth_response = auth_with_provider(bearer_token)
    logger.info('Auth response: [{}]'.format(auth_response))
    auth_payload = json.loads(auth_response)
    ## Required parameters needed to be sent to the datastore endpoint for the HTTP
request to go through
    auth_payload["isAuthorized"] = bool(auth_payload["active"])
    auth_payload["nbf"] = auth_payload["iat"]
    return {"authPayload": auth_payload, "iamRoleARN": user_role_arn}
```

```
## Access the server
def auth_with_provider(token):
    data = {'token': token, 'token_type_hint': 'access_token'}
    req = request.Request(url=auth_endpoint + '/v1/introspect',
    data=parse.urlencode(data).encode(), headers=headers)
    with request.urlopen(req) as resp:
        return resp.read().decode()
```

修改 Lambda 函數的執行角色

建立 Lambda 函數之後，您需要更新執行角色，以包含呼叫 Secrets Manager 的必要許可。在 Secrets Manager 中，您建立的每個秘密都有 ARN。若要套用最低權限，執行角色應只能存取執行 Lambda 函數所需的資源。

您可以在 IAM 主控台中搜尋 Lambda 函數，或在 Lambda 主控台中選擇組態，以修改 Lambda 函數的執行角色。若要進一步了解如何管理 Lambda 函數執行角色，請參閱 [Lambda 執行角色](#)。

Example 授予 存取權的 Lambda 函數執行角色 GetSecretValue

將 IAM 動作新增至 GetSecretValue 執行角色會授予範例 Lambda 函數運作所需的許可。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret-name-DKodTA"
    }
  ]
}
```

此時，您已建立 Lambda 函數，可用於驗證作為 FHIR REST 請求的一部分提供的存取字符，該請求會傳送到啟用 FHIR 的 SMART 資料存放區。

建立 HealthLake 服務角色以用於用來解碼 JWT 的 AWS Lambda 函數

人物：IAM 管理員

可新增或移除 IAM 政策，並建立新的 IAM 身分的使用者。

服務 角色

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](#)。

JSON Web Token (JWT) 解碼後，Lambda 需要授權，才能傳回 IAM 角色 ARN。此角色必須具有執行 REST API 請求的必要許可，否則會因為許可不足而失敗。

使用 IAM 設定自訂政策時，最好授予所需的最低許可。若要進一步了解，請參閱《IAM 使用者指南》中的[套用最低權限許可](#)。

建立 HealthLake 服務角色以在授權 Lambda 函數中指定 需要兩個步驟。

- 首先，您需要建立 IAM 政策。此政策必須指定您在授權伺服器中為 提供範圍的 FHIR 資源存取權。
- 其次，您需要建立 服務角色。當您建立角色時，您可以指定信任關係，並連接您在步驟 1 中建立的政策。信任關係會將 HealthLake 指定為服務委託人。在此步驟中，您需要指定 HealthLake 資料存放區 ARN 和 AWS 帳戶 ID。

建立新的 IAM 政策

您在授權伺服器中定義的範圍會決定已驗證使用者在 HealthLake 資料存放區中可存取的 FHIR 資源。

您可以量身打造您建立的 IAM 政策，以符合您定義的範圍。

您可以在 IAM 政策陳述式的 Action 元素中定義下列動作。對於資料表 Action 中的每個，您可以定義 Resource types。在 HealthLake 中，資料存放區是唯一支援的資源類型，可在 IAM 許可政策陳述式的 Resource 元素中定義。

個別 FHIR 資源不是您可以定義為 IAM 許可政策中 元素的資源。

HealthLake 定義的動作

動作	描述	存取層級	資源類型 (必要)
CreateResource	准許建立資源	寫入	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
DeleteResource	准許刪除資源	寫入	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
ReadResource	准許讀取資源	讀取	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
SearchWithGet	准許使用 GET 方法搜尋資源	讀取	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
SearchWithPost	准許使用 POST 方法搜尋資源	讀取	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
StartFHIRExportJobWithPost	准許使用 GET 開始 FHIR 匯出任務	寫入	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>
UpdateResource	准許更新資源	寫入	資料存放區 ARN : <code>arn : aws : healthlakeyour-region : 111122223333 : datastore/fhir/your-datastore-id</code>

若要開始使用，您可以使用 `AmazonHealthLakeFullAccess`。此政策會授予資料存放區中所有 FHIR 資源的讀取、寫入、搜尋和匯出。若要授予資料存放區的唯一讀許可，請使用 `AmazonHealthLakeReadOnlyAccess`。

若要進一步了解如何使用 AWS 管理主控台、AWS CLI 或 IAM SDKs 建立自訂政策，請參閱《[IAM 使用者指南](#)》中的 [建立 IAM 政策](#)。

建立 HealthLake 的服務角色 (IAM 主控台)

使用此程序來建立服務角色。建立服務時，您也需要指定 IAM 政策。

建立 HealthLake 的服務角色 (IAM 主控台)

1. 登入 AWS 管理主控台 並開啟位於 <https://console.aws.amazon.com/iam/> 的 IAM 主控台。
2. 在 IAM 主控台的導覽窗格中，選擇角色。
3. 然後，選擇 Create role (建立角色)。
4. 在選取信任實體頁面上，選擇自訂信任政策。
5. 接著，在自訂信任政策下更新範例政策，如下所示。**your-account-id** 以您的帳號取代，並新增您要在匯入或匯出任務中使用的資料存放區的 ARN。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "healthlake.amazonaws.com"
      },
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:healthlake:us-east-1:123456789012:datastore/fhir/your-datastore-id"
        }
      }
    }
  ]
}
```

6. 然後選擇下一步。
7. 在新增許可頁面上，選擇您希望 HealthLake 服務擔任的政策。若要尋找您的政策，請在許可政策下進行搜尋。
8. 然後，選擇連接政策。
9. 然後在角色名稱下的名稱、檢閱和建立頁面上輸入名稱。
10. (選用) 接著在描述下，為您的角色新增簡短描述。
11. 如果可能，請輸入角色名稱或角色名稱後綴，以協助您識別此角色的用途。角色名稱在您的 AWS 帳戶內必須是獨一無二的。它們無法透過大小寫進行區分。例如，您無法建立名為 **PRODRole** 和 **prodrole** 的角色。因為有各種實體可能會參照角色，所以您無法在建立角色之後編輯角色名稱。
12. 檢閱角色詳細資訊，然後選擇建立角色。

若要了解如何在範例 Lambda 函數中指定角色 ARN，請參閱 [建立 AWS Lambda 函數](#)。

Lambda 執行角色

Lambda 函數的執行角色是授予函數存取 AWS 服務和資源的許可的 IAM 角色。本頁提供有關如何建立、檢視和管理 Lambda 函數執行角色的資訊。

根據預設，當您使用 建立新的 Lambda 函數時，Lambda 會建立具有最少許可的執行角色 AWS 管理主控台。若要管理執行角色中授予的許可，請參閱 Lambda 開發人員指南中的 [在 IAM 主控台中建立執行角色](#)。

本主題中提供的 Lambda 函數範例使用 Secrets Manager 來隱藏授權伺服器的憑證。

如同您建立的任何 IAM 角色，請務必遵循最低權限最佳實務。在開發片語期間，有時您可能會授予超出必要範圍的許可。在生產環境中發佈您的函數之前，最佳實務是調整政策以僅包含必要的許可。如需詳細資訊，請參閱《IAM 使用者指南》中的 [套用最低權限](#)。

允許 HealthLake 觸發您的 Lambda 函數

因此 HealthLake 可以代表您叫用 Lambda 函數，您必須執行下列動作：

- 您需要將 設定為 `IdpLambdaArn` 等於您希望 HealthLake 在 `CreateFHIRDatastore` 請求中調用之 Lambda 函數的 ARN。
- 您需要以資源為基礎的政策，允許 HealthLake 代表您叫用 Lambda 函數。

當 HealthLake 在啟用 FHIR 的 SMART 資料存放區上收到 FHIR REST API 請求時，它需要許可來代表您調用資料存放區建立時指定的 Lambda 函數。若要授予 HealthLake 存取權，您將使用以資源為基礎的政策。若要進一步了解如何為 Lambda 函數建立資源型政策，請參閱《AWS Lambda 開發人員指南》中的[允許 AWS 服務呼叫 Lambda 函數](#)。

為您的 Lambda 函數佈建並行

Important

HealthLake 需要 Lambda 函數的最大執行時間少於一秒 (1000 毫秒)。
如果您 Lambda 函數超過執行時間限制，您會收到 TimeOut 例外狀況。

為了避免發生此例外狀況，建議您設定佈建並行。您可以在增加呼叫前配置佈建並行，來確保所有請求均由具有低延遲的初始化執行個體所提供。若要進一步了解如何設定佈建並行，請參閱 Lambda 開發人員指南中的[設定佈建並行](#)

若要查看 Lambda 函數的平均執行時間，目前請使用 Lambda 主控台上 Lambda 函數的監控頁面。根據預設，Lambda 主控台會提供持續時間圖表，顯示函數程式碼處理事件的平均、最短和最長時間。若要進一步了解監控 Lambda 函數，請參閱 [Lambda 開發人員指南中的 Lambda 主控台](#)中的[監控函數](#)。

如果您已為 Lambda 函數佈建並行，並想要監控它，請參閱《Lambda 開發人員指南》中的[監控並行](#)。

在啟用 FHIR 的 HealthLake 資料存放區上搭配 SMART 使用精細授權

僅[範圍](#)不會為您提供請求者有權在資料存放區中存取哪些資料的必要具體性。在啟用 FHIR 的 HealthLake 資料存放區上授予 SMART 存取權時，使用精細授權可實現更高層級的特異性。若要使用精細授權，請在 CreateFHIRDatastore 請求的 IdentityProviderConfiguration 參數 True 中將設定為 FineGrainedAuthorizationEnabled 等於。

您也可以使用 [更新](#)，FineGrainedAuthorizationEnabled 啟用或停用現有資料存放區的精細授權 UpdateFHIRDatastore。如需詳細資訊，請參閱[更新 HealthLake 資料存放區](#)。

如果您啟用精細授權，您的授權伺服器會傳回 `fhirUser` 範圍 `id_token` 以及存取權杖。這允許用戶端應用程式擷取有關使用者的資訊。用戶端應用程式應將 `fhirUser` 宣告視為代表目前使用者的 FHIR 資源 URI。此值可以為 Patient、Practitioner 或 RelatedPerson。授權伺服器的回應也包含定義使用者可存取哪些資料 `user/` 的範圍。這會使用針對 FHIR 資源特定範圍相關範圍定義的語法：

```
user/(fhir-resource | '*').('read' | 'write' | '*')
```

以下是如何使用精細授權進一步指定資料存取相關 FHIR 資源類型的範例。

- 當 fhirUser 為 Practitioner，精細授權會決定使用者可存取的患者集合。fhirUser 僅允許病患參考 fhirUser 做為一般從業人員的那些病患存取。

```
Patient.generalPractitioner : [{Reference(Practitioner)}]
```

- 當 fhirUser 是 Patient 或 RelatedPerson 且請求中參考的患者與不同 fhirUser，精細授權會決定所請求患者對 fhirUser 的存取權。在請求 Patient 的資源中指定關係時，允許存取。

```
Patient.link.other : {Reference(Patient|RelatedPerson)}
```

在 FHIR 探索文件上擷取 SMART

SMART 定義探索文件，可讓用戶端了解 HealthLake 資料存放區支援的授權端點 URLs 和功能。此資訊可協助用戶端將授權請求導向正確的端點，並建構 HealthLake 資料存放區支援的授權請求。

若要讓用戶端應用程式成功向 HealthLake 提出 FHIR REST 請求，必須收集 HealthLake 資料存放區定義的授權要求。不需要承載字符（授權），此請求才能成功。

請求 HealthLake 資料存放區的探索文件

- 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱 [取得資料存放區屬性](#)。
- 使用 HealthLake region 和 datastoreId 的收集值來建構請求的 URL。附加 /.well-known/smart-configuration 至 URL 的端點。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/.well-known/smart-configuration
```

- 使用 GET 搭配 [AWS Signature 第 4 版](#) 簽署通訊協定來傳送請求。若要檢視整個範例，請捲動至複製按鈕。

curl

```
curl --request GET \
```

```
'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/.well-known/
smart-configuration \
--aws-sigv4 'aws:amz:region:healthlake' \
--user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
--header "x-amz-security-token:$AWS_SESSION_TOKEN" \
--header 'Accept: application/json'
```

HealthLake 資料存放區的探索文件會傳回為 JSON blob，您可以在其中找到 `authorization_endpoint` 和 `token_endpoint`，以及資料存放區的規格和定義功能。

```
{
  "authorization_endpoint": "https://oidc.example.com/authorize",
  "token_endpoint": "https://oidc.example.com/oauth/token",
  "capabilities": [
    "launch-ehr",
    "client-public"
  ]
}
```

`token_endpoint` 需要 `authorization_endpoint` 和 才能啟動用戶端應用程式。

- 授權端點 — 授權用戶端應用程式或使用者的 URL。
- 權杖端點 — 用戶端應用程式用來與之通訊的授權伺服器端點。

在已啟用 SMART 功能的 HealthLake 資料存放區上提出 FHIR REST API 請求

您可以在啟用 FHIR 的 HealthLake 資料存放區上的 SMART 上提出 FHIR REST API 請求。下列範例顯示來自用戶端應用程式的請求，其中包含授權標頭中的 JWT，以及 Lambda 應如何解碼回應。用戶端應用程式請求經過授權和驗證後，必須從授權伺服器收到承載字符。在啟用 FHIR 的 HealthLake 資料存放區上傳送 SMART 上的 FHIR REST API 請求時，請使用授權標頭中的承載字符。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/[ID]
Authorization: Bearer auth-server-provided-bearer-token
```

因為在授權標頭中找到承載字符，且未偵測到 IAM AWS 身分，所以 HealthLake 會叫用在建立啟用 SMART on FHIR 的 HealthLake 資料存放區時指定的 Lambda 函數。當您的 Lambda 函數成功解碼字符時，以下範例回應會傳送至 HealthLake。

```
{
  "authPayload": {
    "iss": "https://authorization-server-endpoint/oauth2/token", # The issuer
    identifier of the authorization server
    "aud": "https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/", #
    Required, data store endpoint
    "iat": 1677115637, # Identifies the time at which the token was issued
    "nbf": 1677115637, # Required, the earliest time the JWT would be valid
    "exp": 1997877061, # Required, the time at which the JWT is no longer valid
    "isAuthorized": "true", # Required, boolean indicating the request has been
    authorized
    "uid": "100101", # Unique identifier returned by the auth server
    "scope": "system/*.*" # Required, the scope of the request
  },
  "iamRoleARN": "iam-role-arn" #Required, IAM role to complete the request
}
```

的 FHIR R4 支援 AWS HealthLake

AWS HealthLake 支援 FHIR R4 規格進行運作狀態資料交換。下列各節提供 HealthLake 如何使用 FHIR R4 規格來協助您使用 FHIR R4 RESTful APIs [管理和搜尋](#) HealthLake 資料存放區中的 FHIR 資源的支援資訊。

主題

- [的 FHIR R4 功能陳述式 AWS HealthLake](#)
- [HealthLake 的 FHIR 設定檔驗證](#)
- [HealthLake 的 FHIR R4 支援的資源類型](#)
- [HealthLake 的 FHIR R4 搜尋參數](#)
- [\\$operations 適用於 HealthLake 的 FHIR R4](#)

的 FHIR R4 功能陳述式 AWS HealthLake

若要尋找作用中 HealthLake 資料存放區的 FHIR 相關功能（行為），您必須擷取其功能陳述式。功能陳述式用作實際伺服器功能的陳述式，或所需或所需伺服器實作的陳述式。FHIR [capabilities](#) 互動會擷取 HealthLake 資料存放區功能及其支援的 FHIR 規格部分的相關資訊。HealthLake 會根據 FHIR R4 資源驗證 FHIR [StructureDefinition](#) 資源類型。

取得 HealthLake 資料存放區的功能陳述式

1. 收集 HealthLake region 和 datastoreId 值。如需詳細資訊，請參閱[取得資料存放區屬性](#)。
2. 使用 HealthLake region 和 的收集值來建構請求的 URLdatastoreId。同時在 URL 中包含 FHIR metadata 元素。若要在下列範例中檢視整個 URL 路徑，請捲動至複製按鈕。

```
https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/metadata
```

3. 傳送 請求。FHIR capabilities 互動使用具有 [AWS Signature 第 4 版](#) 簽署通訊協定的 GET 請求。下列 curl 範例取得 所指定 HealthLake 資料存放區的功能陳述式 datastoreId。若要檢視整個範例，請捲動至複製按鈕。

curl

```
curl --request GET \  
  'https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/metadata \  
  --aws-sigv4 'aws:amz:region:healthlake' \  
  --user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \  
  --header "x-amz-security-token:$AWS_SESSION_TOKEN" \  
  --header 'Accept: application/json'
```

您將會收到 HealthLake 資料存放區的 200 HTTP 回應代碼和功能陳述式。如需詳細資訊，請參閱 FHIR R4 文件 [CapabilityStatement](#) 中的。

HealthLake 的 FHIR 設定檔驗證

AWS HealthLake 支援基本 [FHIR R4 規格](#)。基本 FHIR R4 規格中包含 FHIR 設定檔。設定檔用於 FHIR 資源類型，以使用基本資源類型的限制條件和/或延伸來定義更具體的資源類型定義。例如，FHIR 設定檔可以識別必要欄位，例如延伸項目和值集。資源可以支援多個設定檔。所有 HealthLake 資料存放區都支援使用 FHIR Profiles。

Note

將資料新增至 HealthLake 資料存放區時，不需要新增 FHIR 設定檔。如果在新增或更新資源時未指定 FHIR 描述檔，則資源只會根據基本 FHIR R4 結構描述進行驗證。

FHIR 資源符合的 FHIR 設定檔會在匯入 HealthLake 之前包含在資源中。因此，FHIR 設定檔會在匯入期間由 HealthLake 驗證。

FHIR 設定檔在實作指南中指定。FHIR 實作指南 (IG) 是一組說明，說明如何將 FHIR 標準用於特定用途。HealthLake 會驗證下列實作指南中定義的 FHIR 設定檔。

支援的 FHIR 設定檔 AWS HealthLake

名稱	版本	實作指南	功能	美國東部 (俄亥俄)	美國東部 (維吉尼亞北部)	美國西部 (奧勒岡)	亞太地區 (孟買)	亞太地區 (悉尼)	加拿大 (中部)	歐洲 (愛爾蘭)	歐洲 (倫敦)
美國核心	3.1	http://hl7.org/fhir/us/core/STU3.1.1/	預設	X	X	X	X	X	X	X	X
美國核心	4.0	https://hl7.org/fhir/us/core/STU4/index.html	支援	X	X	X	X	X	X	X	X
美國核心	5.0	https://hl7.org/fhir/us/core/STU5.0.1/index.html	支援	X	X	X	X	X	X	X	X
美國核心	6.1	https://hl7.org/fhir/us/core/STU6.1/index.html	支援	X	X	X	X	X	X	X	X
美國核心	7.0	https://hl7.org/fhir/us/core/STU7/	支援	X	X	X	X	X	X	X	X
英國核心	2.0	https://simplifier.net/guide/uk-core-implementation-guide-stu2/Home/ProfilesandExtensions/Pro	支援	X	X	X	X			X	X

名稱	版本	實作指南	功能	美國東部 (俄亥俄)	美國東部 (維吉尼亞北部)	美國西部 (奧勒岡)	亞太地區 (孟買)	亞太地區 (悉尼)	加拿大 (中部)	歐洲 (愛爾蘭)	歐洲 (倫敦)
		filesIndex?version=2.0.1									
CARIN 藍色按鈕	1.1	http://hl7.org/fhir/us/carin-bb/STU1.1/	預設	X	X	X	X	X	X	X	X
CARIN 藍色按鈕	1.0.0	https://hl7.org/fhir/us/carin-bb/history.html	支援	X	X	X	X	X	X	X	X
Da Vinci 付款人資料交換	1.0	https://hl7.org/fhir/us/davinci-pdex/	預設	X	X	X	X	X	X	X	X
Da Vinci 付款人資料交換	2.0.0	https://hl7.org/fhir/us/davinci-pdex/history.html	支援	X	X	X	X	X	X	X	X
Da Vinci 運作狀態記錄交換 (HREx)	0.2	https://hl7.org/fhir/us/davinci-hrex/2020Sep/	預設	X	X	X	X	X	X	X	X
DaVinci PDEX 計畫淨額	1.1	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1.1/	預設	X	X	X	X	X	X	X	X
DaVinci PDEX 計畫淨額	1.0	https://hl7.org/fhir/us/davinci-pdex-plan-net/STU1/	支援	X	X	X	X	X	X	X	X

名稱	版本	實作指南	功能	美國東部 (俄亥俄)	美國東部 (維吉尼亞北部)	美國西部 (奧勒岡)	亞太地區 (孟買)	亞太地區 (悉尼)	加拿大 (中部)	歐洲 (愛爾蘭)	歐洲 (倫敦)
DaVinci Payer Data Exchange (PDex) 美國藥物配方	1.1	https://hl7.org/fhir/us/davinci-drug-formulary/STU1.1/	預設	X	X	X	X	X	X	X	X
DaVinci Payer Data Exchange (PDex) 美國藥物配方	1.0.1	https://hl7.org/fhir/us/davinci-drug-formulary/history.html	支援	X	X	X	X	X	X	X	X
Da Vinci 臨床資料交換 (CDex)	2.1	https://build.fhir.org/ig/HL7/davinci-ecd/index.html	預設	X	X	X	X	X	X	X	X
Da Vinci 預先授權支援 (PAS) FHIR IG	2.1	https://hl7.org/fhir/us/davinci-pas/	預設	X	X	X	X	X	X	X	X
NCQA HEDIS® 實作指南	0.3	https://www.ncqa.org/resources/hedis-ig-re-source-page/	預設	X	X	X	X			X	X

名稱	版本	實作指南	功能	美國東部 (俄亥俄)	美國東部 (維吉尼亞北部)	美國西部 (奧勒岡)	亞太地區 (孟買)	亞太地區 (悉尼)	加拿大 (中部)	歐洲 (愛爾蘭)	歐洲 (倫敦)
國際患者摘要 (IPS)	2.0 選票	https://hl7.org/fhir/uv/ips/2024Sep/	預設	X	X	X	X	X	X	X	X
品質指標	5.0	https://registry.fhir.org/package/hl7.fhir.us.cqfmeasures%7C5.0.0	預設	X	X	X	X			X	X
基因體報告	3.0	https://build.fhir.org/ig/HL7/genomics-reporting/index.html	預設	X	X	X	X			X	X
國家衛生局的 Ayushman Bharat Digital Mission (ABDM)	2.0	https://www.nrces.in/ndhm/fhir/r4/index.html	預設	X	X	X	X			X	X
CA Core+	1.1	https://simplifier.net/ca-core	支援						X		

名稱	版本	實作指南	功能	美國東部 (俄亥俄)	美國東部 (維吉尼亞北部)	美國西部 (奧勒岡)	亞太地區 (孟買)	亞太地區 (悉尼)	加拿大 (中部)	歐洲 (愛爾蘭)	歐洲 (倫敦)
CA : eReC Pan-Canadian eReferral-eConsult	1.1	https://simplifier.net/CA-eReC/~introduction	支援						X		
病患摘要 加拿大版本 - (PS-CA)	2.1	https://simplifier.net/PS-CA-R1/~introduction	支援						X		
Ontario Digital Health 藥物儲存庫	4.0	https://simplifier.net/ca-on-dhdr-r4/~introduction	支援						X		
AU 核心	1.0	https://hl7.org.au/fhir/core/	支援					X			
Magentus Practice Management	1.2	https://fhir-versions.dev.geniesolutions.io/1.2.16/downloads.html	支援					X			

驗證資源中指定的 FHIR 設定檔

若要驗證 FHIR 設定檔，請使用實作指南中指定的設定檔 URL，將其新增至個別資源的 profile 元素。

當您將新資源新增至資料存放區時，會驗證 FHIR 設定檔。若要新增資源，您可以使用 StartFHIRImportJob API 操作、提出新增資源的 POST 請求，或提出 PUT 以更新現有資源。

Example– 查看在資源中參考哪些 FHIR 設定檔

設定檔 URL 會新增至 "meta" : "profile" 鍵/值對中的 profile 元素。為了清楚起見，已截斷此資源。

```
{
  "resourceType": "Patient",
  "id": "abcd1234efgh5678hijk9012",
  "meta": {
    "lastUpdated": "2023-05-30T00:48:07.8443764-07:00",
    "profile": [
      "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
    ]
  }
}
```

Example– 如何參考非預設支援的 FHIR 設定檔

若要驗證受支援的非預設設定檔，請將版本控制的設定檔 URL 新增至 meta.profile 元素。版本控制的 URL 包含基本設定檔 URL，後面接著管道字元 (|) 和版本號碼。為了清楚起見，已截斷此範例資源。

```
{
  "resourceType": "ExplanationOfBenefit",
  "id": "sample-EOB",
  "meta": {
    "lastUpdated": "2024-02-02T05:56:09.4+00:00",
    "profile": [
      "http://hl7.org/fhir/us/carina-bb/StructureDefinition/C4BB-ExplanationOfBenefit-Pharmacy|1.0.0"
    ]
  }
}
```

您也可以同時包含版本控制的 URL 和基本設定檔 URL。這兩種格式都是有效的。

```
{
  "resourceType": "ExplanationOfBenefit",
  "id": "sample-EOB",
  "meta": {
    "lastUpdated": "2024-02-02T05:56:09.4+00:00",
    "profile": [
      "http://hl7.org/fhir/us/caribb/StructureDefinition/C4BB-ExplanationOfBenefit-Pharmacy|1.0.0",
      "http://hl7.org/fhir/us/caribb/StructureDefinition/C4BB-ExplanationOfBenefit-Pharmacy"
    ]
  }
}
```

HealthLake 的 FHIR R4 支援的資源類型

下表列出 支援的 FHIR R4 資源類型 AWS HealthLake。如需詳細資訊，請參閱 FHIR R4 文件中的[資源索引](#)。

HealthLake 支援的 FHIR R4 資源類型

帳戶	DetectedIssue	發票	從業人員
ActivityDefinition	裝置	程式庫	PractitionerRole
AdverseEvent	DeviceDefinition	連結	程序
AllergyIntolerance	DeviceMetric	清單	證明
約定	DeviceUseStatement	Location	問卷
AppointmentResponse	DeviceRequest	量值	QuestionnaireResponse
AuditEvent - 請參閱備註	DiagnosticReport	MeasureReport	RelatedPerson
二進位	DocumentManifest	媒體	RequestGroup
BodyStructure	DocumentReference	藥物	ResearchStudy

套件 - 請參閱備註	EffectEvidenceSynt hesis	MedicationAdminist ration	ResearchSubject
CapabilityStatement	遇到	MedicationDispense	RiskAssessment
CarePlan	Endpoint	MedicationKnowledge	RiskEvidenceSynthe sis
CareTeam	EpisodeOfCare	MedicationRequest	Schedule
ChargeItem	EnrollmentRequest	MedicationStatement	ServiceRequest
ChargeItemDefinition	EnrollmentResponse	MessageHeader	插槽
取得	ExplanationOfBenefit	MolecularSequence	樣本
ClaimResponse	FamilyMemberHistory	NutritionOrder	StructureDefinition
Communication	旗標	觀察	StructureMap
CommunicationReque st	目標	OperationOutcome - 請參閱備註	物質
合成	Group	組織	SupplyDelivery
ConceptMap	GuidanceResponse	OrganizationAffiliation	SupplyRequest
條件	HealthcareService	參數 - 請參閱備註	任務
同意	ImagingStudy	病患	ValueSet
合約	預防	PaymentNotice	VisionPrescription
涵蓋範圍	ImmunizationEvalua tion	PaymentReconciliat ion	VerificationResult - 請 參閱備註
CoverageEligibilit yRequest	ImmunizationRecomm endation	個人	
CoverageEligibilit yResponse	InsurancePlan	PlanDefinition	

⚠️ FHIR 規格和 HealthLake

- 您不能使用 FHIR `OperationOutcomeParameters` 和資源類型提出 GET 或 POST 請求。
- `AuditEvent` — 可以建立或讀取 `AuditEvent` 資源，但無法更新或刪除。
- 套件 — HealthLake 管理套件請求的方式有多種。如需詳細資訊，請參閱[綁定 FHIR 資源](#)。
- `VerificationResult` — 只有 2023 年 12 月 9 日之後建立的資料存放區才支援此資源類型。

HealthLake 的 FHIR R4 搜尋參數

使用 FHIR [search](#) 互動，根據一些篩選條件在 HealthLake 資料存放區中搜尋一組 FHIR 資源。您可以使用 GET 或 POST 請求來執行 `search` 互動。對於涉及個人身分識別資訊 (PII) 或受保護醫療資訊 (PHI) 的搜尋，建議使用 POST 請求，因為 PII 和 PHI 會新增為請求內文的一部分，並在傳輸中加密。

📌 Note

本章所述的 FHIR `search` 互動是根據 HL7 FHIR R4 標準的醫療保健資料交換而建置。由於它是 HL7 FHIR 服務的表示，因此不會透過 AWS CLI 和 AWS SDKs 提供。如需詳細資訊，請參閱 FHIR R4 RESTful API 文件 [search](#) 中的。

您也可以使用 Amazon Athena 查詢具有 SQL 的 HealthLake 資料存放區。如需詳細資訊，請參閱[整合](#)。

HealthLake 支援下列 FHIR R4 搜尋參數子集。如需詳細資訊，請參閱[HealthLake 的 FHIR R4 搜尋參數](#)。

支援的搜尋參數類型

下表顯示 HealthLake 中支援的搜尋參數類型。

支援的搜尋參數類型

搜尋參數	Description
<code>_id</code>	資源 ID (非完整 URL)
<code>_lastUpdated</code>	上次更新日期。伺服器可自行決定邊界精確度。

搜尋參數	Description
_tag	依資源標籤搜尋。
_profile	搜尋標記設定檔的所有資源。
_安全性	搜尋套用至此資源的安全標籤。
_source	搜尋資源的來源。
_text	搜尋資源的敘述。
createdAt	搜尋自訂擴充功能 createdAt。

Note

下列搜尋參數僅支援 2023 年 12 月 9 日之後建立的資料存放區：_security , _source , _text , createdAt。

下表顯示如何根據指定資源類型的指定資料類型修改查詢字串的範例。為了清楚起見，範例欄中的特殊字元尚未編碼。若要成功進行查詢，請確定查詢字串已正確編碼。

搜尋參數範例

搜尋參數類型	詳細資訊	範例
Number	搜尋指定資源中的數值。觀察到重要的數字。有效數字的數量依搜尋參數值在中是特定的，前導零除外。允許比較字首。	[parameter]=100 [parameter]=1e2 [parameter]=lt100
日期/DateTime	搜尋特定日期或時間。預期的格式是 yyyy-mm-ddThh:mm:ss[Z (+ -)hh:mm]，但可能會有所不同。	[parameter]=eq2013-01-14 [parameter]=gt2013-01-14T10:00

搜尋參數類型	詳細資訊	範例
	<p>接受下列資料類型： date、dateTime、Period、instant和Timing。如需在搜尋中使用這些資料類型的詳細資訊，請參閱 FHIR R4 RESTful API 文件中的日期。</p> <p>允許比較字首。</p>	<pre>[parameter]=ne2013-01-14</pre>
String	<p>以區分大小寫的方式搜尋一系列字元。</p> <p>同時支援 HumanName 和 Address 類型。如需詳細資訊，請參閱 FHIR R4 文件中的HumanName 資料類型項目和Address 資料類型項目。</p> <p>使用:text 修飾詞支援進階搜尋。</p>	<pre>[base]/Patient?given=eve</pre> <pre>[base]/Patient?given:contains=eve</pre>
權杖	<p>搜尋與字元字串close-to-exact 相符項目，通常與一對醫療代碼值相比。</p> <p>區分大小寫會連結到建立查詢時使用的程式碼系統。以訂閱為基礎的查詢有助於減少與區分大小寫相關的問題。為了清楚起見， 尚未編碼。</p>	<pre>[parameter]=[system] [code] : 這裡[system]參考編碼系統，並[code]參考該特定系統中找到的程式碼值。</pre> <pre>[parameter]=[code] : 在這裡，您的輸入將符合程式碼或系統。</pre> <pre>[parameter]= [code] : 在此，您的輸入將符合程式碼，且系統屬性沒有識別符。</pre>

搜尋參數類型	詳細資訊	範例
複合	<p>使用修飾詞\$和,操作,在單一資源類型中搜尋多個參數。</p> <p>允許比較字首。</p>	<pre>/Patient?language=FR,NL&language=EN</pre> <pre>Observation?component-code-value-quantity=http://loinc.org 8480-6\$lt60</pre> <pre>[base]/Group?characteristic-value=gender\$mixed</pre>
數量	<p>搜尋數字、系統和程式碼做為值。號碼是必要項目,但系統和程式碼是選用項目。根據數量資料類型。如需詳細資訊,請參閱 FHIR R4 文件中的數量。</p> <p>使用以下假設語法 [parameter]=[prefix][number][system][code]</p>	<pre>[base]/Observation?value-quantity=5.4 http://unitsofmeasure.org mg</pre> <pre>[base]/Observation?value-quantity=5.4 http://unitsofmeasure.org mg</pre> <pre>[base]/Observation?value-quantity=5.4 http://unitsofmeasure.org mg</pre> <pre>[base]/Observation?value-quantity=le5.4 http://unitsofmeasure.org mg</pre>
參考資料	搜尋其他資源的參考。	<pre>[base]/Observation?subject=Patient/23test</pre>

搜尋參數類型	詳細資訊	範例
URI	搜尋明確識別特定資源的字元字串。	[base]/ValueSet?url=http://acme.org/fhir/ValueSet/123
特別	根據整合式醫療 NLP 延伸模組進行搜尋。	

HealthLake 支援的進階搜尋參數

HealthLake 支援下列進階搜尋參數。

名稱	描述	範例	功能
<code>_include</code>	用來請求在搜尋請求中傳回其他資源。它會傳回由目標資源執行個體參考的資源。	Encounter? _include=Encounter:subject	
<code>_revinclude</code>	用來請求在搜尋請求中傳回其他資源。它會傳回參考主要資源執行個體的資源。	Patient? id=patient- identifier &_revinclude=Encounter:patient	
<code>_summary</code>	摘要可用來請求資源的子集。	Patient? summary=text	支援下列摘要參數： _summary=true、 _summary=false、 _summary=text、 _summary=data。
<code>_elements</code>	請求傳回一組特定的元素，做為搜尋結果中資源的一部分。	Patient? elements=identify	

名稱	描述	範例	功能
		<code>r,active,link</code>	
<code>_total</code>	傳回符合搜尋參數的資源數目。	<code>Patient?_total=accurate</code>	支援 <code>_total=accurate</code> 、 <code>_total=none</code> 。
<code>_sort</code>	使用逗號分隔清單，指出傳回搜尋結果的排序順序。- 字首可用於逗號分隔清單中的任何排序規則，以表示遞減順序。	<code>Observation?_sort=status,-date</code>	支援依類型為的欄位排序 <code>Number</code> 、 <code>String</code> 、 <code>Quantity</code> 、 <code>Token</code> 、 <code>URI</code> 、 <code>Reference</code> 。只有 2023 年 12 月 9 日之後建立的資料存放區 <code>Date</code> 才支援依 排序。最多支援 5 個排序規則。
<code>_count</code>	控制搜尋套件每頁傳回多少資源。	<code>Patient?_count=100</code>	頁面大小上限為 100。
<code>chainin</code>	搜尋參考資源的元素。會將鏈結搜尋，導向至參考資源內的元素。	<code>DiagnosticReport?subject:Patient.name=peter</code>	
<code>reverse chainin (_has)</code>	根據參考資源的資源元素來搜尋資源。	<code>Patient?_has:Observation:patient:code=1234-5</code>	

`_include`

在搜尋查詢 `_include` 中使用 也允許傳回其他指定的 FHIR 資源。使用 `_include` 來包含向前連結的資源。

Example– 使用 `_include` 來尋找被診斷出有口鼻的患者或患者群組

您會搜尋指定診斷碼用於止吐Condition的資源類型，然後使用 `_include` 指定也要傳回該診斷subject的。在 Condition 資源類型中，`subject`是指病患資源類型或群組資源類型。

為了清楚起見，範例中的特殊字元尚未編碼。若要讓查詢成功，請確定查詢字串已正確編碼。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Condition?code=49727002&_include=Condition:subject
```

`_include:iterate` 修改器

`_include:iterate` 修飾詞允許跨兩個層級遞迴地包含參考的資源。例如

```
GET /ServiceRequest?identifier=025C0931195&_include=ServiceRequest:requester&_include:iterate=PractitionerRole:practitioner
```

將傳回 ServiceRequest、其相關聯的 PractitionerRole（透過請求者參考），然後遞迴地包含該 PractitionerRole 參考的 Practitioner。此修飾詞適用於 HealthLake 中的所有資源類型。

`_include=*` 修改器

`_include=*` 修飾詞是一種萬用字元，會自動包含搜尋結果直接參考的所有資源。例如

```
GET /ServiceRequest?specimen.accession=12345&_include=*
```

將傳回相符的 ServiceRequest 及其參考的所有資源（例如病患、從業人員、樣本等），而不需要個別指定每個參考路徑。此修飾詞適用於 HealthLake 中的所有資源類型。

`_revinclude`

在搜尋查詢 `_revinclude` 中使用也允許傳回其他指定的 FHIR 資源。使用 `_revinclude` 來包含向後連結的資源。

Example– 使用 `_revinclude` 包含連結至特定病患的相關事件和觀察資源類型

若要進行此搜尋，您必須先在 `_id` 搜尋參數中指定其識別符 Patient 來定義個人。然後，您可以使用結構 `Encounter:patient` 和指定其他 FHIR 資源 `Observation:patient`。

為了清楚起見，範例中的特殊字元尚未編碼。若要讓查詢成功，請確定查詢字串已正確編碼。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/  
Patient?_id=patient-  
identifier&_revinclude=Encounter:patient&_revinclude=Observation:patient
```

_summary

在搜尋查詢 `_summary` 中使用 可讓使用者請求 FHIR 資源的子集。它可以包含下列其中一個值：`true`、`text`、`data`、`false`。任何其他值都會被視為無效。傳回的資源會在 `meta.tag 'SUBSETTED'` 中標示為 ，表示資源不完整。

- `true`：傳回資源（基本定義）中標示為「摘要」的所有支援元素。
- `text`：僅傳回 `'text'`、`'id'`、`'meta'` 元素，以及僅傳回頂層強制性元素。
- `data`：傳回除了「文字」元素以外的所有部分。
- `false`：傳回資源的所有部分（多個）

在單一搜尋請求中，`_summary=text` 無法與 `_include` 或 `_revinclude` 搜尋參數結合。

Example– 取得資料存放區中病患資源的「文字」元素。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?  
_summary=text
```

_elements

在搜尋查詢 `_elements` 中使用 允許請求特定的 FHIR 資源元素。傳回的資源會在 `meta.tag 'SUBSETTED'` 中標示為 ，表示資源不完整。

`_elements` 參數包含以逗號分隔的基本元素名稱清單，例如在資源的根層級定義的元素。只有列出的元素才會傳回。如果 `_elements` 參數值包含無效的元素，伺服器會忽略它們並傳回強制性元素和有效的元素。

`_elements` 不適用於包含的資源（搜尋模式為 的傳回資源 `include`）。

在單一搜尋請求中，`_elements` 無法與 `_summary` 搜尋參數結合。

Example– 取得 HealthLake 資料存放區中病患資源的「識別符」、「作用中」、「連結」元素。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?  
_elements=identifier,active,link
```

_total

在搜尋查詢 `_total` 中使用 `會傳回符合所請求搜尋參數的資源數量`。HealthLake 會在搜尋 `Bundle.total` 回應的 `中傳回相符資源的總數`（傳回的搜尋模式為 `的資源match`）。

`_total` 支援 `accurate`、`none` 參數值。`_total=estimate` 不支援。任何其他值都會被視為無效。`_total` 不適用於包含的資源（傳回的資源，其搜尋模式為 `include`）。

Example– 取得資料存放區中病患資源的總數：

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
_total=accurate
```

_sort

在搜尋查詢 `_sort` 中使用 `會依特定順序排列結果`。結果會根據排序規則的逗號分隔清單，依優先順序排序。排序規則應該是有效的搜尋參數。任何其他值都會被視為無效。

在單一搜尋請求中，您最多可以使用 5 個排序搜尋參數。您可以選擇使用 `-` 字首來表示遞減順序。伺服器預設會依遞增順序排序。

支援的排序搜尋參數類型為：Number, String, Date, Quantity, Token, URI, Reference。如果搜尋參數是指巢狀元素，則不支援此搜尋參數進行排序。例如，搜尋資源類型的 `'name'` 病患參考具有 HumanName 資料類型的 `Patient.name` 元素會被視為巢狀。因此，不支援依「名稱」排序病患資源。

Example– 在資料存放區中取得病患資源，並依出生日期遞增排序：

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
_sort=birthdate
```

_count

參數 `_count` 定義為伺服器關於在單一頁面中應傳回多少資源的指示。

頁面大小上限為 100。大於 100 的任何值都是無效的。`_count=0` 不支援。

Example– 搜尋病患資源，並將搜尋頁面大小設定為 25：

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?_count=25
```

Chaining and Reverse Chaining(_has)

在 FHIR 中鏈結和反向鏈結可提供更有效率且更精簡的方式來取得互連的資料，減少對多個個別查詢的需求，並讓開發人員和使用者更方便地擷取資料。

如果任何層級的遞迴傳回超過 100 個結果，HealthLake 將傳回 4xx，以保護資料存放區免於超載並導致多個分頁。

Example– 鏈結 - 取得所有 DiagnosticReport，其是指病患名稱為 Peter 的病患。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/DiagnosticReport?
subject:Patient.name=peter
```

Example– 反向鏈結 - 取得病患資源，其中病患資源由至少一個觀察參考，其中觀察的程式碼為 1234，而觀察是指病患搜尋參數中的病患資源。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient?
_has:Observation:patient:code=1234
```

支援的搜尋修飾詞

搜尋修飾詞會與字串型欄位搭配使用。HealthLake 中的所有搜尋修飾詞都使用布林式邏輯。例如，您可以:contains指定 指定較大的字串欄位應包含一個小字串，以便將其包含在搜尋結果中。

支援的搜尋修飾詞

搜尋修飾詞	Type
: missing	除了 之外的所有參數 Composite
: exact	String
: 包含	String
: not	權杖
: text	權杖
: 識別符	參考資料
: 以下	URI

支援的搜尋比較器

您可以使用搜尋比較器來控制搜尋中相符項目的性質。您可以在搜尋數字、日期和數量欄位時使用比較器。下表列出 HealthLake 支援的搜尋比較器及其定義。

支援的搜尋比較器

搜尋比較程式	Description
eq	資源中 參數的值等於提供的值。
ne	資源中 參數的值不等於提供的值。
gt	資源中 參數的值大於提供的值。
lt	資源中 參數的值小於提供的值。
ge	資源中 參數的值大於或等於提供的值。
le	資源中 參數的值小於或等於提供的值。
sa	資源中 參數的值會在提供的值之後啟動。
eb	資源中 參數的值會在提供的值之前結束。

HealthLake 不支援 FHIR 搜尋參數

HealthLake 支援所有 FHIR 搜尋參數，但下表所列的參數除外。如需 FHIR 搜尋參數的完整清單，請參閱 [FHIR 搜尋參數登錄檔](#)。

不支援的搜尋參數

Bundle-composition	Location-near
套件識別符	Consent-source-reference
Bundle-message	合約患者
Bundle-type	資源內容
Bundle-timestamp	資源查詢

\$operations 適用於 HealthLake 的 FHIR R4

FHIR \$ 操作（也稱為「美元操作」）是特殊的伺服器端函數，超出 FHIR Read 規格中的標準 CRUD (Create、Update、Delete) 操作。這些操作會以「\$」字首識別，並啟用複雜處理、資料轉換和大量操作，這些操作在使用標準 REST API 呼叫時難以或效率不佳。\$ 操作可以在系統層級、資源類型層級或特定資源執行個體上叫用，提供與 FHIR 資料互動的彈性方式。AWS HealthLake 支援多個 FHIR \$operations。如需其他詳細資訊，請參閱以下每個個別頁面。

主題

- [HealthLake 的 FHIR R4 \\$attribution-status 操作](#)
- [使用 刪除資源類型 \\$bulk-delete](#)
- [\\$bulk-member-match HealthLake 的 操作](#)
- [HealthLake 的 FHIR R4 \\$confirm-attribution-list 操作](#)
- [HealthLake 的 FHIR R4 \\$davinci-data-export 操作](#)
- [使用 產生臨床文件 \\$document](#)
- [使用 永久移除資源 \\$erase](#)
- [使用 取得患者資料 Patient/\\$everything](#)
- [使用 擷取 ValueSet 程式碼 \\$expand](#)
- [使用 FHIR 匯出 HealthLake 資料 \\$export](#)
- [HealthLake 的 FHIR \\$inquire 操作](#)
- [使用 擷取概念詳細資訊 \\$lookup](#)
- [\\$member-add HealthLake 的 操作](#)
- [\\$member-match HealthLake 的 操作](#)
- [\\$member-remove HealthLake 的 操作](#)
- [使用 移除病患室資源 \\$purge](#)
- [HealthLake 的 FHIR \\$questionnaire-package 操作](#)
- [HealthLake 的 FHIR \\$submit 操作](#)
- [使用 驗證 FHIR 資源 \\$validate](#)

HealthLake 的 FHIR R4 \$attribution-status 操作

擷取特定成員的屬性狀態，傳回包含與病患相關所有屬性資源的套件。此操作是 [FHIR 成員屬性 \(ATR\) 清單 IG 2.1.0 實作的一部分](#)。

Endpoint

```
POST [base]/Group/[id]/$attribution-status
```

請求參數

操作接受下列選用參數：

參數	類型	說明
memberId	識別符	請求其屬性狀態之成員的 MemberId
patientReference	參考資料	生產者系統中病患資源的參考

Note

patientReference 可以提供 memberId 或 ，或兩者都用於驗證目的。

請求範例

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "memberId",
      "valueIdentifier": {
        "system": "http://example.org",
        "value": "MBR123456789"
      }
    },
    {
      "name": "patientReference",
      "valueReference": {
        "reference": "Patient/patient-123",
        "display": "John Doe"
      }
    }
  ]
}
```

回應

傳回包含與病患相關屬性資源的套件：

資源	基數	Location
病患	1..1	Group.member.entity
涵蓋範圍	0..1	Group.member.extension : coverageReference
Organization/Practitioner/PractitionerRole	0..1	Group.member.extension : attributedProvider
任何資源	0..1	Group.member.extension : associatedData

回應範例

```
{
  "resourceType": "Bundle",
  "id": "bundle-response",
  "meta": {
    "lastUpdated": "2014-08-18T01:43:33Z"
  },
  "type": "collection",
  "entry": [
    {
      "fullUrl": "http://example.org/fhir/Patient/12423",
      "resource": {
        "resourceType": "Patient",
        "id": "12423",
        "meta": {
          "versionId": "1",
          "lastUpdated": "2014-08-18T01:43:31Z"
        },
        "active": true,
        "name": [
          {
            "use": "official",
            "family": "Chalmers",
            "given": ["Peter", "James"]
          }
        ]
      }
    }
  ]
}
```

```

    }
  ],
  "gender": "male",
  "birthDate": "1974-12-25"
}
},
{
  "fullUrl": "http://example.org/fhir/Coverage/123456",
  "resource": {
    "resourceType": "Coverage",
    "id": "1"
    // ... additional Coverage resource details
  }
},
{
  "fullUrl": "http://example.org/fhir/Organization/666666",
  "resource": {
    "resourceType": "Organization",
    "id": "2"
    // ... additional Organization resource details
  }
}
]
}

```

錯誤處理

操作會處理下列錯誤條件：

錯誤	HTTP 狀態	Description
無效的操作請求	400	不符合的請求參數或結構
找不到群組資源	404	指定的群組 ID 不存在
找不到病患資源	404	指定的病患參考不存在

授權和安全性

SMART 範圍需求

用戶端必須具有適當的權限，才能讀取群組資源和相關歸因資源

標準 FHIR 授權機制適用於所有操作

使用 刪除資源類型 `$bulk-delete`

AWS HealthLake 支援 `$bulk-delete` 操作，可刪除資料存放區中特定類型的所有資源。當您需要：

- 執行季節性稽核和清除
- 大規模管理資料生命週期
- 移除特定資源類型
- 遵循資料保留政策

Usage

您可以使用 POST 方法叫用 `$bulk-delete` 操作：

```
POST [base]/[ResourceType]/$bulk-delete?isHardDelete=false&deleteAuditEvent=true
```

Parameters

參數	Type	必要	預設	Description
<code>isHardDelete</code>	布林值	否	false	當為 true 時，會從儲存體永久移除資源
<code>deleteAuditEvent</code>	布林值	否	true	為 true 時，會刪除相關聯的稽核事件
<code>_since</code>	string	否	資料存放區 建立時間	輸入時，會根據資源的lastModified時間 選取開始截止時間來尋找資源。無法與 開始或結束搭配使用
<code>start</code>	string	否	資料存放區 建立時間	輸入時，會根據資源的lastModified時間 選取截止時間來尋找資源。可與 end 搭 配使用
<code>end</code>	string	否	任務提交時 間	輸入時，會根據資源的lastModified時間 選取結束截止時間以尋找資源

範例

範例請求

```
POST [base]/Observation/$bulk-delete?isHardDelete=false
```

回應範例

```
{
  "jobId": "jobId",
  "jobStatus": "SUBMITTED"
}
```

任務狀態

若要檢查大量刪除任務的狀態：

```
GET [base]/$bulk-delete/[jobId]
```

操作會傳回任務狀態資訊：

```
{
  "datastoreId": "datastoreId",
  "jobId": "jobId",
  "status": "COMPLETED",
  "submittedTime": "2025-10-09T15:09:51.336Z"
}
```

Behavior (行為)

\$bulk-delete 操作：

1. 以非同步方式處理大量資源
2. 維護 ACID 交易的資料完整性
3. 提供具有資源刪除計數的任務狀態追蹤
4. 支援軟式和硬式刪除模式
5. 包括刪除活動的完整稽核記錄
6. 允許選擇性刪除歷史版本和稽核事件

稽核記錄

`$bulk-delete` 操作會記錄為 `StartFHIRBulkDeleteJob` 和 `DescribeFHIRBulkDeleteJob`，其中包含詳細的操作資訊。

限制

- 當 `isHardDelete` 設為 `true` 時，硬刪除的資源不會出現在搜尋結果或 `_history` 查詢中。
- 透過此操作刪除的資源在處理期間可能暫時無法存取
- 僅根據歷史版本調整儲存計量 - `deleteVersionHistory=false` 不會調整資料存放區儲存

`$bulk-member-match` HealthLake 的操作

AWS HealthLake 支援非同步處理多個成員比對請求 `$bulk-member-match` 的操作。此操作可讓醫療保健組織在單一大量請求中使用人口統計和涵蓋範圍資訊，在不同的醫療保健系統中有效率地比對數百個成員的唯一識別符。此功能對於大規模 payer-to-payer 資料交換、成員轉換和 CMS 合規要求至關重要，並遵循 FHIR [規格](#)。

Note

`$bulk-member-match` 操作是以目前實驗性且可能會有所變更的基礎 FHIR 規格為基礎。隨著規格的演進，此 API 的行為和界面也會隨之更新。建議開發人員監控 AWS HealthLake 版本備註和相關的 FHIR 規格更新，以隨時掌握可能影響其整合的任何變更。

當您需要執行下列動作時，此操作特別有用：

- 在開放註冊期間大規模處理成員比對
- 促進付款人之間的大量成員轉換
- 支援大規模 CMS 合規資料交換需求
- 有效率地比對跨醫療保健網路的成員群組
- 將 API 呼叫降至最低，並提高大量比對案例的操作效率

Usage

`$bulk-member-match` 操作是使用 POST 方法在群組資源上調用的非同步操作：

```
POST [base]/Group/$bulk-member-match
```

提交大量比對請求後，您可以使用下列方式輪詢任務狀態：

```
GET [base]/$bulk-member-match-status/{jobId}
```

支援的參數

HealthLake 支援下列 FHIR \$bulk-member-match 參數：

參數	Type	必要	說明
MemberPatient	病患	是	包含要比對之成員人口統計資訊的患者資源。
CoverageToMatch	涵蓋範圍	是	用於比對現有記錄的涵蓋資源。
CoverageToLink	涵蓋範圍	否	比對程序期間要連結的覆蓋資源。
Consent	同意	是	授權用途的同意資源也會儲存。這與不需要同意的個別\$member-match 操作不同。

提交大量成員比對任務的 POST 請求

下列範例顯示提交大量成員比對任務的 POST 請求。每個成員都會包裝在MemberBundle參數中MemberPatient，其中包含必要的 CoverageToMatch、和 Consent 資源，以及選用的 CoverageToLink。

```
POST [base]/Group/$bulk-member-match
Content-Type: application/fhir+json
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "MemberBundle",
      "part": [
        {
```

```
"name": "MemberPatient",
"resource": {
  "resourceType": "Patient",
  "identifier": [
    {
      "system": "http://example.org/patient-id",
      "value": "patient-0"
    }
  ],
  "name": [
    {
      "family": "Smith",
      "given": ["James"]
    }
  ],
  "gender": "male",
  "birthDate": "1950-01-01"
}
},
{
  "name": "CoverageToMatch",
  "resource": {
    "resourceType": "Coverage",
    "status": "active",
    "identifier": [
      {
        "system": "http://example.org/coverage-id",
        "value": "cov-0"
      }
    ],
    "subscriberId": "sub-0",
    "beneficiary": {
      "reference": "Patient/patient123"
    },
    "relationship": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/subscriber-
relationship",
          "code": "self"
        }
      ]
    }
  },
  "payor": [
```

```
        {
          "reference": "Organization/org123"
        }
      ]
    }
  },
  {
    "name": "Consent",
    "resource": {
      "resourceType": "Consent",
      "status": "active",
      "scope": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/consentscope",
            "code": "patient-privacy"
          }
        ]
      },
      "category": [
        {
          "coding": [
            {
              "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
              "code": "IDSCCL"
            }
          ]
        }
      ],
      "patient": {
        "reference": "Patient/patient123"
      },
      "performer": [
        {
          "reference": "Patient/patient123"
        }
      ],
      "sourceReference": {
        "reference": "http://example.org/DocumentReference/consent-source"
      },
      "policy": [
        {
          "uri": "http://hl7.org/fhir/us/davinci-hrex/StructureDefinition-hrex-consent.html#regular"
        }
      ]
    }
  }
]
```

```

    }
  ],
  "provision": {
    "type": "permit",
    "period": {
      "start": "2024-01-01",
      "end": "2025-12-31"
    },
  },
  "actor": [
    {
      "role": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/provenance-
participant-type",
            "code": "performer"
          }
        ]
      },
      "reference": {
        "identifier": {
          "system": "http://hl7.org/fhir/sid/us-npi",
          "value": "9876543210"
        },
        "display": "Old Health Plan"
      }
    },
    {
      "role": {
        "coding": [
          {
            "system": "http://terminology.hl7.org/CodeSystem/v3-
ParticipationType",
            "code": "IRCP"
          }
        ]
      },
      "reference": {
        "identifier": {
          "system": "http://hl7.org/fhir/sid/us-npi",
          "value": "0123456789"
        },
        "display": "New Health Plan"
      }
    }
  ]
}

```

```

    }
  ],
  "action": [
    {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/consentaction",
          "code": "disclose"
        }
      ]
    }
  ]
}
},
{
  "name": "CoverageToLink",
  "resource": {
    "resourceType": "Coverage",
    "status": "active",
    "identifier": [
      {
        "system": "http://example.org/coverage-link-id",
        "value": "cov-link-0"
      }
    ],
    "subscriberId": "new-sub-0",
    "beneficiary": {
      "reference": "Patient/patient123"
    },
    "relationship": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/subscriber-
relationship",
          "code": "self"
        }
      ]
    },
    "payor": [
      {
        "identifier": {
          "system": "http://hl7.org/fhir/sid/us-npi",
          "value": "0123456789"
        }
      }
    ]
  }
}
}

```

```

        },
        "display": "New Health Plan"
      }
    ]
  }
}
]
}
]
}
}

```

使用輸出完成任務回應

當任務完成時，回應會包含任務中繼資料和 FHIR 參數資源，其中包含分類相符結果的三個群組資源。

```

{
  "datastoreId": "datastoreId",
  "jobId": "jobId",
  "status": "COMPLETED",
  "submittedTime": "2026-03-20T18:45:26.321Z",
  "numberOfMembers": 3,
  "numberOfMembersProcessedSuccessfully": 3,
  "numberOfMembersWithCustomerError": 0,
  "numberOfMembersWithServerError": 0,
  "output": {
    "resourceType": "Parameters",
    "meta": {
      "profile": [
        "http://hl7.org/fhir/us/davinci-pdex/StructureDefinition/pdex-parameters-multi-member-match-bundle-out"
      ]
    },
    "parameter": [
      {
        "name": "MatchedMembers",
        "resource": {
          "resourceType": "Group",
          "id": "group1",
          "text": {
            "status": "generated",
            "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Matched members group</div>"
          },
          "contained": [

```

```
{
  "resourceType": "Patient",
  "id": "1",
  "identifier": [
    {
      "system": "http://example.org/patient-id",
      "value": "patient-0"
    }
  ],
  "name": [
    {
      "family": "Smith",
      "given": ["James"]
    }
  ],
  "gender": "male",
  "birthDate": "1950-01-01"
}
],
"type": "person",
"actual": true,
"code": {
  "coding": [
    {
      "system": "http://hl7.org/fhir/us/davinci-pdex/CodeSystem/
PdexMultiMemberMatchResultCS",
      "code": "match",
      "display": "Matched"
    }
  ]
},
"quantity": 1,
"member": [
  {
    "entity": {
      "extension": [
        {
          "url": "http://hl7.org/fhir/us/davinci-pdex/StructureDefinition/
base-ext-match-parameters",
          "valueReference": {
            "reference": "#1"
          }
        }
      ]
    }
  }
],
```

```

        "reference": "Patient/patient123"
      }
    }
  ]
}
},
{
  "name": "NonMatchedMembers",
  "resource": {
    "resourceType": "Group",
    "id": "Group2",
    "text": {
      "status": "generated",
      "div": "<div xmlns=\\"http://www.w3.org/1999/xhtml\\">Non-matched members
group</div>"
    },
    "contained": [
      {
        "resourceType": "Patient",
        "id": "1",
        "identifier": [
          {
            "system": "http://example.org/patient-id",
            "value": "patient-501"
          }
        ],
        "name": [
          {
            "family": "Carter",
            "given": ["Emily"]
          }
        ],
        "gender": "female",
        "birthDate": "1985-06-15"
      }
    ],
    "type": "person",
    "actual": true,
    "code": {
      "coding": [
        {
          "system": "http://hl7.org/fhir/us/davinci-pdex/CodeSystem/
PdexMultiMemberMatchResultCS",
          "code": "nomatch",

```

```

        "display": "Not Matched"
      }
    ]
  },
  "quantity": 1,
  "member": [
    {
      "entity": {
        "extension": [
          {
            "url": "http://hl7.org/fhir/us/davinci-pdex/StructureDefinition/
base-ext-match-parameters",
            "valueReference": {
              "reference": "#1"
            }
          }
        ],
        "reference": "Patient/patient123"
      }
    }
  ]
}
},
{
  "name": "ConsentConstrainedMembers",
  "resource": {
    "resourceType": "Group",
    "id": "group3",
    "text": {
      "status": "generated",
      "div": "<div xmlns=\"http://www.w3.org/1999/xhtml\">Consent constrained
members group</div>"
    },
    "contained": [
      {
        "resourceType": "Patient",
        "id": "1",
        "identifier": [
          {
            "system": "http://example.org/patient-id",
            "value": "patient-502"
          }
        ],
        "name": [

```


HealthLake 如何將成員分類為輸出群組

在`$bulk-member-match`請求中提交的每個成員都會透過循序管道進行評估。每個步驟的結果會決定成員所在的輸出群組。

1. 結構驗證 — MemberBundle 是否符合必要的設定檔？失敗時：錯誤（不是在任何群組中）。
2. 患者比對 — HealthLake 可以找到符合所提交人口統計特性的患者嗎？失敗時：NonMatchedMembers。
3. 涵蓋範圍確認 — HealthLake 是否可以縮小到只有一位具有有效 CoverageToMatch 的患者？失敗時：NonMatchedMembers。
4. 同意評估 - 提交的同意現在是否可以接受？（狀態 = 作用中，期間涵蓋目前日期，可以驗證執行者）。失敗時：ConsentConstrainedMembers。
5. 成功 — 所有檢查都通過。儲存在資料存放區的同意。放置在 MatchedMembers 中的成員。

金鑰原則：成員只能出現在一個目的地中。第一個失敗步驟決定放置位置。未通過步驟 2 或 3 的成員絕不會置於 ConsentConstrainedMembers 中 — 該群組僅適用於成功配對但無法接受其同意的成員。

同意評估詳細資訊（步驟 4）：

- 檢查 1 — 同意狀態：是否`Consent.status`等於「作用中」？如果不是 → ConsentConstrainedMembers。
- 檢查 2 — 佈建期間：是否`provision.period`涵蓋目前日期？如果目前日期在`period.end` → ConsentConstrainedMembers 之前`period.start`或之後。
- 檢查 3 — 執行者驗證：是否可以驗證`Consent.performer`參考？如果在資料存放區中找不到參考的資源，或未與相符的患者相關聯 → ConsentConstrainedMembers。

所有檢查都必須通過，才能將成員放置在 MatchedMembers 中，並儲存同意。

涵蓋範圍比對行為

在成員比對期間，只會根據回應付款人的資料存放區進行 CoverageToMatch 驗證。

CoverageToLink 屬於新的/請求的付款人，不會根據舊付款人的資料存放區進行驗證。在請求 CoverageToLink 中包含 不會影響相符的結果。

請求中的每個病患 + 涵蓋範圍組合都會獨立處理。同一個病患可以使用不同的涵蓋範圍計劃多次提交，而且每個項目都會根據其特定涵蓋範圍收到自己的結果。

同意執行者參考處理

新的付款人可能會在 `中` 傳送臨時或本機患者參考 `Consent.performer` (例如, 與 `中` 使用的相同參考 `Consent.patient`)。HealthLake 會自動解析這些參考：

- 如果 `Consent.performer` 包含與 `中` 相同的本機參考 `Consent.patient` , 則 HealthLake 會在相符成功後將其取代為實際相符的患者參考。
- HealthLake 支援 `Patient`、`RelatedPerson`、`Practitioner`、`PractitionerRole` 和 `Organization` 類型的執行器參考 (包括直接參考和邏輯識別符參考) 。
- 如果執行者驗證失敗 (資源找不到或與相符的患者沒有關聯) , 成員會放置在 `ConsentConstrainedMembers` 中 , 而不是傳回錯誤。

輸出群組資源

已完成的任務會傳回包含三個群組資源的參數資源：

MatchedMembers 群組

包含所有成功配對成員的病患參考, 其同意在請求時為作用中且有效。系統會為每個相符的成員建立並存放資料存放區中的同意資源。此群組會在資料存放區中執行個體化, 可直接與 `搭配使用` `$davinci-data-export`。

NonMatchedMembers 群組

包含對找不到唯一相符項目之成員的參考。當資料存放區中沒有患者符合提供的人口統計特性、任何相符患者候選者沒有有效的涵蓋範圍, 或多個患者符合人口統計特性, 且多個患者具有有效的涵蓋範圍 (模稜兩可) 時, 就會放置成員。

ConsentConstrainedMembers 群組

包含成功配對 (人口統計特性和涵蓋範圍已確認) 但請求時無法履行同意之成員的患者參考。不會為受同意限制的成員存放 同意資源。相符的成員身分 (`MemberIdentifier` 和 `MemberId`) 仍包含在內, 因此請求付款人知道誰受到限制。

`Group.quantity` 欄位包含其各自群組中的成員總數。

群組成員參考：

- `Group.member.entity.reference` — 對於 `MatchedMembers` 和 `ConsentConstrainedMembers` , 包含回應付款人系統中相符成員的患者 ID。對於 `NonMatchedMembers` , 會參考包含的輸入病患。

- `Group.member.entity.extension (base-ext-match-parameters)` — 包含來自原始輸入請求的患者 ID (請求付款人提交的 ID, 衍生自 `Patient.id`、`Coverage.beneficiary.reference` 或 `Consent.patient.reference`)。

同意病患連結

Important

儲存的同意資源會完全依照請求付款人所提交的內容來保留病患參考。HealthLake 不會自動更新同意的患者欄位，以指向接收資料存放區中相符的患者。

若要將儲存的同意連結至相符的病患，請使用任務輸出：`MatchedMembers` 群組中的每個成員都有 `member.entity.reference` 指向相符的病患，以及指向包含的輸入病患 `member.entity.extension(base-ext-match-parameters)`。將這些與同意的病患欄位交叉參考，以在應用程式層中建置映射。

儲存的內容與暫時性的內容

下表記錄 HealthLake 在 `$bulk-member-match` 處理期間保留到資料存放區的內容，以及僅存在於任務回應中的內容：

資源	已存放？	可透過 REST 查詢？	備註
MemberPatient (輸入)	否	否	僅用於比對；不保留
CoverageToMatch (輸入)	否	否	僅用於範圍確認
CoverageToLink (輸入)	否	否	未針對資料存放區進行驗證；屬於新的付款人
同意 (相符的成員)	是	是 — GET <code>【base】 / Consent/{id}</code>	從請求付款人收到時儲存

資源	已存放？	可透過 REST 查詢？	備註
同意 (受限成員)	否	否	未儲存。回應中仍包含成員身分。
MatchedMembers 群組 (輸出)	是	是 — GET 【base】 / Group/{id}	執行個體化；可與 \$davinci-data-export 搭配使用
NonMatchedMembers 群組	否	否	僅限任務回應
ConsentConstrained Members 群組	否	否	僅限任務回應

與 \$davinci-data-export 整合

傳回的 MatchedMembers 群組資源 \$bulk-member-match 可以直接與 \$davinci-data-export 操作搭配使用，以擷取大量成員資料：

```
POST [base]/Group/{matched-group-id}/$davinci-data-export
GET [base]/Group/{matched-group-id}
```

此整合可讓您在初次大量識別相符成員的高效工作流程，然後使用產生的群組資源匯出其完整的運作狀態記錄。

匯出前使用 \$member-remove

如果您需要在相符之後排除特定成員匯出（例如，成員撤銷相符和匯出之間的同意），請在 MatchedMembers 群組 \$member-remove 上使用。

Important

透過 移除成員會在群組中將成員 \$member-remove 標記為非作用中，但 \$davinci-data-export 只有在群組更新為「最終」狀態後才會排除非作用中成員。如果您在仍然具有預設狀態的群組 \$davinci-data-export 上呼叫，移除的成員仍可能會出現在匯出結果中。

工作流程：

1. POST [base]/Group/{id}/\$member-remove — 將成員標記為非作用中
2. PUT [base]/Group/{id} — 將群組狀態更新為「最終」
3. POST [base]/Group/{id}/\$davinci-data-export — 匯出現在排除已移除的成員

效能特性

\$bulk-member-match 操作專為大量處理而設計，並以非同步方式執行：

- 並行：每個資料存放區最多 5 個並行操作。
- 可擴展性：每個請求最多可處理 500 個成員 (5 MB 承載限制)。
- 平行操作：與並行匯入、匯出或大量刪除操作相容。

Authorization

API 在 FHIR 授權通訊協定上使用 SMART 搭配下列必要範圍：

- system/Patient.read — 搜尋和比對患者資源時必填。
- system/Coverage.read — 驗證涵蓋範圍資訊時需要。
- system/Group.write — 建立結果群組資源時需要。
- system/Organization.read — 有條件，如果涵蓋範圍參考組織則為必要。
- system/Practitioner.read — 有條件，如果涵蓋範圍參考從業人員則為必要。
- system/PractitionerRole.read — 有條件，如果涵蓋範圍參考從業人員角色則為必要。
- system/Consent.write — 有條件，如果提供同意資源則為必要。

此操作也支援程式設計存取的 AWS IAM Signature 第 4 版 (SigV4) 授權。

驗證規則

下列驗證規則會套用至步驟 1 的每個 MemberBundle。未通過驗證的成員會報告為錯誤，不會出現在任何輸出群組中。

MemberPatient

欄位	HealthLake 如何使用它	驗證失敗，如果...
name.family	人口統計搜尋	缺少
name.given	人口統計搜尋	缺少 (至少需要一個)
birthDate	人口統計搜尋	缺少
gender	人口統計搜尋；如果不存在，則使用 Birth Sex 延伸	性別和性別不存在 (hrex-pat-1)
identifier	存在時包含在搜尋中；提高可信度	永遠不會導致失敗 (選用)

CoverageToMatch / CoverageToLink

欄位	HealthLake 如何使用它	驗證失敗，如果...
status	確認涵蓋範圍是可行的	缺少
beneficiary	將涵蓋範圍連結至病患候選項目	缺少
payor	存在多個候選項目時的歧義	缺少或多個付款人
relationship	確認訂閱者與受益人的關係	缺少
identifier (MB) or subscriberId	主要歧義金鑰	兩者皆不存在

同意

欄位	HealthLake 如何使用它	驗證失敗，如果...
scope	確認同意範圍為患者隱私	缺少或沒有病患隱私代碼
category	確認揭露分類	缺少

欄位	HealthLake 如何使用它	驗證失敗，如果...
patient	識別同意主體	缺少
performer	識別誰同意	缺少
sourceReference	記錄同意來源	缺少
policy.uri	決定資料共用範圍	遺失或 URI 結尾不是 #regular 或 #sensitive
provision.type	必須為每個 HReX 同意設定檔的「允許」	遺失或沒有「許可」（包括「拒絕」）
provision.period	在步驟 4 評估的同意限制檢查	遺失或沒有開始/結束
status	在步驟 4（非步驟 1）評估	永遠不會造成步驟 1 失敗 — HealthLake 接受任何有效狀態，並在步驟 4 進行評估

Note

HReX 同意設定檔會定義固定值為「作用中」的狀態。HealthLake 刻意放寬此限制，讓非作用中的同意收到有意義的分類 (ConsentConstrainedMembers)，而不是拒絕括號驗證。

相符行為

- 病患搜尋（步驟 2）— HealthLake 使用 name.family + name.given（確切、不區分大小寫）、birthDate（確切）、gender（確切；如果性別不存在，則使用 birth Sex）和 identifier（如果存在，則為選用）進行搜尋。
- 覆蓋範圍歧義（步驟 3）— 找到多個病患候選項目時，CoverageToMatch 會用來縮小至一個。當在 subscriberId 或 identifier (MB 類型) 和 上比對的資料存放區中存在作用中的涵蓋範圍資源時，涵蓋範圍「有效」payor。
- 同意評估（步驟 4）— 僅在成功的唯一相符項目後執行。請參閱上述同意評估詳細資訊一節。

錯誤處理

操作會處理下列錯誤條件：

- 400 錯誤的請求：無效的請求格式、缺少必要的參數或承載超過大小限制 (500 個成員或 5 MB)。
- 422 無法處理的實體：在任務執行期間處理錯誤。
- 個別成員錯誤：當特定成員無法處理時，操作會繼續保留剩餘成員，並在 NonMatchedMembers 群組中包含錯誤詳細資訊，並具有適當的原因代碼。例如，MemberBundle 遺失 birthDate 參數之病患所在的 會傳回下列錯誤：

```
"errors": [  
  {  
    "memberIndex": 1,  
    "jsonBlob": {  
      "resourceType": "OperationOutcome",  
      "issue": [  
        {  
          "severity": "error",  
          "code": "invalid",  
          "diagnostics": "MemberPatient.birthDate is required"  
        }  
      ],  
      "statusCode": 400  
    }  
  }  
]
```

錯誤詳細資訊可透過狀態輪詢端點取得，並包含：

- numberOfMembersWithCustomerError：發生驗證或輸入錯誤的成員計數。
- numberOfMembersWithServerError：發生伺服器端處理錯誤的成員計數。

相關的 操作

- [the section called “\\$member-match”](#) — 個別成員比對操作。
- [the section called “\\$davinci-data-export”](#) — 使用群組資源大量匯出資料。
- [the section called “\\$操作”](#) — 支援操作的完整清單。

HealthLake 的 FHIR R4 `$confirm-attribution-list` 操作

向生產者指出消費者不再需要對屬性清單進行變更。此操作會透過從清單中移除非作用中的成員、將狀態變更為「最終」，以及確認操作，來完成歸因清單。此操作是 [FHIR 成員屬性 \(ATR\) List IG 2.1.0 實作的一部分](#)。

Endpoint

```
POST [base]/Group/[id]/$confirm-attribution-list
```

請求

不需要參數。

```
POST [base]/Group/[id]/$confirm-attribution-list
```

回應

傳回 HTTP 201 與確認訊息。

回應範例

```
HTTP Status Code: 201
```

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "message",
      "valueString": "Accepted."
    }
  ]
}
```

確認後的群組狀態

確認成功後，群組資源的屬性清單狀態會設為「最終」：

```
{
  "resourceType": "Group",
  "id": "fullexample",
  "extension": [
```

```

    {
      "url": "http://hl7.org/fhir/us/davinci-atr/StructureDefinition/ext-
      attributionListStatus",
      "valueCode": "final"
    }
  ]
  // ... other Group properties
}

```

錯誤處理

操作會處理下列錯誤條件：

錯誤	HTTP 狀態	Description
無效的操作請求	400	不符合的請求參數或結構
找不到群組資源	404	指定的群組 ID 不存在

授權和安全性

SMART 範圍需求

用戶端必須具有適當的權限，才能讀取群組資源和相關歸因資源

對於 `$confirm-attribution-list`，用戶端也必須具有寫入權限才能修改群組資源

標準 FHIR 授權機制適用於所有操作

HealthLake 的 FHIR R4 `$davinci-data-export` 操作

`$davinci-data-export` 操作是一種非同步 FHIR 操作，可用來從中匯出醫療保健資料 AWS HealthLake。此操作支援多種匯出類型，包括成員屬性 (ATR)、PDex 提供者存取、Payer-to-Payer 和成員存取 APIs。它是標準 FHIR `$export` 操作的特殊版本，旨在符合 DaVinci 實作指南的要求。

主要功能

- 非同步處理：遵循標準 FHIR 非同步請求模式
- 群組層級匯出：匯出特定群組資源中成員的資料
- 多種匯出類型：支援 ATR（成員屬性）、PDex 提供者存取、Payer-to-Payer 和成員存取 APIs

- 全方位設定檔支援：包括美國核心、CARIN 藍色按鈕和 PDex 設定檔
- 彈性篩選：支援依病患、資源類型和時間範圍進行篩選
- NDJSON 輸出：以換行分隔 JSON 格式提供資料

操作端點

```
GET [base]/Group/[id]/$davinci-data-export
POST [base]/Group/[id]/$davinci-data-export
```

請求參數

參數	基數	Description
patient	0..*	要匯出其資料的特定成員。省略時，會匯出群組中的所有成員。
_type	0..1	要匯出的以逗號分隔的 FHIR 資源類型清單。省略時，會包含指定匯出類型的所有支援資源類型。對於 ATR 匯出，這預設為 8 個屬性資源類型。對於 PDex 匯出，這包括所有屬性資源類型，以及美國核心、CARIN 藍色按鈕和 PDex 設定檔中的臨床和宣告資源類型。
_since	0..1	僅包含在此日期和時間之後更新的資源。
_until	0..1	僅包含在此日期和時間之前更新的資源。
exportType	0..1	要執行的匯出類型。有效值：h17.fhir.us.davinci-atr、h17.fhir.us.davinci-pdex、h17.fhir.us.davinci-pdex.p2p、h17.fhir.us.davinci-pdex.member。預設：h17.fhir.us.davinci-atr。
_includeExplanationOfBenefit	0..1	指定是否要在移除財務資料時包含 CARIN BB 2.x ExplanationOfBenefit 資源。預設：false。

參數	基數	Description
<code>_security</code>	0..*	依 <code>meta.security</code> 編碼值篩選匯出的資源。使用 <code>system code</code> 格式 (管道字元必須以 URL 編碼為 <code>%7C</code>)。提供多個值時，資源必須符合所有值 (AND 語意)。使用 <code>system </code> (追蹤管道、無程式碼) 來比對來自指定系統的任何程式碼。
<code>_tag</code>	0..*	依 <code>meta.tag</code> 編碼值篩選匯出的資源。使用與相同的 <code>system code</code> 格式和 AND 語意 <code>_security</code> 。同時指定 <code>_tag_security</code> 和 <code>_security</code> 時，資源必須符合兩個篩選條件。

支援的資源類型

支援的資源類型取決於您指定的匯出類型。對於 ATR 匯出，支援下列資源類型：

- Group
- Patient
- Coverage
- RelatedPerson
- Practitioner
- PractitionerRole
- Organization
- Location

對於 PDex 匯出 (Provider Access、Payer-to-Payer 和 Member Access)，除了上述類型之外，還支援所有臨床和宣告資源類型。如需支援資源類型的完整清單，請參閱 [美國核心實作指南 \(STU 6.1\)](#)、[CARIN 藍色按鈕實作指南](#)，以及 [Da Vinci 預先授權支援實作指南](#)。

匯出類型

`$davinci-data-export` 操作支援下列匯出類型。您可以使用 `exportType` 參數指定匯出類型。

匯出類型	用途	資料範圍	時間限制
hl7.fhir.us.davinci-atr	成員屬性清單	屬性相關資源	無
hl7.fhir.us.davinci-pdex	提供者存取 API	屬性患者的臨床和宣告資料	5 年
hl7.fhir.us.davinci-pdex.p2p	Payer-to-Payer 交換	保險轉換的歷史成員資料	5 年
hl7.fhir.us.davinci-pdex.member	成員存取 API	成員自己的運作狀態資料	5 年

Note

對於 PDex 匯出，5 年時間限制不適用於 ATR 資源類型

(Group、Patient、Coverage、RelatedPerson、Practitioner、PractitionerRole、Organization)

無論年齡為何，一律包含這些資源。

ATR (hl7.fhir.us : //davinci-atr)

使用 ATR 匯出類型，您可以匯出成員屬性清單資料。使用此匯出類型來擷取群組內成員的屬性相關資源。如需詳細資訊，請參閱 [Da Vinci ATR 匯出操作](#)。

支援的資源類型

Group, Patient, Coverage, RelatedPerson, Practitioner, PractitionerRole, Organization, Location

時間篩選

不會套用時間篩選。無論日期為何，都會匯出所有相符的資源。

PDex 匯出類型

所有 PDex 匯出類型共用相同的支援設定檔和篩選邏輯。如需詳細資訊，請參閱 [Da Vinci PDex 提供者存取 API](#)。支援下列設定檔：

- 美國核心 3.1.1、6.1.0 和 7.0.0
- PDex 事前授權 (不支援成員存取)
- CARIN BB 2.x 基礎描述檔：住院機構、門診機構、專業NonClinician、口頭、藥學

對於 PDex 匯出，會自動為群組中的每個病患探索臨床和宣告資源。您不需要在群組資源中明確參考這些資源。操作會搜尋屬於歸因病患的所有病患室資源 Condition Coverage (例如 Observation、MedicationRequest、RelatedPerson、和 ExplanationOfBenefit)。只有 Patient、Group和非non-patient-compartment ATR 類型 (Practitioner、PractitionerRole、Organization、Location) 需要群組中的明確參考。

提供者存取 (h17.fhir.us.davinci-pdex)

讓網路內提供者擷取歸因患者的患者資料。

Payer-to-Payer(h17.fhir.us.davinci-pdex.p2p)

當患者變更保險時，啟用付款人之間的資料交換。

成員存取 (h17.fhir.us.davinci-pdex.member)

可讓成員存取自己的運作狀態資料。此匯出類型可能包含宣告資源中的財務資料。

設定檔支援和包含邏輯

對於 PDex 匯出，\$davinci-data-export操作會使用 meta.profile元素中的設定檔宣告來決定匯出中要包含哪些資源。

ExplanationOfBenefit 資源處理

ExplanationOfBenefit (EOB) 資源會根據其meta.profile宣告從 PDex 匯出中包含或排除：

- 具有 CARIN BB 1.x 設定檔的 ExplanationOfBenefit 資源會從匯出中排除。
- 匯出會排除未設定 ExplanationOfBenefit 資源。meta.profile
- 一律包含具有 CARIN BB 2.x Basis 描述檔的 ExplanationOfBenefit 資源。
- 根據預設，會排除具有包含財務資料的 CARIN BB 2.x 設定檔的 ExplanationOfBenefit 資源。設定 `_includeEOB2xWoFinancial=true` 時，它們會包含在財務資料分割中，並將資源轉換為對應的基礎設定檔。
- 一律包含具有 PDex 預先授權設定檔的 ExplanationOfBenefit 資源。

財務資料轉換

當您設定 `時_includeE0B2xWoFinancial=true`，此操作會透過移除財務資料，將 [CARIN BB 2.x ExplanationOfBenefit](#) 資源轉換為其對應的 Basis 設定檔。例如，`C4BB ExplanationOfBenefit Oral`資源會轉換為 `C4BB ExplanationOfBenefit Oral Basis`，這會根據 FHIR 規格從記錄分割財務資料。

下列財務資料元素會在轉換期間移除：

- `total` 元素上的所有切片
- 具有 `amounttype` 配量的所有 `adjudication` 元素
- 包含金額資訊的所有 `item.adjudication` 元素

此操作也會在轉換期間更新設定檔中繼資料：

- `meta.profile` 已更新為 Basis 設定檔正式 URL
- 版本已更新為 CARIN BB 2.x Basis 版本
- 不會修改資料存放區中的現有資源
- 匯出的資源不會保留回資料存放區

設定檔偵測規則

操作使用下列規則來偵測和驗證設定檔：

- 版本偵測是以 `meta.profile` 正式 URLs 為基礎
- 如果任何宣告的設定檔符合匯出條件，則會包含資源
- 設定檔驗證會在匯出處理期間進行

PDex 匯出的五年時間篩選

對於所有 PDex 匯出類型，HealthLake 會根據資源上次更新的時間套用 5 年的時間篩選條件。暫時篩選條件適用於所有資源，但下列核心屬性資源類型除外，無論使用期限為何，一律會匯出這些資源：

- Patient
- Coverage
- Organization
- Practitioner

- PractitionerRole
- RelatedPerson
- Location
- Group

這些管理和人口統計資源是豁免的，因為它們提供匯出資料的基本內容。ATR 匯出不受任何時間篩選的限制。

請求範例

下列範例示範如何開始不同匯出類型的匯出任務。

ATR 匯出

```
GET https://healthlake.{region}.amazonaws.com/datastore/
{datastoreId}/r4/Group/example-group/$davinci-data-export?
_type=Group, Patient, Coverage, Practitioner, Organization&exportType=hl7.fhir.us.davinci-
atr

POST https://healthlake.{region}.amazonaws.com/datastore/
{datastoreId}/r4/Group/example-group/$davinci-data-export?
_type=Group, Patient, Coverage, Practitioner, Organization&exportType=hl7.fhir.us.davinci-
atr
Content-Type: application/json

{
  "DataAccessRoleArn": "arn:aws:iam::444455556666:role/your-healthlake-service-role",
  "JobName": "attribution-export-job",
  "OutputDataConfig": {
    "S3Configuration": {
      "S3Uri": "s3://your-export-bucket/EXPORT-JOB",
      "KmsKeyId":
"arn:aws:kms:region:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  }
}
```

使用 ExplanationOfBenefit 財務資料移除來匯出供應商存取

```
GET https://healthlake.{region}.amazonaws.com/datastore/
{datastoreId}/r4/Group/example-group/$davinci-data-export?
```

```
_type=Patient,Observation,Condition,MedicationRequest,ExplanationOfBenefit&exportType=h17.fhir.pdex&_includeE0B2xWoFinancial=true
```

Payer-to-Payer 匯出

```
GET https://healthlake.{region}.amazonaws.com/datastore/{datastoreId}/r4/Group/example-group/$davinci-data-export?_type=Patient,Coverage,ExplanationOfBenefit,Condition,Procedure&exportType=h17.fhir.us.davinci.pdex.p2p&_includeE0B2xWoFinancial=true
```

特定病患的成員存取匯出

```
GET https://healthlake.{region}.amazonaws.com/datastore/{datastoreId}/r4/Group/example-group/$davinci-data-export?_type=Patient,Observation,Condition,ExplanationOfBenefit,MedicationRequest&exportType=h17.fhir.pdex.member&patient=Patient/example-patient-id
```

回應範例

```
{
  "datastoreId": "eae622d8406b41eb86c0f4741201ff9",
  "jobStatus": "SUBMITTED",
  "jobId": "48d7b91dae4a64d00d54b70862f33f61"
}
```

資源關係

操作會根據其在成員屬性清單中的關係匯出資源：

```
Group (Attribution List)
### Patient (Members)
### Coverage # RelatedPerson (Subscribers)
### Practitioner (Attributed Providers)
### PractitionerRole # Location
### Organization (Attributed Providers)
```

Note

上述資源關係圖適用於 ATR 匯出。對於 PDex 匯出，臨床和宣告資源會透過病患搜尋自動探索，且不需要群組資源中的明確參考。

資源來源

資源	來源位置	Description
Patient	Group.member.entity	屬於屬性清單成員的患者
Coverage	Group.member.extension:coverageReference	導致病患成員資格的涵蓋範圍
Organization	Group.member.extension:attributedProvider	患者所屬的組織
Practitioner	Group.member.extension:attributedProvider	患者所屬的個別從業人員
PractitionerRole	Group.member.extension:attributedProvider	患者所屬的從業人員角色
RelatedPerson	Coverage.subscriber	涵蓋範圍的訂閱者
Location	PractitionerRole.location	與從業人員角色相關聯的位置
Group	輸入端點	屬性清單本身

任務管理

檢查任務狀態

```
GET [base]/export/[job-id]
```

取消任務

```
DELETE [base]/export/[job-id]
```

任務生命週期

- SUBMITTED - 已收到任務並排入佇列
- IN_PROGRESS - 任務正在主動處理
- COMPLETED - 任務成功完成，檔案可供下載

- FAILED - 任務發生錯誤

輸出格式

- 檔案格式：NDJSON (換行分隔 JSON)
- 檔案組織：每個資源類型的個別檔案
- 副檔名：.ndjson
- 位置：指定的 S3 儲存貯體和路徑

錯誤處理

針對下列情況，操作會傳回具有 `OperationOutcome` 的 HTTP 400 錯誤請求：

授權錯誤

在中指定的 IAM 角色 `DataAccessRoleArn` 沒有足夠的許可來執行匯出操作。如需所需 S3 和 KMS 許可的完整清單，請參閱 [設定匯出任務的許可](#)。

參數驗證錯誤

- `patient` 參數的格式不是 `Patient/id,Patient/id,...`
- 一或多個病患參考無效或不屬於指定的群組
- `exportType` 參數值不是支援的匯出類型
- `_type` 參數包含指定匯出類型不支援的資源類型
- `_type` 參數缺少 `hl7.fhir.us.davinci-atr` 匯出類型所需的資源類型 (Group、Patient、Coverage)
- `_includeE0B2xWoFinancial` 參數值不是有效的布林值

資源驗證錯誤

- 指定的群組資源不存在於資料存放區中
- 指定的群組資源沒有成員
- 一或多個群組成員參考資料存放區中不存在的患者資源

安全性和授權

- 適用標準 FHIR 授權機制

- 資料存取角色必須具有 S3 和 KMS 操作所需的 IAM 許可。如需所需許可的完整清單，請參閱[設定匯出任務的許可](#)。

最佳實務

- 資源類型選擇：僅請求所需的資源類型，以將匯出大小和處理時間降至最低
- 以時間為基礎的篩選：使用 `_since` 參數進行增量匯出
- 病患篩選：當您只需要特定成員的資料時，請使用 `patient` 參數
- 任務監控：定期檢查大型匯出的任務狀態
- 錯誤處理：針對失敗的任務實作適當的重試邏輯
- 暫時篩選條件感知：對於 PDex 匯出，請在選取資源類型時考慮 5 年暫時篩選條件
- 財務資料移除：當您需要沒有財務資訊的宣告資料 `_includeE0B2xWoFinancial=true` 時使用
- 設定檔管理：確保資源具有適當的設定檔宣告、在擷取之前針對目標設定檔進行驗證，並使用設定檔版本控制來控制匯出行為

限制

- 參數中最多可指定 500 名病患 `patient`
- 匯出僅限於群組層級操作
- 僅支援每個匯出類型的預先定義資源類型集
- 輸出一律為 NDJSON 格式
- PDex 匯出僅限於 5 年的臨床和宣告資料
- 財務資料轉換僅適用於 CARIN BB 2.x ExplanationOfBenefit 設定檔

其他資源

- [Da Vinci 成員屬性清單 IG](#)
- [Da Vinci 付款人資料交換 IG](#)
- [CARIN 消費者導向付款人資料交換 IG](#)
- [美國核心實作指南](#)
- [FHIR 大量資料存取規格](#)

使用 產生臨床文件 \$document

AWS HealthLake 現在支援合成資源的 \$document 操作，可讓您將合成與其所有參考的資源綁定到單一的整合套件中，以產生完整的臨床文件。此操作對於需要：

- 建立標準化臨床文件
- Exchange 完整患者記錄
- 存放完整的臨床文件
- 產生包含所有相關內容的報告

Usage

您可以使用 GET 和 POST 方法在合成資源上叫用 \$document 操作：

受支援的 操作

```
GET/POST [base]/Composition/[id]/$document
```

支援的參數

HealthLake 支援下列 FHIR \$document 參數：

參數	Type	必要	預設	Description
persist	布林值	否	false	布林值，指出伺服器是否應存放產生的文件套件

範例

GET 請求

```
GET [base]/Composition/180f219f-97a8-486d-99d9-ed631fe4fc57/$document?persist=true
```

具有參數的 POST 請求

```
POST [base]/Composition/180f219f-97a8-486d-99d9-ed631fe4fc57/$document
Content-Type: application/fhir+json
```

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "persist",
      "valueBoolean": true
    }
  ]
}
```

回應範例

操作會傳回「文件」類型的套件資源，其中包含合成和所有參考的資源：

```
{
  "resourceType": "Bundle",
  "id": "180f219f-97a8-486d-99d9-ed631fe4fc57",
  "type": "document",
  "identifier": {
    "system": "urn:ietf:rhc:3986",
    "value": "urn:uuid:0c3151bd-1cbf-4d64-b04d-cd9187a4c6e0"
  },
  "timestamp": "2024-06-21T15:30:00Z",
  "entry": [
    {
      "fullUrl": "http://example.org/fhir/Composition/180f219f-97a8-486d-99d9-ed631fe4fc57",
      "resource": {
        "resourceType": "Composition",
        "id": "180f219f-97a8-486d-99d9-ed631fe4fc57",
        "status": "final",
        "type": {
          "coding": [
            {
              "system": "http://loinc.org",
              "code": "34133-9",
              "display": "Summary of Episode Note"
            }
          ]
        },
        "subject": {
          "reference": "Patient/example"
        }
      }
    }
  ]
}
```

```
    "section": [
      {
        "title": "Allergies",
        "entry": [
          {
            "reference": "AllergyIntolerance/123"
          }
        ]
      }
    ]
  },
  {
    "fullUrl": "http://example.org/fhir/Patient/example",
    "resource": {
      "resourceType": "Patient",
      "id": "example",
      "name": [
        {
          "family": "Smith",
          "given": ["John"]
        }
      ]
    }
  },
  {
    "fullUrl": "http://example.org/fhir/AllergyIntolerance/123",
    "resource": {
      "resourceType": "AllergyIntolerance",
      "id": "123",
      "patient": {
        "reference": "Patient/example"
      },
      "code": {
        "coding": [
          {
            "system": "http://snomed.info/sct",
            "code": "418689008",
            "display": "Allergy to penicillin"
          }
        ]
      }
    }
  }
}
```

```
]
}
```

Behavior (行為)

\$document 操作：

1. 將指定的合成資源做為文件的基礎
2. 識別和擷取合成直接參考的所有資源
3. 將合成和所有參考的資源封裝到類型為「文件」的套件中
4. 當持久性參數設定為 true 時，將產生的文件套件儲存在資料存放區中
5. 識別和擷取 合成間接參考的資源，以產生全面的文件

\$document 操作目前支援以下列格式擷取資源參考：

1. `GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Resource/id`
2. 資源/*id*

合成資源中不支援的資源參考將從產生的文件篩選出。

錯誤處理

操作會處理下列錯誤條件：

- 400 錯誤的請求：無效的 \$document 操作（不符合的請求），或者如果產生的文件在持久性設定為 true 時因篩選掉的參考而失敗 FHIR 驗證
- 找不到 404：找不到合成資源

如需 \$document 操作規格的詳細資訊，請參閱 [FHIR R4 合成 \\$document](#) 文件。

使用 永久移除資源 \$erase

AWS HealthLake 支援 \$erase 操作，可永久刪除特定資源及其歷史版本。當您需要執行下列動作時，此操作特別有用：

- 永久移除個別資源

- 刪除特定版本歷史記錄
- 管理個別資源生命週期
- 符合特定資料移除要求

Usage

操作\$erase可以在兩個層級叫用：

資源執行個體層級

```
POST [base]/[ResourceType]/[ID]/$erase?deleteAuditEvent=true
```

版本特定層級

```
POST [base]/[ResourceType]/[ID]/_history/[VersionID]/$erase
```

Parameters

參數	Type	必要	預設	Description
deleteAuditEvent	布林值	否	false	為 true 時，會刪除相關聯的稽核事件

範例

範例請求

```
POST [base]/Patient/example-patient/$erase
```

回應範例

```
{
  "jobId": "5df47e2f51ff3c731847678cb8cad48e",
}
```

```
"jobStatus": "SUBMITTED"
}
```

任務狀態

若要檢查清除任務的狀態：

```
GET [base]/$erase/[jobId]
```

操作會傳回任務狀態資訊：

```
{
  "datastoreId": "36622996b1fceb7e12ee2ee085308d3",
  "jobId": "5df47e2f51ff3c731847678cb8cad48e",
  "status": "COMPLETED",
  "submittedTime": "2025-10-30T16:39:24.160Z"
}
```

Behavior (行為)

\$erase 操作：

1. 非同步處理以確保資料完整性
2. 維護 ACID 交易
3. 提供任務狀態追蹤
4. 永久移除指定的資源及其版本
5. 包括刪除活動的完整稽核記錄
6. 支援選擇性刪除稽核事件

稽核記錄

\$erase 操作會以 DeleteResource 記錄使用者 ID、時間戳記和資源詳細資訊。

限制

- \$erased 資源不會出現在搜尋結果或_history查詢中。
- 正在清除的資源在處理期間可能暫時無法存取
- 當資源永久移除時，儲存體計量會立即調整

使用 取得患者資料 **Patient/\$everything**

Patient/\$everything 操作用於查詢 FHIR Patient 資源，以及與該 相關的任何其他資源Patient。此操作可用來讓病患存取其整個記錄，或讓提供者執行與病患相關的大量資料下載。HealthLake Patient/\$everything 支援特定病患 id。

Patient/\$everything 是一種 FHIR REST API 操作，可以叫用，如以下範例所示。

GET request

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/$everything
```

Note

回應中的資源會依資源類型和資源 排序id。
回應一律會填入 Bundle.total。

Patient/\$everything 參數

HealthLake 支援下列查詢參數

參數	詳細資訊
入門	取得指定開始日期之後的所有Patient資料。
end	在指定的結束日期之前取得Patient所有資料。
since	在指定日期之後更新Patient所有資料。
_type	取得特定資源類型的Patient資料。
_count	取得Patient資料並指定頁面大小。

Example- 取得指定開始日期之後的所有患者資料

Patient/\$everything 可以使用start篩選條件僅查詢特定日期之後的資料。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/
$everything?start=2024-03-15T00:00:00.000Z
```

Example- 在指定的結束日期之前取得Patient所有資料

病患 \$所有項目都可以使用end篩選條件，只在特定日期之前查詢資料。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/
$everything?end=2024-03-15T00:00:00.000Z
```

Example- 在指定日期之後更新Patient所有資料

Patient/\$everything 可以使用since篩選條件，僅查詢特定日期之後更新的資料。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/
$everything?since=2024-03-15T00:00:00.000Z
```

Example- 取得特定資源類型的Patient資料

病患 \$所有項目都可以使用_type篩選條件來指定要包含在回應中的特定資源類型。您可以在逗號分隔清單中指定多種資源類型。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/
$everything?_type=Observation,Condition
```

Example- 取得Patient資料並指定頁面大小

病患 \$所有項目都可以使用 _count來設定頁面大小。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/id/
$everything?_count=15
```

Patient/\$everything start 和 end 屬性

HealthLake 支援 Patient/ \$everythingstart 和 end 查詢參數的下列資源屬性。

資源	資源元素
帳戶	Account.servicePeriod.start

資源	資源元素
AdverseEvent	AdverseEvent.date
AllergyIntolerance	AllergyIntolerance.recordedDate
約定	Appointment.start
AppointmentResponse	AppointmentResponse.start
AuditEvent	AuditEvent.period.start
基本	Basic.created
BodyStructure	NO_DATE
CarePlan	CarePlan.period.start
CareTeam	CareTeam.period.start
ChargeItem	ChargeItem.occurrenceDateTime, ChargeItem.occurrencePeriod.start, ChargeItem.occurrenceTiming.event
取得	Claim.billablePeriod.start
ClaimResponse	ClaimResponse.created
ClinicalImpression	ClinicalImpression.date
Communication	Communication.sent

資源	資源元素
CommunicationRequest	CommunicationRequest.occurrenceDateTime, CommCommunicationRequest.occurrencePeriod.start
合成	Composition.date
條件	Condition.recordedDate
同意	consent.dateTime
涵蓋範圍	Coverage.period.start
CoverageEligibilityRequest	CoverageEligibilityRequest.created
CoverageEligibilityResponse	CoverageEligibilityResponse.created
DetectedIssue	DetectedIssue.identified
DeviceRequest	DeviceRequest.authoredOn
DeviceUseStatement	DeviceUseStatement.recordedOn
DiagnosticReport	DiagnosticReport.effective
DocumentManifest	DocumentManifest.created
DocumentReference	DocumentReference.context.period.start

資源	資源元素
遇到	Encounter.period.start
EnrollmentRequest	EnrollmentRequest.created
EpisodeOfCare	EpisodeOfCare.period.start
ExplanationOfBenefit	ExplanationOfBenefit.billablePeriod.start
FamilyMemberHistory	NO_DATE
旗標	Flag.period.start
目標	Goal.statusDate
Group	NO_DATE
ImagingStudy	ImagingStudy.started
預防	Immunization.recorded
ImmunizationEvaluation	ImmunizationEvaluation.date
ImmunizationRecommendation	ImmunizationRecommendation.date
發票	Invoice.date
清單	List.date

資源	資源元素
MeasureReport	MeasureReport.period.start
媒體	媒體。發行
MedicationAdministration	MedicationAdministration.effective
MedicationDispense	MedicationDispense.whenPrepared
MedicationRequest	MedicationRequest.authoredOn
MedicationStatement	MedicationStatement.dateAsserted
MolecularSequence	NO_DATE
NutritionOrder	NutritionOrder.dateTime
觀察	Observation.effective
病患	NO_DATE
個人	NO_DATE
程序	Procedure.performed
證明	Provenance.occurredPeriod.start,Provenance.occurredDateTime
QuestionnaireResponse	QuestionnaireResponse.authored

資源	資源元素
RelatedPerson	NO_DATE
RequestGroup	RequestGroup.authoredOn
ResearchSubject	ResearchSubject.period
RiskAssessment	RiskAssessment.occurrenceDateTime,RiskAssessment.occurrencePeriod.start
Schedule	Schedule.planningHorizon
ServiceRequest	ServiceRequest.authoredOn
樣本	配置函式.receivedTime
SupplyDelivery	SupplyDelivery.occurrenceDateTime, SupplyDelivery.occurrencePeriod.start, SupplyDelivery.occurrenceTiming.event
SupplyRequest	SupplyRequest.authoredOn
VisionPrescription	VisionPrescription.dateWritten

使用 擷取 ValueSet 程式碼 `$expand`

AWS HealthLake 現在支援您以客戶身分擷取的 ValueSets (ValueSet) `$expand` 操作，可讓您擷取包含在這些 ValueSet 資源中的代碼完整清單。當您需要：

- 擷取所有可能的程式碼以進行驗證
- 在使用者介面中顯示可用的選項
- 在特定術語內容中執行全面的程式碼查詢

Usage

您可以使用 GET 和 POST 方法在 ValueSet 資源上叫用 \$expand 操作：

受支援的 操作

```
GET/POST [base]/ValueSet/[id]/$expand
GET [base]/ValueSet/$expand?url=http://example.com
POST [base]/ValueSet/$expand
```

支援的參數

HealthLake 支援 FHIR R4 \$expand 參數的子集：

參數	Type	必要	Description
url	uri	否	要展開之 ValueSet 的正式 URL
id	id	否	要展開的 ValueSet 資源 ID (適用於 GET 或 POST 操作)
filter	string	否	篩選程式碼擴展結果
count	integer	否	要傳回的代碼數量
offset	integer	否	傳回前要略過的相符代碼數量。在篩選後套用，且僅適用於相符的代碼，不適用於原始 ValueSet 的完整、未篩選內容

範例

依 ID 的 GET 請求

```
GET [base]/ValueSet/example-valueset/$expand
```

使用篩選條件依 URL 提出 GET 請求

```
GET [base]/ValueSet/$expand?url=http://example.com/ValueSet/my-valueset&filter=male&count=5
```

含參數的 POST 請求 (依 ID)

```
POST [base]/ValueSet/example-valueset/$expand  
Content-Type: application/fhir+json
```

```
{  
  "resourceType": "Parameters",  
  "parameter": [  
    {  
      "name": "count",  
      "valueInteger": 10  
    },  
    {  
      "name": "filter",  
      "valueString": "admin"  
    }  
  ]  
}
```

含參數的 POST 請求 (依 URL)

```
POST [base]/ValueSet/$expand  
Content-Type: application/fhir+json
```

```
{  
  "resourceType": "Parameters",  
  "parameter": [  
    {  
      "name": "url",  
      "valueUri": "http://hl7.org/fhir/ValueSet/administrative-gender"  
    },  
    {  
      "name": "count",  
      "valueInteger": 10  
    }  
  ]  
}
```

回應範例

操作會傳回具有 expansion 元素的 ValueSet 資源，其中包含展開的程式碼：

```
{
  "resourceType": "ValueSet",
  "id": "administrative-gender",
  "status": "active",
  "expansion": {
    "identifier": "urn:uuid:12345678-1234-1234-1234-123456789abc",
    "timestamp": "2024-01-15T10:30:00Z",
    "total": 4,
    "parameter": [
      {
        "name": "count",
        "valueInteger": 10
      }
    ],
    "contains": [
      {
        "system": "http://hl7.org/fhir/administrative-gender",
        "code": "male",
        "display": "Male"
      },
      {
        "system": "http://hl7.org/fhir/administrative-gender",
        "code": "female",
        "display": "Female"
      },
      {
        "system": "http://hl7.org/fhir/administrative-gender",
        "code": "other",
        "display": "Other"
      },
      {
        "system": "http://hl7.org/fhir/administrative-gender",
        "code": "unknown",
        "display": "Unknown"
      }
    ]
  }
}
```

回應包括：

- `expansion.total`：展開 ValueSet 中的程式碼總數
- `expansion.contains`：展開的程式碼陣列及其系統、程式碼和顯示值
- `expansion.parameter`：擴展請求中使用的參數

如需 `$expand` 操作規格的詳細資訊，請參閱 [FHIR R4 ValueSet \\$expand](#) 文件。

使用 FHIR 匯出 HealthLake 資料 `$export`

您可以使用 FHIR `$export` 操作，從 HealthLake 資料存放區大量匯出資料。HealthLake 支援 `$export` 使用 POST 和 GET 請求的 FHIR。若要使用 提出匯出請求 POST，您必須擁有具有必要許可的 IAM 使用者、群組或角色、指定 `$export` 做為請求的一部分，並在請求內文中包含所需的參數。

Note

使用 FHIR 提出的所有 HealthLake 匯出請求 `$export` 都會以 ndjson 格式傳回，並匯出至 Amazon S3 儲存貯體，其中每個 Amazon S3 物件僅包含單一 FHIR 資源類型。您可以根據 AWS 帳戶服務配額將匯出請求排入佇列。如需詳細資訊，請參閱 [Service Quotas](#)。

HealthLake 支援下列三種類型的大量匯出端點請求。

HealthLake 大量 `$export` 類型

匯出類型	Description	語法
系統	從 HealthLake FHIR 伺服器匯出所有資料。	POST <code>https://healthlake. region.amazonaws.com/dat astore/ <i>datastoreId</i> /r4/\$export</code>
所有病患	匯出所有與病患相關的所有資料，包括與病患資源類型相關的資源類型。	POST <code>https://healthlake. region.amazonaws.com/dat astore/ <i>datastoreId</i> /r4/Patient/\$export</code>

匯出類型	Description	語法
		GET https://healthlake. <i>region</i> .amazonaws.com/datastore/ <i>datastoreId</i> /r4/Patient/\$export
病患群組	匯出與群組 ID 所指定之一組患者相關的所有資料。	POST https://healthlake. <i>region</i> .amazonaws.com/dat astore/ <i>datastoreId</i> /r4/Group/ <i>id</i> / \$export GET https://healthlake. <i>region</i> .amazonaws.com/datastore/ <i>datastoreId</i> /r4/Group / <i>id</i> /\$export

開始之前

符合下列要求，以使用適用於 HealthLake 的 FHIR REST API 提出匯出請求。

- 您必須設定具有提出匯出請求必要許可的使用者、群組或角色。如需詳細資訊，請參閱 [授權 \\$export 請求](#)。
- 您必須已建立服務角色，以授予 HealthLake 存取要匯出資料的 Amazon S3 儲存貯體。服務角色也必須指定 HealthLake 做為服務主體。如需設定許可的詳細資訊，請參閱 [設定匯出任務的許可](#)。

授權 \$export 請求

若要使用 FHIR REST API 提出成功的匯出請求，請使用 IAM 或 OAuth2.0。您還必須擁有 服務角色。

使用 IAM 授權請求

當您提出 \$export 請求時，使用者、群組或角色必須在政策中包含 IAM 動作。如需詳細資訊，請參閱 [設定匯出任務的許可](#)。

在 FHIR 上使用 SMART 授權請求 (OAuth 2.0)

當您在啟用 FHIR 的 HealthLake 資料存放區上對 SMART 提出 \$export 請求時，您必須指派適當的範圍。如需詳細資訊，請參閱 [HealthLake 的 FHIR 資源範圍上的 SMART](#)。

Note

\$export 具有GET請求的 FHIR 需要相同的身分驗證方法或承載字符（在 FHIR 上使用 SMART 的情況下）來請求匯出和擷取檔案。使用 FHIR \$export 搭配 匯出的檔案GET可供下載 48 小時。

提出\$export請求

本節說明使用 FHIR REST API 提出匯出請求時必須採取的必要步驟。

為了避免 AWS 您的帳戶意外產生費用，建議您在不提供\$export語法的情況下提出請求來測試您的POST請求。

若要提出請求，您必須執行下列動作：

1. 在支援端點的POST請求 URL \$export中指定。
2. 指定必要的標頭參數。
3. 指定定義必要參數的請求內文。

步驟 1：在支援端點的POST請求 URL \$export中指定。

HealthLake 支援三種類型的大量匯出端點請求。若要提出大量匯出請求，您必須在三個支援的端點之一上提出POST以 為基礎的請求。下列範例示範要在請求 URL \$export中指定何處。

- POST `https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/$export`
- POST `https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Patient/$export`
- POST `https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/Group/id/$export`

您可以在POST請求字串中使用下列支援的搜尋參數。

支援的搜尋參數

HealthLake 在大量匯出請求中支援下列搜尋修飾詞。

下列範例包含特殊字元，在提交請求之前必須先編碼。

名稱	是否為必要？	Description	範例
<code>_outputFormat</code>	否	要產生的請求大量資料檔案格式。接受的值為 <code>application/fhir+ndjson</code> 、 <code>application/ndjson</code> 、 <code>ndjson</code> 。	
<code>_type</code>	否	您想要包含在匯出任務中的逗號分隔 FHIR 資源類型的字串。我們建議包含 <code>_type</code> 因為匯出所有資源時，這可能會牽涉成本。	<code>&_type=MedicationStatement,Observation</code>
<code>_since</code>	否	在日期時間戳記當天或之後修改的資源類型。如果資源類型沒有上次更新的時間，它們將包含在回應中。	<code>&_since=2024-05-09T00%3A00%3A00Z</code>
<code>_until</code>	否	在日期時間戳記當天或之前修改的資源類型。與 <code>_since</code> 結合使用，以定義匯出的特定時間範圍。如果資源類型沒有上次更新的時間，則會將其排除在您的回應中。	<code>&_until=2024-12-31T23%3A59%3A59Z</code>
<code>_security</code>	否	依 <code>meta.security</code> 編碼值篩選	<code>&_security=https://</code>

名稱	是否為必要？	Description	範例
		匯出的資源。使用 <code>system code</code> 格式。提供多個值時，資源必須符合所有值 (AND 語意)。使用 <code>system </code> (追蹤管道、無程式碼) 來比對來自指定系統的任何程式碼。	<code>/myorg.com/tenant%7Cclinic-A</code>
<code>_tag</code>	否	依 <code>meta.tag</code> 編碼值篩選匯出的資源。使用與相同的 <code>system code</code> 格式和 AND 語意 <code>_security</code> 。同時指定 <code>_tag</code> <code>_security</code> 和時，資源必須符合兩個篩選條件。	<code>&_tag=https://myorg.com/dept%7Ccardiology</code>

步驟 2：指定必要的標頭參數

若要使用 FHIR REST API 提出匯出請求，您必須指定下列標頭參數。

- `Content-Type`： `application/fhir+json`
- 偏好： `respond-async`

接著，您必須在請求內文中指定必要的元素。

步驟 3：指定 定義必要參數的請求內文。

匯出請求也需要 JSON 格式的內文。內文可以包含下列參數。

金鑰	是否為必要？	Description	Value
DataAccessRoleArn	是	HealthLake 服務角色的 ARN。使用的服務角色必須指定 HealthLake 做為服務主體。	arn:aws:iam:: 444455556666 :role/ your-healthlake-service-role
JobName	否	匯出請求的名稱。	your-export-job-name
S3Uri	是	OutputDataConfig 金鑰的一部分。要下載匯出資料之目的地儲存貯體的 S3 URI。	s3://amzn-s3-demo-bucket/ EXPORT-JOB /
KmsKeyId	是	OutputDataConfig 金鑰的一部分。用於保護 Amazon S3 儲存貯體之 AWS KMS 金鑰的 ARN。	arn:aws:kms: region-of-bucket:123456789012 :key/ 1234abcd-12ab-34cd-56ef-1234567890ab

Example 使用 FHIR REST API 提出的匯出請求內文

若要使用 FHIR REST API 提出匯出請求，您必須指定內文，如下所示。

```
{
  "DataAccessRoleArn": "arn:aws:iam::444455556666:role/your-healthlake-service-role",
  "JobName": "your-export-job",
  "OutputDataConfig": {
    "S3Configuration": {
      "S3Uri": "s3://amzn-s3-demo-bucket/EXPORT-JOB",
      "KmsKeyId": "arn:aws:kms:region-of-bucket:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  }
}
```

```
}
```

當您的請求成功時，您將會收到下列回應。

回應標頭

```
content-location: https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/  
export/your-export-request-job-id
```

回應內文

```
{  
  "datastoreId": "your-data-store-id",  
  "jobStatus": "SUBMITTED",  
  "jobId": "your-export-request-job-id"  
}
```

管理您的匯出請求

成功提出匯出請求後，您可以使用 `管理請求$export`，以描述目前匯出請求的狀態，以及\$export取消目前的匯出請求。

當您使用 REST API 取消匯出請求時，您只需支付截至您提交取消請求之前匯出的部分資料的費用。

下列主題說明如何取得狀態或取消目前的匯出請求。

取消匯出請求

若要取消匯出請求，請提出DELETE請求並在請求 URL 中提供任務 ID。

```
DELETE https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/export/your-  
export-request-job-id
```

當您的請求成功時，您會收到以下內容。

```
{  
  "exportJobProperties": {  
    "jobId": "your-original-export-request-job-id",  
    "jobStatus": "CANCEL_SUBMITTED",
```

```
"datastoreId": "your-data-store-id"
}
```

當您的請求不成功時，您會收到以下內容。

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "not-supported",
      "diagnostics": "Interaction not supported."
    }
  ]
}
```

描述匯出請求

若要取得匯出請求的狀態，請使用 `export` 和 提出 GET 請求 `export-request-job-id`。

```
GET https://healthlake.region.amazonaws.com/datastore/datastoreId/r4/export/your-export-request-id
```

JSON 回應將包含 `ExportJobProperties` 物件。它可能包含下列索引鍵：值對。

名稱	是否為必要？	Description	Value
<code>DataAccessRoleArn</code>	否	HealthLake 服務角色的 ARN。使用的服務角色必須指定 HealthLake 做為服務主體。	<code>arn:aws:iam:: 444455556666 :role/<i>your-healthlake-service-role</i></code>
<code>SubmitTime</code>	否	匯出任務提交的日期。	<code>Apr 21, 2023 5:58:02</code>
<code>EndTime</code>	否	匯出任務完成的時間。	<code>Apr 21, 2023 6:00:08 PM</code>

名稱	是否為必要？	Description	Value
JobName	否	匯出請求的名稱。	your-export-job-name
JobStatus	否		有效的值如下： <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; width: fit-content; margin: 10px auto;"> SUBMITTED IN_PROGRESS COMPLETED _WITH_ERRORS COMPLETED FAILED </div>
S3Uri	是	OutputDataConfig 物件的一部分。要下載匯出資料之目的地儲存貯體的 Amazon S3 URI。	s3://amzn-s3-demo-bucket/ EXPORT-JOB /
KmsKeyId	是	OutputDataConfig 物件的一部分。用於保護 Amazon S3 儲存貯體之 AWS KMS 金鑰的 ARN。	arn:aws:kms: region-of-bucket:123456789012 :key/1234abcd-12ab-34cd-56ef-1234567890ab

Example：描述使用 FHIR REST API 提出之匯出請求的內文

成功時，您會收到下列 JSON 回應。

```
{
  "exportJobProperties": {
    "jobId": "your-export-request-id",
    "jobName": "your-export-job",
    "jobStatus": "SUBMITTED",
    "submitTime": "Apr 21, 2023 5:58:02 PM",
    "endTime": "Apr 21, 2023 6:00:08 PM",
  }
}
```

```

    "datastoreId": "your-data-store-id",
    "outputDataConfig": {
      "s3Configuration": {
        "S3Uri": "s3://amzn-s3-demo-bucket/EXPORT-JOB",
        "KmsKeyId": "arn:aws:kms:region-of-
bucket:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    },
    "DataAccessRoleArn": "arn:aws:iam::444455556666:role/your-healthlake-service-role",
  }
}

```

HealthLake 的 FHIR \$inquire 操作

\$inquire 操作可讓您檢查先前提交的預先授權請求的狀態。此操作實作 [Da Vinci 預先授權支援 \(PAS\) 實作指南](#)，提供標準化的 FHIR 工作流程來擷取目前的授權決策。

運作方式

- 提交查詢：您傳送 FHIR 套件，其中包含您要檢查的宣告和支援資訊
- 搜尋：HealthLake 會在您的資料存放區中搜尋對應的 ClaimResponse
- 擷取：擷取最新的授權狀態
- 回應：您會收到目前授權狀態的立即回應（佇列、核准、拒絕等）

Note

\$inquire 是擷取現有授權狀態的唯讀操作。它不會修改或更新資料存放區中的任何資源。

API 端點

```

POST /datastore/{datastoreId}/r4/Claim/$inquire
Content-Type: application/fhir+json

```

請求結構

套件需求

您的請求必須是具有下列項目的 FHIR 套件資源：

- `Bundle.type` : 必須為 "collection"
- `Bundle.entry` : 必須包含一個具有下列項目的宣告資源 :
 - `use = "preauthorization"`
 - `status = "active"`
- 參考資源 : 宣告參考的所有資源必須包含在套件中

Query-by-Example

輸入套件中的資源做為搜尋範本。HealthLake 會使用提供的資訊來尋找對應的 ClaimResponse。

必要的資源

資源	基數	設定檔	Description
宣告	1	PAS 宣告查詢	您正在查詢的預先授權
病患	1	PAS 受益人病患	病患人口統計資訊
Organization (保險業者)	1	PAS 保險公司組織	保險公司
Organization (提供者)	1	PAS 請求組織	提交請求的醫療保健供應商

重要搜尋條件

HealthLake 使用下列項目搜尋 ClaimResponse :

- 來自宣告的患者參考
- 來自宣告的保險業者參考
- 來自宣告的提供者參考
- 從宣告建立的日期 (做為時間篩選條件)

i 僅限患者特定的查詢

所有查詢都必須繫結至特定病患。不允許沒有患者識別的全系統查詢。

範例請求

```
POST /datastore/example-datastore/r4/Claim/$inquire
Content-Type: application/fhir+json
Authorization: Bearer <your-token>

{
  "resourceType": "Bundle",
  "id": "PASClaimInquiryBundleExample",
  "meta": {
    "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-pas-inquiry-request-bundle"]
  },
  "identifier": {
    "system": "http://example.org/SUBMITTER_TRANSACTION_IDENTIFIER",
    "value": "5269368"
  },
  "type": "collection",
  "timestamp": "2005-05-02T14:30:00+05:00",
  "entry": [
    {
      "fullUrl": "http://example.org/fhir/Claim/MedicalServicesAuthorizationExample",
      "resource": {
        "resourceType": "Claim",
        "id": "MedicalServicesAuthorizationExample",
        "meta": {
          "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claim-inquiry"]
        },
        "status": "active",
        "type": {
          "coding": [{
            "system": "http://terminology.hl7.org/CodeSystem/claim-type",
            "code": "professional"
          }]
        },
        "use": "preauthorization",
        "patient": {
```

```
    "reference": "Patient/SubscriberExample"
  },
  "created": "2005-05-02T11:01:00+05:00",
  "insurer": {
    "reference": "Organization/InsurerExample"
  },
  "provider": {
    "reference": "Organization/UMOExample"
  }
}
},
{
  "fullUrl": "http://example.org/fhir/Patient/SubscriberExample",
  "resource": {
    "resourceType": "Patient",
    "id": "SubscriberExample",
    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-beneficiary"]
    },
    "name": [{
      "family": "SMITH",
      "given": ["JOE"]
    }],
    "gender": "male"
  }
},
{
  "fullUrl": "http://example.org/fhir/Organization/UMOExample",
  "resource": {
    "resourceType": "Organization",
    "id": "UMOExample",
    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-requestor"]
    },
    "name": "Provider Organization"
  }
},
{
  "fullUrl": "http://example.org/fhir/Organization/InsurerExample",
  "resource": {
    "resourceType": "Organization",
    "id": "InsurerExample",
```

```

    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
insurer"]
    },
    "name": "Insurance Company"
  }
}
]
}

```

回應格式

成功回應 (200 OK)

您將會收到 PAS 查詢回應套件，其中包含：

- 具有目前授權狀態的 ClaimResponse；如果符合搜尋條件，則為多個 ClaimResponse
- 請求中的所有原始資源（已回呼）
- 組合回應時的時間戳記

可能的 ClaimResponse 結果

結果	Description
queued	授權請求仍在等待審核
complete	已做出授權決策（檢查disposition 是否已核准/遭拒）
error	處理期間發生錯誤
partial	已授予部分授權

```

{
  "resourceType": "Bundle",
  "identifier": {
    "system": "http://example.org/SUBMITTER_TRANSACTION_IDENTIFIER",
    "value": "5269367"
  },
  "type": "collection",

```

```
"timestamp": "2005-05-02T14:30:15+05:00",
"entry": [
  {
    "fullUrl": "http://example.org/fhir/ClaimResponse/InquiryResponseExample",
    "resource": {
      "resourceType": "ClaimResponse",
      "id": "InquiryResponseExample",
      "meta": {
        "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claimresponse-inquiry"]
      },
      "status": "active",
      "type": {
        "coding": [{
          "system": "http://terminology.hl7.org/CodeSystem/claim-type",
          "code": "professional"
        }]
      },
      "use": "preauthorization",
      "patient": {
        "reference": "Patient/SubscriberExample"
      },
      "created": "2005-05-02T11:05:00+05:00",
      "insurer": {
        "reference": "Organization/InsurerExample"
      },
      "request": {
        "reference": "Claim/MedicalServicesAuthorizationExample"
      },
      "outcome": "complete",
      "disposition": "Approved",
      "preAuthRef": "AUTH12345"
    }
  },
  {
    "fullUrl": "http://example.org/fhir/Claim/MedicalServicesAuthorizationExample",
    "resource": {
      "resourceType": "Claim",
      "id": "MedicalServicesAuthorizationExample",
      "meta": {
        "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claim-inquiry"]
      },
      "status": "active",
```

```

    "type": {
      "coding": [{
        "system": "http://terminology.hl7.org/CodeSystem/claim-type",
        "code": "professional"
      }]
    },
    "use": "preauthorization",
    "patient": {
      "reference": "Patient/SubscriberExample"
    },
    "created": "2005-05-02T11:01:00+05:00",
    "insurer": {
      "reference": "Organization/InsurerExample"
    },
    "provider": {
      "reference": "Organization/UMOExample"
    }
  }
},
{
  "fullUrl": "http://example.org/fhir/Patient/SubscriberExample",
  "resource": {
    "resourceType": "Patient",
    "id": "SubscriberExample",
    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-beneficiary"]
    },
    "name": [{
      "family": "SMITH",
      "given": ["JOE"]
    }],
    "gender": "male"
  }
},
{
  "fullUrl": "http://example.org/fhir/Organization/UMOExample",
  "resource": {
    "resourceType": "Organization",
    "id": "UMOExample",
    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-requestor"]
    }
  },

```

```

    "name": "Provider Organization"
  }
},
{
  "fullUrl": "http://example.org/fhir/Organization/InsurerExample",
  "resource": {
    "resourceType": "Organization",
    "id": "InsurerExample",
    "meta": {
      "profile": ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
insurer"]
    },
    "name": "Insurance Company"
  }
}
]
}

```

錯誤回應

400 錯誤的請求

當請求格式無效或驗證失敗時傳回。

```

{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "required",
      "diagnostics": "Reference 'Patient/SubscriberExample' at path 'patient' for
'CLAIM' resource not found(at Bundle.entry[0].resource)"
    }
  ]
}

```

401 (未經授權)

當身分驗證憑證遺失或無效時傳回。

```

{
  "resourceType": "OperationOutcome",

```

```

    "issue": [
      {
        "severity": "error",
        "code": "forbidden",
        "diagnostics": "Invalid authorization header"
      }
    ]
  }

```

403 禁止

當已驗證的使用者缺少存取所請求資源的許可時傳回。

```

{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "exception",
      "diagnostics": "Insufficient SMART scope permissions."
    }
  ]
}

```

400 當找不到時

查詢找不到相符的 ClaimResponse 時傳回。

```

{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "not-found",
    "diagnostics": "Resource not found. No ClaimResponse found from the input Claim
that matches the specified Claim properties patient, insurer, provider, and created(at
Bundle.entry[0].resource)"
  }]
}

```

415 不支援的媒體類型

當 Content-Type 標頭不是 application/fhir+json 時傳回。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "value",
    "diagnostics": "Incorrect MIME-type. Update request Content-Type header."
  }]
}
```

429 太多請求

超過速率限制時傳回。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "throttled",
    "diagnostics": "Rate limit exceeded. Please retry after some time."
  }]
}
```

驗證規則

HealthLake 會對您的查詢執行全面的驗證：

套件驗證

- 必須符合 PAS 查詢請求套件描述檔
- `Bundle.type` 必須 "collection"
- 必須僅包含一個宣告資源
- 套件中必須包含所有參考的資源

宣告驗證

- 必須符合 PAS 宣告查詢設定檔
- `Claim.use` 必須 "preauthorization"
- `Claim.status` 必須 "active"
- 必要欄位：patient、insurer、provider、created

資源驗證

- 所有資源必須符合其各自的 PAS 查詢設定檔
- 必要的支援資源必須存在（病患、保險公司組織、提供者組織）
- 交互參考在套件中必須是有效且可解析的

效能規格

指標	規格
資源計數限制	每個套件 500 個資源
套件大小限制	最多 5 MB

所需的許可

若要使用 `$inquire` 操作，請確定您的 IAM 角色具有：

- `healthlake:InquirePreAuthClaim` - 呼叫 操作

FHIR 範圍上的 SMART

所需範圍下限：

- SMART v1：`user/ClaimResponse.read`
- SMART v2：`user/ClaimResponse.s`

重要實作備註

搜尋行為

當您提交查詢時，HealthLake 會使用下列方式搜尋 ClaimResponse：

- 來自輸入宣告的患者參考
- 來自輸入 Claim 的保險業者參考
- 來自輸入宣告的提供者參考

- 從輸入宣告建立日期（做為時間篩選條件）

多個相符項目：如果多個 ClaimResponses 符合您的搜尋條件，則 HealthLake 會傳回所有相符結果。您應該使用最新的 ClaimResponse.created 時間戳記來識別最新狀態。

已更新宣告

如果您已提交相同預先授權的多個更新（例如 Claim v1.1、v1.2、v1.3），\$inquire 操作會根據提供的搜尋條件擷取與最新版本相關聯的 ClaimResponse。

唯讀操作

\$inquire 操作：

- 會擷取現有的授權狀態
- 會傳回最新的 ClaimResponse
- 不修改或更新任何資源
- 不建立新資源
- 不會觸發新的授權處理

工作流程範例

典型的預先授權查詢工作流程

1. Provider submits PA request
POST /Claim/\$submit
Returns ClaimResponse with outcome="queued"
2. Payer reviews request (asynchronous)
Updates ClaimResponse status internally
3. Provider checks status
POST /Claim/\$inquire
Returns ClaimResponse with outcome="queued" (still pending)
4. Provider checks status again later
POST /Claim/\$inquire
Returns ClaimResponse with outcome="complete", disposition="Approved"

相關的 操作

- Claim/\$submit - 提交新的預先授權請求或更新現有的預先授權請求
- Patient/\$everything - 針對預先授權內容擷取完整的患者資料

使用 擷取概念詳細資訊 \$lookup

AWS HealthLake 現在支援 CodeSystem 資源的 \$lookup 操作，可讓您透過提供程式碼等識別資訊，擷取程式碼系統中特定概念的詳細資訊。當您需要：

- 擷取特定醫療代碼的詳細資訊
- 驗證程式碼意義和屬性
- 存取概念定義和關係
- 使用準確的術語資料支援臨床決策

Usage

您可以使用 GET 和 POST 方法在 CodeSystem 資源上叫用 \$lookup 操作：

受支援的 操作

```
GET [base]/CodeSystem/$lookup?system=http://snomed.info/sct&code=73211009&version=20230901
POST [base]/CodeSystem/$lookup
```

支援的參數

HealthLake 支援 FHIR R4 \$lookup 參數的子集：

參數	Type	必要	Description
code	code	是	您要查詢的概念代碼（例如 SNOMED CT 中的 "71620000"）
system	uri	是	程式碼系統的正式 URL（例如 " http://snomed.info/sct "）
version	string	否	特定版本的程式碼系統

範例

GET 請求

```
GET [base]/CodeSystem/$lookup?system=http://snomed.info/sct&code=71620000&version=2023-09
```

POST 請求

```
POST [base]/CodeSystem/$lookup
Content-Type: application/fhir+json

{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "system",
      "valueUri": "http://snomed.info/sct"
    },
    {
      "name": "code",
      "valueCode": "71620000"
    },
    {
      "name": "version",
      "valueString": "2023-09"
    }
  ]
}
```

回應範例

操作會傳回參數資源，其中包含概念詳細資訊：

```
{
  "resourceType": "Parameters",
  "parameter": [{
    "name": "name",
    "valueString": "SNOMED CT Fractures"
  },
  {
    "name": "version",
```

```
    "valueString": "2023-09"
  },
  {
    "name": "display",
    "valueString": "Fracture of femur"
  },
  {
    "name": "property",
    "part": [{
      "name": "code",
      "valueCode": "child"
    },
    {
      "name": "value",
      "valueCode": "263225007"
    },
    {
      "name": "description",
      "valueString": "Fracture of neck of femur"
    }
  ]
},
{
  "name": "property",
  "part": [{
    "name": "code",
    "valueCode": "child"
  },
  {
    "name": "value",
    "valueCode": "263227004"
  },
  {
    "name": "description",
    "valueString": "Fracture of shaft of femur"
  }
]
}
]
```

回應參數

回應會在可用時包含下列參數：

參數	類型	Description
name	string	程式碼系統的名稱
version	string	程式碼系統的版本
display	string	顯示概念的名稱
designation	BackboneElement	此概念的其他表示法。
property	BackboneElement	概念的其他屬性（定義、關係等）

Behavior (行為)

\$lookup 操作：

1. 驗證所需的參數 (code 和 system)
2. 在存放在資料存放區的指定程式碼系統中搜尋概念
3. 傳回詳細的概念資訊，包括顯示名稱、指定項目和屬性。
4. 提供 version 參數時支援版本特定的查詢
5. 僅在明確存放在 HealthLake 資料存放區的程式碼系統上操作

錯誤處理

操作會處理下列錯誤條件：

- 400 錯誤的請求：無效的 \$lookup 操作（不符合的請求或缺少必要的參數）
- 找不到 404：找不到程式碼系統，或在指定的程式碼系統中找不到程式碼

警告

在此版本中，不支援下列項目：

- \$lookup 透過呼叫外部術語伺服器進行操作
- \$lookup CodeSystems 上的操作由 HealthLake 管理，但未明確存放在資料存放區中

如需 \$lookup 操作規格的詳細資訊，請參閱 [FHIR R4 CodeSystem \\$lookup](#) 文件。

\$member-add HealthLake 的操作

FHIR \$member-add 操作會將成員（病患）新增至群組資源，特別是成員屬性清單。此操作是 DaVinci 成員屬性實作指南的一部分，並支援管理成員屬性的調校程序。

操作端點

```
POST [base]/datastore/{datastoreId}/r4/Group/{groupId}/$member-add
Content-Type: application/json
```

Parameters

操作接受具有下列參數組合的 FHIR 參數資源：

參數選項

您可以使用下列其中一個參數組合：

選項 1：成員 ID + 供應商 NPI

```
memberId + providerNpi
```

```
memberId + providerNpi + attributionPeriod
```

選項 2：病患參考 + 提供者參考

```
patientReference + providerReference
```

```
patientReference + providerReference + attributionPeriod
```

參數詳細資訊

memberId（選用）

要新增至群組之成員的識別符。

類型：Identifier

系統：病患識別符系統

```
{
  "name": "memberId",
  "valueIdentifier": {
    "system": "http://example.org/patient-id",
    "value": "patient-new"
  }
}
```

providerNpi (選用)

屬性提供者的國家提供者識別符 (NPI)。

類型：Identifier

系統：<https://http://terminology.hl7.org/CodeSystem/NPI>

```
{
  "name": "providerNpi",
  "valueIdentifier": {
    "system": "http://terminology.hl7.org/CodeSystem/NPI",
    "value": "1234567890"
  }
}
```

patientReference (選用)

要新增之病患資源的直接參考。

類型：參考

```
{
  "name": "patientReference",
  "valueReference": {
    "reference": "Patient/patient-123"
  }
}
```

providerReference (選用)

直接參考提供者資源。

類型：參考

```
{
  "name": "providerReference",
  "valueReference": {
    "reference": "Practitioner/provider-456"
  }
}
```

attributionPeriod (選用)

病患歸因於提供者的期間。

類型：期間

```
{
  "name": "attributionPeriod",
  "valuePeriod": {
    "start": "2024-07-15",
    "end": "2025-07-14"
  }
}
```

請求範例

使用成員 ID 和提供者 NPI

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "memberId",
      "valueIdentifier": {
        "system": "http://example.org/patient-id",
        "value": "patient-new"
      }
    },
    {
      "name": "providerNpi",
      "valueIdentifier": {
        "system": "http://terminology.hl7.org/CodeSystem/NPI",
        "value": "1234567890"
      }
    }
  ],
}
```

```
{
  "name": "attributionPeriod",
  "valuePeriod": {
    "start": "2024-07-15",
    "end": "2025-07-14"
  }
}
```

使用病患和提供者參考

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "patientReference",
      "valueReference": {
        "reference": "Patient/patient-123"
      }
    },
    {
      "name": "providerReference",
      "valueReference": {
        "reference": "Practitioner/provider-456"
      }
    },
    {
      "name": "attributionPeriod",
      "valuePeriod": {
        "start": "2024-07-15",
        "end": "2025-07-14"
      }
    }
  ]
}
```

回應格式

成功的新增回應

```
HTTP Status: 200 OK
Content-Type: application/fhir+json
```

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "success",
      "code": "informational",
      "details": {
        "text": "Member Patient/patient-new successfully added to the Member Attribution List."
      }
    }
  ]
}
```

錯誤回應

無效的請求語法

HTTP 狀態：400 錯誤的請求

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "invalid",
      "details": {
        "text": "Invalid parameter combination provided"
      }
    }
  ]
}
```

找不到資源

HTTP 狀態：找不到 404

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
```

```
    "code": "not-found",
    "details": {
      "text": "Resource not found."
    }
  ]
}
```

版本衝突

HTTP 狀態：409 衝突

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "conflict",
      "details": {
        "text": "Resource version conflict detected"
      }
    }
  ]
}
```

無效的屬性狀態

HTTP 狀態：422 無法處理的實體

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "business-rule",
      "details": {
        "text": "Cannot add member to Attribution List with status 'final'. Status
must be 'draft' or 'open'."
      }
    }
  ]
}
```

業務規則

屬性狀態驗證

只有在群組的屬性狀態為：

- draft
- open

當狀態為 `final` 時，不允許操作 `final`。

防止重複成員

系統會根據以下項目的唯一組合，防止新增重複的成員：

- 成員識別符
- 付款人識別符
- 涵蓋範圍識別符

涵蓋期間驗證

提供 `attributionPeriod` 時，必須落在成員涵蓋期間的界限內。系統會：

- 搜尋成員的涵蓋範圍資源
- 如果存在多個，請使用最新的涵蓋範圍（最高 `versionId`）
- 驗證歸因期間未超過涵蓋期間

參驗證

當同時為相同的資源（病患或提供者）提供 ID 和參考時，系統會驗證它們是否對應至相同的資源。

當同時為相同的資源（病患或提供者）提供 ID 和 `reference.identifier` 欄位時，會擲回錯誤。

身分驗證與授權

此操作需要對下列項目進行 FHIR 上的 SMART 授權：

- 讀取許可 - 驗證患者、提供者和群組資源
- 搜尋許可 - 依識別符尋找資源
- 更新許可 - 修改群組資源

操作行為

資源更新

- 更新群組資源版本 ID
- 在操作之前建立具有原始資源狀態的歷史記錄項目
- 使用 新增成員資訊至 Group.member 陣列：
 - entity.reference 中的病患參考
 - 期間的屬性期間
 - 延伸欄位中的涵蓋範圍和提供者資訊

驗證步驟

- 參數驗證 - 確保有效的參數組合
- 資源存在 - 驗證存在的患者、提供者和群組資源
- 屬性狀態 - 確認群組狀態允許修改
- 重複檢查 - 防止新增現有成員
- 涵蓋範圍驗證 - 確保歸因期間在涵蓋範圍內

限制

- 所有參考的資源都必須存在於相同的資料存放區中
- 操作僅適用於成員屬性清單群組資源
- 屬性期間必須在涵蓋範圍內
- 無法修改具有「最終」狀態的群組

\$member-match HealthLake 的操作

AWS HealthLake 現在支援病患資源 `$member-match` 的操作，讓醫療保健組織能夠使用人口統計和涵蓋範圍資訊，在不同醫療保健系統中尋找成員的唯一識別符。此操作對於實現 CMS 合規和促進安全 payer-to-payer 資料交換，同時維護患者隱私至關重要。

當您需要：

- 啟用組織之間的安全醫療保健資料交換
- 維持不同系統的患者持續照護
- 支援 CMS 合規要求

- 促進跨醫療保健網路的準確成員識別

Usage

您可以使用 POST 方法，在病患資源上叫用 \$member-match 操作：

```
POST [base]/Patient/$member-match
```

支援的參數

HealthLake 支援下列 FHIR \$member-match 參數：

參數	Type	必要	預設	Description
MemberPatient	病患	是	—	包含要比對之成員人口統計資訊的患者資源
CoverageTypeMatch	涵蓋範圍	是	—	將用於比對現有記錄的涵蓋資源
CoverageTypeLink	涵蓋範圍	否	—	比對程序期間要連結的覆蓋資源
同意	同意	否	—	授權用途的同意資源

範例

具有參數的 POST 請求

```
POST [base]/Patient/$member-match
Content-Type: application/fhir+json

{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "MemberPatient",
      "resource": {
        "resourceType": "Patient",
        "name": [
```

```
    {
      "family": "Jones",
      "given": ["Sarah"]
    }
  ],
  "gender": "female",
  "birthDate": "1985-05-15"
}
},
{
  "name": "CoverageToMatch",
  "resource": {
    "resourceType": "Coverage",
    "status": "active",
    "beneficiary": {
      "reference": "Patient/1"
    },
    "relationship": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/subscriber-
relationship",
          "code": "self",
          "display": "Self"
        }
      ]
    },
    "payor": [
      {
        "reference": "Organization/payer456"
      }
    ]
  }
},
{
  "name": "Consent",
  "resource": {
    "resourceType": "Consent",
    "status": "active",
    "scope": {
      "coding": [
        {
          "system": "http://terminology.hl7.org/CodeSystem/consentscope",
          "code": "patient-privacy"
```

```

    }
  ]
},
"category": [
  {
    "coding": [
      {
        "system": "http://terminology.hl7.org/CodeSystem/v3-ActCode",
        "code": "IDSCL"
      }
    ]
  }
],
"patient": {
  "reference": "Patient/1"
},
"performer": [
  {
    "reference": "Patient/patient123"
  }
],
"sourceReference": {
  "reference": "Document/someconsent"
},
"policy": [
  {
    "uri": "http://hl7.org/fhir/us/davinci-hrex/StructureDefinition-hrex-consent.html#regular"
  }
]
}
]
}
}

```

回應範例

操作會傳回參數資源，其中包含相符的結果：

```

{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "MemberIdentifier",

```

```

    "valueIdentifier": {
      "system": "http://hospital.org/medical-record-number",
      "value": "MRN-123456"
    }
  },
  {
    "name": "MemberId",
    "valueReference": {
      "reference": "Patient/patient123"
    }
  },
  {
    "name": "matchAlgorithm",
    "valueString": "DEMOGRAPHIC_MATCH"
  },
  {
    "name": "matchDetails",
    "valueString": "Demographic match: DOB + Name"
  },
  {
    "name": "matchedFields",
    "valueString": "given,birthdate,gender,family"
  }
]
}

```

回應參數

找到相符項目時，回應會包含下列參數：

參數	類型	說明
MemberIdentifier	識別符	相符成員的唯一識別符
MemberId	參考資料	患者資源的參考
matchAlgorithm	String	使用的比對演算法類型 (EXACT_MATCH、STRONG_MATCH 或 DEMOGRAPHIC_MATCH)
matchDetails	String	比對程序的詳細資訊

參數	類型	說明
matchedFields	String	已成功相符的特定欄位清單

相符演算法

\$member-match API 採用多層比對方法，以確保準確的成員識別：

EXACT_MATCH

使用與 Coverage SubscriberId 結合的患者識別符

為成員比對提供最高的可信度層級

STRONG_MATCH

使用具有最低涵蓋範圍資訊的患者識別符

不符合完全相符條件時，提供高可信度

DEMOGRAPHIC_MATCH

依賴基本人口統計資訊

當無法進行識別符型比對時使用

Behavior (行為)

\$member-match 操作：

- 接受患者人口統計特性、涵蓋範圍詳細資訊和選用的同意資訊做為輸入
- 傳回可用於後續互動的唯一成員識別符
- 實作多層比對（確切、強大、人口統計），以確保在不同醫療保健系統中正確識別成員
- 儲存任何提供的同意資訊以供未來授權之用
- 支援安全payer-to-payer資料交換，同時維護患者隱私權
- 符合醫療資料交換的 CMS 要求

Authorization

API 在 FHIR 授權通訊協定上使用 SMART 搭配下列必要範圍：

- system/Patient.read
- system/Coverage.read
- system/Organization.read (有條件)
- system/Practitioner.read (有條件)
- system/PractitionerRole.read (有條件)
- system/Consent.write (有條件)

錯誤處理

操作會處理下列錯誤條件：

- 400 Bad Request：無效的\$member-match操作（不符合的請求或缺少必要的參數）
- 422 Unprocessable Entity：找不到相符項目或多個相符項目

\$member-remove HealthLake 的操作

\$member-remove 操作可讓您從 FHIR 成員屬性清單（群組資源）中移除成員 AWS HealthLake。此操作是 DaVinci 成員屬性實作指南的一部分，並支援管理成員屬性的調校程序。

先決條件

- AWS HealthLake FHIR 資料存放區
- HealthLake 操作的適當 IAM 許可
- 草稿或開啟狀態的成員屬性清單（群組資源）

操作詳細資訊

Endpoint

```
POST /Group/{id}/$member-remove
```

內容類型

```
application/fhir+json
```

Parameters

操作接受具有下列選用參數的 FHIR 參數資源：

參數	基數	Type	說明
memberId	0..1	識別符	要移除之成員的業務識別符
providerNpi	0..1	識別符	屬性提供者的 NPI
patientReference	0..1	參考資料	直接參考 病患資源
providerReference	0..1	參考資料	直接參考提供者資源 (Practitioner、PractitionerRole 或 Organization)
coverageReference	0..1	參考資料	涵蓋範圍資源的參考

支援的參數組合

支援下列參數組合：

- memberId 僅限 - 移除指定成員的所有屬性
- memberId + providerNpi - 移除特定成員提供者組合的屬性
- patientReference 僅限 - 移除指定病患的所有屬性
- patientReference + providerReference - 移除特定患者提供者組合的屬性
- patientReference + providerReference + coverageReference - 根據病患、提供者和涵蓋範圍移除特定歸因

請求範例

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "patientReference",
      "valueReference": {
        "reference": "Patient/12345"
      }
    }
  ],
  {
```

```
    "name": "providerReference",
    "valueReference": {
      "reference": "Practitioner/67890"
    }
  }
]
```

回應

成功回應

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "result",
      "valueBoolean": true
    },
    {
      "name": "effectiveDate",
      "valueDate": "2024-06-30"
    },
    {
      "name": "status",
      "valueCode": "inactive"
    },
    {
      "name": "message",
      "valueString": "Member successfully removed from attribution list"
    }
  ]
}
```

Behavior (行為)

狀態需求

操作僅適用於狀態為 `draft` 或 的屬性清單 `open`

`final` 狀態為 的清單將拒絕 422 錯誤的 操作

成員移除程序

草稿狀態清單：成員標示為非作用中 (`inactive: true`)，且其 `changeType` 延伸項目已更新為 `changed`

開啟狀態清單：類似草稿狀態的行為

最終狀態清單：操作遭拒

驗證

驗證參考以確保它們存在於 HealthLake 資料存放區中

如果同時為相同的資源類型提供識別符和參考，它們必須對應至相同的資源

根據支援的模式驗證參數組合

錯誤處理

常見錯誤回應

找不到資源 (404)

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "not-found",
      "details": {
        "text": "Patient with identifier 'http://example.org/fhir/identifiers|99999'
not found in system"
      },
      "diagnostics": "Cannot remove member from attribution list. Verify patient
identifier and try again.",
      "expression": ["memberId"]
    }
  ]
}
```

屬性清單最終狀態 (422)

```
{
  "resourceType": "OperationOutcome",
```

```

    "issue": [
      {
        "severity": "error",
        "code": "business-rule",
        "details": {
          "coding": [
            {
              "system": "http://hl7.org/fhir/us/davinci-atr/CodeSystem/atr-error-
codes",
              "code": "list-final",
              "display": "Attribution list is final and cannot be modified"
            }
          ]
        },
        "diagnostics": "Cannot modify attribution list with status 'final'. List
modifications are not permitted after finalization.",
        "expression": ["Group.status"]
      }
    ]
  }
}

```

無效操作 (400)

當參數組合無效或格式不正確時傳回。

找到多個相符項目 (412)

當提供的參數符合屬性清單中的多個成員時傳回。

```

{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "error",
      "code": "processing",
      "diagnostics": "Multiple members found matching the criteria"
    }
  ]
}

```

最佳實務

- 使用特定參數：盡可能使用最特定的參數組合，以避免意外移除

- 檢查清單狀態：嘗試移除之前，請先驗證屬性清單狀態
- 正常處理錯誤：針對所有可能的錯誤條件實作適當的錯誤處理
- 驗證參考：確定所有參考的資源都存在，然後再提出請求

使用 移除病患室資源 \$purge

AWS HealthLake 支援 \$purge 操作，可永久刪除病患隔室內的所有資源。當您需要執行下列動作時，此操作特別有用：

- 移除與病患相關聯的所有資料
- 遵循病患資料移除請求
- 管理病患資料生命週期
- 執行全面的病患記錄清除

Usage

您可以在病患資源上叫用 \$purge 操作：

```
POST [base]/Patient/[ID]/$purge?deleteAuditEvent=true
```

Parameters

參數	Type	必要	預設	Description
deleteAuditEvent	布林值	否	false	為 true 時，會刪除相關聯的稽核事件
_since	string	否	資料存放區 建立時間	輸入時，會根據其lastModified的時間選取開始截止時間以尋找資源。無法與開始或結束搭配使用
start	string	否	資料存放區 建立時間	輸入時，會根據資源的lastModified時間選取截止時間來尋找資源。可與 end 搭配使用
end	string	否	任務提交時間	輸入時，會根據其lastModified的時間選取結束截止時間以尋找資源

範例

範例請求

```
POST [base]/Patient/example-patient/$purge?deleteAuditEvent=true
```

回應範例

```
{
  "resourceType": "OperationOutcome",
  "id": "purge-job",
  "issue": [
    {
      "severity": "information",
      "code": "informational",
      "diagnostics": "Purge job started successfully. Job ID:
12345678-1234-1234-1234-123456789012"
    }
  ]
}
```

任務狀態

若要檢查清除任務的狀態：

```
GET [base]/$purge/[jobId]
```

操作會傳回任務狀態資訊：

```
{
  "datastoreId": "36622996b1fceb7e12ee2ee085308d3",
  "jobId": "3dd1c7a5b6c0ef8c110f566eb87e2ef9",
  "status": "COMPLETED",
  "submittedTime": "2025-10-31T18:43:21.822Z"
}
```

Behavior (行為)

\$purge 操作：

1. 以非同步方式處理多個資源

2. 維護 ACID 交易的資料完整性
3. 提供具有資源刪除計數的任務狀態追蹤
4. 永久移除病患隔室中的所有資源
5. 包括刪除活動的完整稽核記錄
6. 支援選擇性刪除稽核事件

稽核記錄

\$purge 操作會記錄為 StartFHIRBulkDeleteJob 和 DescribeFHIRBulkDeleteJob，其中包含詳細的操作資訊。

限制

- 清除的資源不會出現在搜尋回應中
- 正在清除的資源在處理期間可能暫時無法存取
- 病患隔室中的所有資源都會永久移除

HealthLake 的 FHIR \$questionnaire-package 操作

\$questionnaire-package 操作會擷取完整的套件，其中包含 FHIR 問卷及其轉譯和處理問卷所需的所有相依性。此操作會實作 [Da Vinci 文件範本和規則 \(DTR\) 實作指南](#)，針對醫療工作流程中的文件需求啟用動態表單轉譯。

運作方式

- 請求：您傳送識別所需問卷的參數（以及涵蓋範圍和順序內容）
- 擷取：HealthLake 會收集問卷和所有相依性 (ValueSets、CQL 程式庫等)
- 套件：所有資源都以標準化格式綁定在一起
- 回應：您收到準備轉譯和資料收集的完整套件

使用案例

- 預先授權文件：收集預先授權請求所需的臨床資訊
- 涵蓋範圍需求：收集滿足付款人涵蓋範圍需求所需的文件
- 臨床資料交換：結構臨床資料以提交給付款人

- 動態表單：使用預先填入的患者資料和條件式邏輯轉譯問卷

API 端點

```
POST /datastore/{datastoreId}/r4/Questionnaire/$questionnaire-package
Content-Type: application/fhir+json
```

請求參數

輸入參數

請求內文必須包含具有下列參數的 FHIR 參數資源：

參數	Type	基數	Description
coverage	涵蓋範圍	1..* (必要)	用於建立文件成員和涵蓋範圍的 (涵蓋範圍) 資源
questionnaire	正式	0..*	要傳回之特定問卷的正式 URL (s) (可能包含版本)
order	資源	0..*	訂購資源 (DeviceRequest、ServiceRequest、MedicationRequest、Encounter、Appointment) 以建立內容
changedSince	dateTime	0..1	如果存在，則只會傳回在此時間戳記之後變更的資源

參數驗證規則

至少必須提供下列其中一項 (除了必要的 之外coverage)：

- 一或多個questionnaire正式 URLs
- 一或多個order資源

有效的請求組合：

- coverage + questionnaire
- coverage + order
- coverage + questionnaire + order

範例請求

```
POST /datastore/example-datastore/r4/Questionnaire/$questionnaire-package
```

```
Content-Type: application/fhir+json
```

```
Authorization: Bearer <your-token>
```

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "coverage",
      "resource": {
        "resourceType": "Coverage",
        "id": "example-coverage",
        "status": "active",
        "beneficiary": {
          "reference": "Patient/example-patient"
        },
        "payor": [{
          "reference": "Organization/example-payer"
        }],
        "class": [{
          "type": {
            "coding": [{
              "system": "http://terminology.hl7.org/CodeSystem/coverage-class",
              "code": "group"
            }]
          },
          "value": "12345"
        }]
      }
    },
    {
      "name": "questionnaire",
      "valueCanonical": "http://example.org/fhir/Questionnaire/home-oxygen-therapy|2.0"
    }
  ]
}
```

```

    "name": "order",
    "resource": {
      "resourceType": "ServiceRequest",
      "id": "example-service-request",
      "status": "active",
      "intent": "order",
      "code": {
        "coding": [{
          "system": "http://www.ama-assn.org/go/cpt",
          "code": "94660",
          "display": "Continuous positive airway pressure ventilation (CPAP)"
        }]
      },
      "subject": {
        "reference": "Patient/example-patient"
      }
    },
    {
      "name": "changedSince",
      "valueDateTime": "2024-01-01T00:00:00Z"
    }
  ]
}

```

回應格式

成功回應 (200 OK)

操作會傳回包含一或多個套件套件的 FHIR 參數資源。每個套件套件都包含：

項目類型	基數	Description
問卷	1	要轉譯的問卷
QuestionnaireResponse	0..1	預先填入或部分完成的回應（如適用）
Library (程式庫)	0..*	包含預先填入和條件式邏輯的 CQL 程式庫
ValueSet	0..*	擴展的 ValueSets（適用於 <40 個擴展的答案選項）

Example 回應範例

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "PackageBundle",
      "resource": {
        "resourceType": "Bundle",
        "id": "questionnaire-package-example",
        "meta": {
          "profile": ["http://hl7.org/fhir/us/davinci-dtr/StructureDefinition/DTR-
QPackageBundle"]
        },
        "type": "collection",
        "timestamp": "2024-03-15T10:30:00Z",
        "entry": [
          {
            "fullUrl": "http://example.org/fhir/Questionnaire/home-oxygen-therapy",
            "resource": {
              "resourceType": "Questionnaire",
              "id": "home-oxygen-therapy",
              "url": "http://example.org/fhir/Questionnaire/home-oxygen-therapy",
              "version": "2.0",
              "status": "active",
              "title": "Home Oxygen Therapy Documentation",
              "item": [
                {
                  "linkId": "1",
                  "text": "Patient diagnosis",
                  "type": "choice",
                  "answerValueSet": "http://example.org/fhir/ValueSet/oxygen-diagnoses"
                }
              ]
            }
          }
        ]
      }
    },
    {
      "fullUrl": "http://example.org/fhir/Library/oxygen-prepopulation",
      "resource": {
        "resourceType": "Library",
        "id": "oxygen-prepopulation",
        "url": "http://example.org/fhir/Library/oxygen-prepopulation",
        "version": "1.0",

```

```
    "type": {
      "coding": [{
        "system": "http://terminology.hl7.org/CodeSystem/library-type",
        "code": "logic-library"
      }]
    },
    "content": [{
      "contentType": "text/cql",
      "data": "bGlicmFyeSBPeHlnZW5QcmVwb3B1bGF0aW9u..."
    }]
  }
},
{
  "fullUrl": "http://example.org/fhir/ValueSet/oxygen-diagnoses",
  "resource": {
    "resourceType": "ValueSet",
    "id": "oxygen-diagnoses",
    "url": "http://example.org/fhir/ValueSet/oxygen-diagnoses",
    "status": "active",
    "expansion": {
      "timestamp": "2024-03-15T10:30:00Z",
      "contains": [
        {
          "system": "http://hl7.org/fhir/sid/icd-10",
          "code": "J44.0",
          "display": "COPD with acute lower respiratory infection"
        },
        {
          "system": "http://hl7.org/fhir/sid/icd-10",
          "code": "J96.01",
          "display": "Acute respiratory failure with hypoxia"
        }
      ]
    }
  }
},
{
  "fullUrl": "http://example.org/fhir/QuestionnaireResponse/example-prepopulated",
  "resource": {
    "resourceType": "QuestionnaireResponse",
    "id": "example-prepopulated",
    "questionnaire": "http://example.org/fhir/Questionnaire/home-oxygen-therapy|2.0",
```

```

    "status": "in-progress",
    "subject": {
      "reference": "Patient/example-patient"
    },
    "basedOn": [{
      "reference": "ServiceRequest/example-service-request"
    }],
    "item": [
      {
        "linkId": "1",
        "text": "Patient diagnosis",
        "answer": [{
          "valueCoding": {
            "system": "http://hl7.org/fhir/sid/icd-10",
            "code": "J44.0",
            "display": "COPD with acute lower respiratory infection"
          }
        }
      ]
    }
  ]
},
{
  "name": "Outcome",
  "resource": {
    "resourceType": "OperationOutcome",
    "issue": [{
      "severity": "information",
      "code": "informational",
      "details": {
        "text": "Successfully retrieved questionnaire package"
      }
    }
  ]
}
]
}

```

操作工作流程

HealthLake 如何處理您的請求

當您呼叫 `$questionnaire-package` , HealthLake 會執行下列步驟：

1. 識別患者和付款人：從您的 `coverage` 參數擷取患者和保險組織。
2. 尋找正確問卷：
 - 搭配 `questionnaire` 參數：使用您提供的正式 URL
 - 使用 `order` 參數：比對訂單代碼 (CPT/HCPCS/LOINC) 和付款人，以尋找適當的問卷
3. 收集相依性：自動擷取轉譯問卷所需的一切：
 - CQL 程式庫 - 預先填入和條件式問題的邏輯
 - ValueSets - 答案選項 (如果 <40 個選項，則自動展開)
 - QuestionnaireResponse - 任何現有的進行中或已完成的回應
4. 將所有內容一起封裝：
 - 封裝所有資源 (每個資源僅包含一次)
 - 如果提供，則依 `changedSince` 時間戳記篩選
 - Outcome 如果遺失任何資源，將警告新增至

結果：準備進行轉譯的完整、獨立套件。

錯誤回應

400 錯誤的請求

請求驗證失敗時傳回。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "required",
    "details": {
      "text": "At least one of 'questionnaire' or 'order' must be provided along with 'coverage'"
    }
  }]
}
```

424 失敗相依性

當無法擷取相依資源時傳回。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "warning",
    "code": "not-found",
    "details": {
      "text": "Referenced Library 'http://example.org/fhir/Library/missing-library'
could not be retrieved"
    }
  ]
}
```

401 (未經授權)

當身分驗證憑證遺失或無效時傳回。

403 禁止

當已驗證的使用者缺少存取所請求資源的許可時傳回。

406 無法接受

當無法提供請求的內容類型時傳回。

409 衝突

發生版本或並行衝突時傳回。

410 Gone

當請求的資源永久刪除時傳回。

429 太多請求

超過速率限制時傳回。

500 內部伺服器錯誤

發生非預期的伺服器錯誤時傳回。

501 未實作

請求的操作尚未實作時傳回。

驗證規則

輸入驗證

- coverage 參數為必要 (1..* 基數)
- 至少order必須提供其中一個 questionnaire或
- 所有涵蓋範圍資源都必須是有效的 FHIR 資源
- 所有訂單資源必須是有效的 FHIR 資源
- 正式 URLs 必須正確格式化
- changedSince 必須是有效的 ISO 8601 dateTime

QuestionnaireResponse驗證

- status 必須適當 (in-progress、completed、amended)
- 結構必須符合參考的問卷
- basedOn 必須參考有效的訂單資源
- subject 必須參考有效的病患資源

資源重複資料刪除

- 每個資源只會在套件中出現一次
- 例外狀況：可能同時包含相同資源的不同版本
- 透過其正式 URL 和版本來識別資源

效能規格

指標	規格
資源計數限制	每個套件 500 個資源
套件大小限制	最多 5 MB

所需的許可

若要使用 \$questionnaire-package操作，請確定您的 IAM 角色具有：

- `healthlake:QuestionnairePackage` - 呼叫 操作
- `healthlake:ReadResource` - 擷取問卷和相依資源
- `healthlake:SearchWithPost` - 搜尋 `QuestionnaireResponse`和相關資源

FHIR 範圍上的 SMART

所需範圍下限：

- SMART v1 : `user/Questionnaire.read` `user/Library.read` `user/ValueSet.read` `user/QuestionnaireResponse.read`
- SMART v2 : `user/Questionnaire.rs` `user/Library.rs` `user/ValueSet.rs` `user/QuestionnaireResponse.rs`

重要實作備註

資源擷取策略

問卷識別優先順序：

- 正式 URL (如果提供 `questionnaire` 參數) - 最高優先順序
- 訂單分析 (如果提供 `order` 參數) :
 - 比對訂單代碼 (CPT、HCPCS、LOINC) 與付款人醫療政策
 - 使用涵蓋範圍付款人篩選付款人特定的問卷
 - 考慮其他內容的原因代碼

相依性解析

CQL 程式庫：

- 透過問卷資源上的 `cqf-library` 擴充功能擷取
- 透過 `Library.relatedArtifact` 類型以遞迴方式擷取相依程式庫 `depends-on`
- 套件中包含所有程式庫相依性

ValueSets：

- 如果包含的概念少於 40 個，則自動擴展
- 包含較大的 ValueSets，無需擴展
- 包括問卷和程式庫資源中參考 ValueSets

QuestionnaireResponse 預先填入

下列情況下，操作可能會傳回具有預先填入資料的 QuestionnaireResponse：

- 找到現有的進行中或已完成的回應
- 關聯程式庫中的 CQL 邏輯可以從病患記錄擷取資料
- 回應會連結到相關順序和涵蓋範圍

QuestionnaireResponse 的搜尋條件：

搜尋參數	FHIR 路徑	Description
based-on	QuestionnaireResponse.basedOn	ServiceRequest 或 CarePlan 的連結
patient	QuestionnaireResponse.subject	做為主體的患者
questionnaire	QuestionnaireResponse.questionnaire	回答的問卷

變更資源篩選

提供 changedSince 參數時：

- 僅包含指定時間戳記後修改的資源
- 如果沒有變更任何資源，200 OK 會以空的套件傳回
- 適用於增量更新和快取策略
- 時間戳記比較使用資源 meta.lastUpdated 欄位

多個套件套件

下列情況下，操作可能會傳回多個套件套件：

- 透過正式 URLs 請求多個問卷
- 多個訂單需要不同的問卷
- 相同問卷的不同版本適用

每個套件套件都與所有必要的相依性獨立。

常用案例

使用案例 1：預先授權文件

案例：提供者需要收集家庭供氧治療預先授權的文件。

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "coverage",
      "resource": { /* Patient's insurance coverage */ }
    },
    {
      "name": "order",
      "resource": {
        "resourceType": "ServiceRequest",
        "code": {
          "coding": [{
            "system": "http://www.ama-assn.org/go/cpt",
            "code": "94660"
          }]
        }
      }
    }
  ]
}
```

結果：傳回包含氧氣治療問卷的套件，預先填入來自 EHR 的患者生命值和診斷碼。

使用案例 2：擷取特定問卷版本

案例：供應商需要特定版本的問卷才能合規。

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "coverage",
      "resource": { /* Coverage resource */ }
    },
    {
      "name": "questionnaire",
      "valueCanonical": "http://example.org/fhir/Questionnaire/dme-request|3.1.0"
    }
  ]
}
```

結果：傳回含所有相依性的確切版本 3.1.0 的 DME 請求問卷。

使用案例 3：檢查更新

案例：供應商想要檢查自上次擷取以來是否有任何問卷資源已更新。

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "coverage",
      "resource": { /* Coverage resource */ }
    },
    {
      "name": "questionnaire",
      "valueCanonical": "http://example.org/fhir/Questionnaire/medication-request"
    },
    {
      "name": "changedSince",
      "valueDateTime": "2024-03-01T00:00:00Z"
    }
  ]
}
```

結果：僅傳回在 2024 年 3 月 1 日之後修改的資源，如果沒有變更，則傳回空白套件。

使用案例 4：多個訂單

案例：供應商提交多個可能需要不同問卷的服務請求。

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "coverage",
      "resource": { /* Coverage resource */ }
    },
    {
      "name": "order",
      "resource": { /* ServiceRequest for imaging */ }
    },
    {
      "name": "order",
      "resource": { /* ServiceRequest for DME */ }
    }
  ]
}
```

結果：傳回多個套件套件，每個適用的問卷各一個。

與其他 Da Vinci IGs 整合

涵蓋範圍需求探索 (CRD)

工作流程整合：

- 供應商在其 EHR 中訂購服務
- CRD 勾點觸發，檢查涵蓋範圍需求
- 付款人回應，指出需要文件
- 供應商呼叫 `$questionnaire-package` 以擷取文件表單
- 供應商完成問卷
- 文件是透過 PAS 或 CDex 提交

預先授權支援 (PAS)

工作流程整合：

- 使用 `$questionnaire-package` 擷取文件需求
- 使用所需的臨床資料完成 `QuestionnaireResponse`
- 使用 `Claim/$submit` 搭配已完成的 `QuestionnaireResponse` 提交預先授權
- 使用 檢查狀態 `Claim/$inquire`

臨床資料交換 (CDex)

工作流程整合：

- 付款人請求申請其他文件
- 提供者使用 `$questionnaire-package` 擷取結構化資料收集表單
- 供應商完成 `QuestionnaireResponse`
- 文件會透過 CDex 連接工作流程提交給付款人

故障診斷指南

問題：未傳回問卷

可能的原因：

- 正式 URL 不符合資料存放區中的任何問卷
- 訂單代碼未對應至付款人醫療政策中的任何問卷
- 涵蓋範圍付款人沒有相關聯的問卷

解決方案：

- 確認正式 URL 正確且問卷存在
- 檢查訂單代碼 (CPT/HCPCS) 是否正確指定
- 確認付款人組織已設定問卷

問題：套件中缺少相依性

可能的原因：

- 參考的程式庫或 ValueSet 不存在於資料存放區中
- 程式庫參考已中斷或不正確
- ValueSet 擴展失敗

解決方案：

- 檢查 Outcome 參數是否有資源遺失的相關警告
- 驗證資料存放區中存在所有參考的資源
- 確保 ValueSet URLs 正確且可解決

問題：含 changedSince 的空套件

可能的原因：

- 這是預期的行為 - 自指定的時間戳記以來未修改任何資源

解決方案：

- 使用套件的快取版本
- 移除 changedSince 參數以擷取完整套件

問題：未預先填入 QuestionnaireResponse

可能的原因：

- 找不到現有的 QuestionnaireResponse
- CQL Library 邏輯無法擷取所需的資料
- 病患資料遺失或不完整

解決方案：

- 這可能是預期的 - 並非所有問卷都有預先填入邏輯
- 檢查資料存放區中是否存在病患資料

- 檢閱 CQL Library 邏輯以了解資料擷取需求

最佳實務

1. 搭配版本使用正式 URLs

請求特定問卷時，請務必指定版本編號：

```
{
  "name": "questionnaire",
  "valueCanonical": "http://example.org/fhir/Questionnaire/dme|2.1.0"
}
```

原因：確保一致性，並防止更新問卷時的意外變更。

2. 利用 changedSince for Performance

對於經常存取的問題，請使用 changedSince 將資料傳輸降至最低：

```
{
  "name": "changedSince",
  "valueDateTime": "2024-03-10T15:30:00Z"
}
```

原因：僅擷取更新的資源，以減少延遲和頻寬使用量。

3. 包含完整涵蓋範圍資訊

提供全面的涵蓋範圍詳細資訊，以確保正確的問卷選擇：

```
{
  "name": "coverage",
  "resource": {
    "resourceType": "Coverage",
    "beneficiary": { "reference": "Patient/123" },
    "payor": [ { "reference": "Organization/payer-abc" } ],
    "class": [ { /* Group/plan information */ } ]
  }
}
```

原因：協助 HealthLake 識別付款人特定的問卷和要求。

相關的 操作

- Claim/\$submit - 提交包含已完成文件的預先授權請求
- Claim/\$inquire - 檢查提交的預先授權狀態
- ValueSet/\$expand - 展開用於答案選項的 ValueSets

HealthLake 的 FHIR \$submit 操作

\$submit 此操作可讓您以電子方式提交預先授權請求給付款人以進行核准。此操作實作 [Da Vinci 預先授權支援 \(PAS\) 實作指南](#)，為預先授權提交提供標準化的 FHIR 工作流程。

運作方式

- 提交：您傳送 FHIR 套件，其中包含您的預先授權請求和支援的臨床資料
- 驗證：HealthLake 會根據 PAS 要求驗證提交
- 持久性：所有資源都存放在 HealthLake 資料存放區中
- 回應：您立即收到「佇列」狀態的回應
- 程序：由付款人非同步處理授權決策

API 端點

```
POST /datastore/{datastoreId}/r4/Claim/$submit
Content-Type: application/fhir+json
```

請求結構

套件需求

您的請求必須是具有下列項目的 FHIR 套件資源：

- Bundle.type：必須為 "collection"
- Bundle.entry：必須包含一個具有的宣告資源 use = "preauthorization"
- 參考資源：宣告參考的所有資源必須包含在套件中

必要的資源

資源	基數	設定檔	Description
宣告	1	PAS 宣告	預先授權請求
病患	1	PAS 病患	病患人口統計資訊
Organization (保險業者)	1	PAS Insurer	保險公司
Organization (提供者)	1	PAS 請求者	提交請求的醫療保健供應商
涵蓋範圍	1 個或多個	PAS 涵蓋範圍	保險涵蓋範圍詳細資訊

選用資源

資源	基數	設定檔	Description
從業人員	0 或以上	PAS 從業人員	醫療保健從業人員
PractitionerRole	0 或以上	PAS PractitionerRole	從業人員角色
ServiceRequest	0 或以上	PAS ServiceRequest	請求的醫療保健
DeviceRequest	0 或以上	PAS DeviceRequest	請求的醫療設備
MedicationRequest	0 或以上	PAS MedicationRequest	請求的藥品
DocumentReference	0 或以上	PAS DocumentReference	支援臨床文件

範例請求

```
POST /datastore/example-datastore/r4/Claim/$submit
Content-Type: application/fhir+json
Authorization: Bearer <your-token>

{
  "resourceType" : "Bundle",
  "id" : "MedicalServicesAuthorizationBundleExample",
  "meta" : {
    "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-pas-request-bundle"]
  },
  "identifier" : {
    "system" : "http://example.org/SUBMITTER_TRANSACTION_IDENTIFIER",
    "value" : "5269367"
  },
  "type" : "collection",
  "timestamp" : "2005-05-02T11:01:00+05:00",
  "entry" : [{
    "fullUrl" : "http://example.org/fhir/Claim/MedicalServicesAuthorizationExample",
    "resource" : {
      "resourceType" : "Claim",
      "id" : "MedicalServicesAuthorizationExample",
      "meta" : {
        "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claim"]
      },
      "identifier" : [{
        "system" : "http://example.org/PATIENT_EVENT_TRACE_NUMBER",
        "value" : "111099"
      }],
      "status" : "active",
      "type" : {
        "coding" : [{
          "system" : "http://terminology.hl7.org/CodeSystem/claim-type",
          "code" : "professional"
        }]
      },
      "use" : "preauthorization",
      "patient" : {
        "reference" : "Patient/SubscriberExample"
      },
      "created" : "2005-05-02T11:01:00+05:00",
```

```
"insurer" : {
  "reference" : "Organization/InsurerExample"
},
"provider" : {
  "reference" : "Organization/UMOExample"
},
"priority" : {
  "coding" : [{
    "system" : "http://terminology.hl7.org/CodeSystem/processpriority",
    "code" : "normal"
  }]
},
"insurance" : [{
  "sequence" : 1,
  "focal" : true,
  "coverage" : {
    "reference" : "Coverage/InsuranceExample"
  }
}],
"item" : [{
  "extension" : [{
    "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
serviceItemRequestType",
    "valueCodeableConcept" : {
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/1525",
        "code" : "IN",
        "display" : "Initial Medical Services Reservation"
      }]
    }
  ]
},
{
  "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
certificationType",
  "valueCodeableConcept" : {
    "coding" : [{
      "system" : "https://codesystem.x12.org/005010/1322",
      "code" : "I",
      "display" : "Initial"
    }]
  }
}
],
{
```

```

        "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
authorizationNumber",
        "valueString" : "1122344"
    },
    {
        "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
administrationReferenceNumber",
        "valueString" : "33441122"
    }
}],
"sequence" : 1,
"category" : {
    "coding" : [{
        "system" : "https://codesystem.x12.org/005010/1365",
        "code" : "1",
        "display" : "Medical Care"
    }
    ]
},
"productOrService" : {
    "coding" : [{
        "system" : "http://www.cms.gov/Medicare/Coding/HCPCSReleaseCodeSets",
        "code" : "99212",
        "display" : "Established Office Visit"
    }
    ]
},
"servicedDate" : "2005-05-10",
"locationCodeableConcept" : {
    "coding" : [{
        "system" : "https://www.cms.gov/Medicare/Coding/place-of-service-codes/
Place_of_Service_Code_Set",
        "code" : "11"
    }
    ]
}
}
}],
}
},
{
    "fullUrl" : "http://example.org/fhir/Organization/UM0Example",
    "resource" : {
        "resourceType" : "Organization",
        "id" : "UM0Example",
        "meta" : {
            "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
requestor"]
        }
    }
},

```

```

    "identifier" : [{
      "system" : "http://hl7.org/fhir/sid/us-npi",
      "value" : "8189991234"
    }],
    "active" : true,
    "type" : [{
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/98",
        "code" : "X3"
      }]
    }],
    "name" : "DR. JOE SMITH CORPORATION",
    "address" : [{
      "line" : ["111 1ST STREET"],
      "city" : "SAN DIEGO",
      "state" : "CA",
      "postalCode" : "92101",
      "country" : "US"
    }]
  }
},
{
  "fullUrl" : "http://example.org/fhir/Organization/InsurerExample",
  "resource" : {
    "resourceType" : "Organization",
    "id" : "InsurerExample",
    "meta" : {
      "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-insurer"]
    },
    "identifier" : [{
      "system" : "http://hl7.org/fhir/sid/us-npi",
      "value" : "1234567893"
    }],
    "active" : true,
    "type" : [{
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/98",
        "code" : "PR"
      }]
    }],
    "name" : "MARYLAND CAPITAL INSURANCE COMPANY"
  }
},

```

```
{
  "fullUrl" : "http://example.org/fhir/Coverage/InsuranceExample",
  "resource" : {
    "resourceType" : "Coverage",
    "id" : "InsuranceExample",
    "meta" : {
      "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-coverage"]
    },
    "status" : "active",
    "subscriberId" : "1122334455",
    "beneficiary" : {
      "reference" : "Patient/SubscriberExample"
    },
    "relationship" : {
      "coding" : [{
        "system" : "http://terminology.hl7.org/CodeSystem/subscriber-relationship",
        "code" : "self"
      }],
      {
        "system" : "https://codesystem.x12.org/005010/1069",
        "code" : "18"
      }
    ]
  },
  "payor" : [{
    "reference" : "Organization/InsurerExample"
  }]
},
{
  "fullUrl" : "http://example.org/fhir/Patient/SubscriberExample",
  "resource" : {
    "resourceType" : "Patient",
    "id" : "SubscriberExample",
    "meta" : {
      "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-subscriber"]
    },
    "extension" : [{
      "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-militaryStatus",
      "valueCodeableConcept" : {
        "coding" : [{
          "system" : "https://codesystem.x12.org/005010/584",
```

```

        "code" : "RU"
      ]]
    }
  ]],
  "identifier" : [{
    "type" : {
      "coding" : [{
        "system" : "http://terminology.hl7.org/CodeSystem/v2-0203",
        "code" : "MB"
      }]
    },
    "system" : "http://example.org/MIN",
    "value" : "12345678901"
  }],
  "name" : [{
    "family" : "SMITH",
    "given" : ["JOE"]
  }],
  "gender" : "male"
}
]]
}

```

回應格式

成功回應 (200 OK)

您將會收到 PAS 回應套件，其中包含：

- 使用 `outcome: "queued"` 和的 `ClaimResponse status: "active"`
- 請求中的所有原始資源
- 確認接收的時間戳記

```

{
  "resourceType" : "Bundle",
  "identifier": {
    "system": "http://example.org/SUBMITTER_TRANSACTION_IDENTIFIER",
    "value": "5269367"
  },
  "type" : "collection",
  "timestamp" : "2005-05-02T11:02:00+05:00",
  "entry" : [{

```

```

    "fullUrl" : "http://example.org/fhir/ClaimResponse/
PractitionerRequestorPendingResponseExample",
    "resource" : {
      "resourceType" : "ClaimResponse",
      "id" : "PractitionerRequestorPendingResponseExample",
      "meta" : {
        "profile" : ["http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
claimresponse"]
      },
      "identifier" : [{
        "system" : "http://example.org/PATIENT_EVENT_TRACE_NUMBER",
        "value" : "111099"
      }],
      "status" : "active",
      "type" : {
        "coding" : [{
          "system" : "http://terminology.hl7.org/CodeSystem/claim-type",
          "code" : "professional"
        }]
      },
      "use" : "preauthorization",
      "patient" : {
        "reference" : "Patient/SubscriberExample"
      },
      "created" : "2005-05-02T11:02:00+05:00",
      "insurer" : {
        "reference" : "Organization/InsurerExample"
      },
      "requestor" : {
        "reference" : "PractitionerRole/ReferralPractitionerRoleExample"
      },
      "request" : {
        "reference" : "Claim/MedicalServicesAuthorizationExample"
      },
      "outcome" : "queued"
    }
  },
  {
    "fullUrl" : "http://example.org/fhir/Claim/MedicalServicesAuthorizationExample",
    "resource" : {
      "resourceType" : "Claim",
      "id" : "MedicalServicesAuthorizationExample",
      "meta" : {
        "profile": [

```

```

        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claim",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-claim|
2.1.0"
    ]
  },
  "identifier" : [{
    "system" : "http://example.org/PATIENT_EVENT_TRACE_NUMBER",
    "value" : "111099"
  }
  ]],
  "status" : "active",
  "type" : {
    "coding" : [{
      "system" : "http://terminology.hl7.org/CodeSystem/claim-type",
      "code" : "professional"
    }
    ]
  },
  "use" : "preauthorization",
  "patient" : {
    "reference" : "Patient/SubscriberExample"
  },
  "created" : "2005-05-02T11:01:00+05:00",
  "insurer" : {
    "reference" : "Organization/InsurerExample"
  },
  "provider" : {
    "reference" : "Organization/UMOExample"
  },
  "priority" : {
    "coding" : [{
      "system" : "http://terminology.hl7.org/CodeSystem/processpriority",
      "code" : "normal"
    }
    ]
  },
  "insurance" : [{
    "sequence" : 1,
    "focal" : true,
    "coverage" : {
      "reference" : "Coverage/InsuranceExample"
    }
  }
  ]],
  "item" : [{
    "extension" : [{

```

```
    "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
serviceItemRequestType",
    "valueCodeableConcept" : {
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/1525",
        "code" : "IN",
        "display" : "Initial Medical Services Reservation"
      }]
    }
  },
  {
    "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
certificationType",
    "valueCodeableConcept" : {
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/1322",
        "code" : "I",
        "display" : "Initial"
      }]
    }
  },
  {
    "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
authorizationNumber",
    "valueString" : "1122344"
  },
  {
    "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-
administrationReferenceNumber",
    "valueString" : "33441122"
  }
}],
"sequence" : 1,
"category" : {
  "coding" : [{
    "system" : "https://codesystem.x12.org/005010/1365",
    "code" : "1",
    "display" : "Medical Care"
  }]
},
"productOrService" : {
  "coding" : [{
    "system" : "http://www.cms.gov/Medicare/Coding/HCPCSReleaseCodeSets",
    "code" : "99212",
    "display" : "Established Office Visit"
  }]
}
```

```

    ]]
  },
  "servicedDate" : "2005-05-10",
  "locationCodeableConcept" : {
    "coding" : [{
      "system" : "https://www.cms.gov/Medicare/Coding/place-of-service-codes/
Place_of_Service_Code_Set",
      "code" : "11"
    }]
  }
}]
},
{
  "fullUrl" : "http://example.org/fhir/Organization/UMOExample",
  "resource" : {
    "resourceType" : "Organization",
    "id" : "UMOExample",
    "meta" : {
      "profile": [
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-requestor",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-requestor|
2.1.0"
      ]
    }
  },
  "identifier" : [{
    "system" : "http://hl7.org/fhir/sid/us-npi",
    "value" : "8189991234"
  }],
  "active" : true,
  "type" : [{
    "coding" : [{
      "system" : "https://codesystem.x12.org/005010/98",
      "code" : "X3"
    }]
  }],
  "name" : "DR. JOE SMITH CORPORATION",
  "address" : [{
    "line" : ["111 1ST STREET"],
    "city" : "SAN DIEGO",
    "state" : "CA",
    "postalCode" : "92101",
    "country" : "US"
  }
}

```

```

    ]]
  }
},
{
  "fullUrl" : "http://example.org/fhir/Organization/InsurerExample",
  "resource" : {
    "resourceType" : "Organization",
    "id" : "InsurerExample",
    "meta" : {
      "profile": [
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
insurer",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
insurer|2.1.0"
      ]
    },
    "identifier" : [{
      "system" : "http://hl7.org/fhir/sid/us-npi",
      "value" : "1234567893"
    }],
    "active" : true,
    "type" : [{
      "coding" : [{
        "system" : "https://codesystem.x12.org/005010/98",
        "code" : "PR"
      }]
    }],
    "name" : "MARYLAND CAPITAL INSURANCE COMPANY"
  }
},
{
  "fullUrl" : "http://example.org/fhir/Coverage/InsuranceExample",
  "resource" : {
    "resourceType" : "Coverage",
    "id" : "InsuranceExample",
    "meta": {
      "profile": [
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
coverage",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-
coverage|2.1.0"
      ]
    },
    "status" : "active",

```

```

    "subscriberId" : "1122334455",
    "beneficiary" : {
      "reference" : "Patient/SubscriberExample"
    },
    "relationship" : {
      "coding" : [{
        "system" : "http://terminology.hl7.org/CodeSystem/subscriber-relationship",
        "code" : "self"
      }],
      {
        "system" : "https://codesystem.x12.org/005010/1069",
        "code" : "18"
      }
    ],
    "payor" : [{
      "reference" : "Organization/InsurerExample"
    }]
  }
},
{
  "fullUrl" : "http://example.org/fhir/Patient/SubscriberExample",
  "resource" : {
    "resourceType" : "Patient",
    "id" : "SubscriberExample",
    "meta": {
      "profile": [
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-subscriber",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-beneficiary",
        "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/profile-beneficiary|2.1.0"
      ]
    },
    "extension" : [{
      "url" : "http://hl7.org/fhir/us/davinci-pas/StructureDefinition/extension-militaryStatus",
      "valueCodeableConcept" : {
        "coding" : [{
          "system" : "https://codesystem.x12.org/005010/584",
          "code" : "RU"
        }]
      }
    }]
  }
}],

```

```
"identifier" : [{
  "type" : {
    "coding" : [{
      "system" : "http://terminology.hl7.org/CodeSystem/v2-0203",
      "code" : "MB"
    }]
  },
  "system" : "http://example.org/MIN",
  "value" : "12345678901"
}],
"name" : [{
  "family" : "SMITH",
  "given" : ["JOE"]
}],
"gender" : "male"
}
}]
}
```

錯誤回應

400 錯誤的請求

當請求格式無效或格式不正確時傳回。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "invalid",
    "diagnostics": "The provided payload was invalid and could not be parsed
correctly."
  }]
}
```

412 先決條件失敗

已提交相同的預先授權請求時傳回（偵測到重複提交）。

```
{
  "resourceType": "OperationOutcome",
  "issue": [{
```

```

    "severity": "error",
    "code": "processing",
    "diagnostics": "PreAuth Claim already exists"
  ]}
}
```

冪等性

`$submit` 操作是冪等的。多次提交相同的請求不會建立重複的預先授權請求。反之，您會收到 412 錯誤，指示您使用 `$inquire` 來檢查原始提交的狀態。

422 無法處理的實體

FHIR 驗證失敗時傳回。

```

{
  "resourceType": "OperationOutcome",
  "issue": [{
    "severity": "error",
    "code": "required",
    "diagnostics": "Bundle contains more than one preauthorization claim"
  }]
}
```

驗證規則

HealthLake 會對您的提交執行全面的驗證：

套件驗證

- 必須符合 PAS 請求套件描述檔
- `Bundle.type` 必須 "collection"
- 可以包含多個宣告資源
- 不過，必須包含一個具有預先授權使用的宣告資源
 - 此宣告資源必須是套件中的第一個項目
- 套件中必須包含所有參考的資源

宣告驗證

- 必須符合 PAS 宣告描述檔
- Claim.use 必須 "preauthorization"
- 必要欄位 : patient、insurer、provider、created、priority
- 業務識別符必須存在且有效

資源驗證

- 所有資源必須符合其各自的 PAS 設定檔
- 必要的支援資源必須存在 (病患、涵蓋範圍、組織)
- 交互參考在套件中必須是有效且可解析的

效能規格

指標	規格
套件大小限制	最多 5 MB
資源計數限制	每個套件 500 個資源

所需的許可

若要使用 \$submit 操作，使用者可以在 FHIR 上使用 AWS Sigv4 或 SMART：

- 確保您的 IAM 角色具有：healthlake:SubmitPreAuthClaim- 呼叫 操作

FHIR 範圍上的 SMART

所需範圍下限：

- SMART v1 : user/Claim.write & <all_resourceTypes_in_Bundle>.write
- SMART v2 : user/Claim.c & <all_resourceTypes_in_Bundle>.c or system/*.*

重要實作備註

資源持久性

- 所有套件項目都會保留為資料存放區中的個別 FHIR 資源
- 客戶提供IDs 會在提供時保留
- 維護版本歷史記錄以供稽核之用
- 重複偵測可防止資源衝突

處理行為

- 每個有效的提交只會產生一個 ClaimResponse "queued"結果
- 無效提交會傳回 400 或 422 狀態碼，其中包含詳細的錯誤資訊
- 系統錯誤會傳回適當的 5xx 狀態碼
- 所有成功的提交都會傳回 200 狀態與待定的 ClaimResponse

套件需求

- `Bundle.entry.fullUrl` 值必須是 REST URLs或"`urn:uuid:[guid]`"格式
- 所有 GUIDs提交之間必須是唯一的（相同資源執行個體除外）
- 參考的資源必須存在於套件中或可解析

相關的 操作

- `Claim/$inquire` - 查詢提交的預先授權請求的狀態
- `Patient/$everything` - 針對預先授權內容擷取完整的患者資料

使用 驗證 FHIR 資源 `$validate`

AWS HealthLake 現在支援 FHIR 資源`$validate`的操作，可讓您根據 FHIR 規格驗證資源，並檢查其是否符合指定的設定檔或基本資源定義，而無需執行任何儲存操作。當您需要：

- 驗證 FHIR CMS 合規要求
- 在生產環境中使用資源之前進行測試
- 在使用者編輯臨床資料時提供即時驗證意見回饋

- 減少無效的資料提交以建立和更新 APIs

Usage

您可以使用 POST 方法在 FHIR 資源上叫用 \$validate 操作：

受支援的 操作

```
POST [base]/[type]/[id]/$validate
POST [base]/[type]/$validate
```

支援的承載

參數資源

HealthLake 支援下列 FHIR \$validate 參數：

參數	Type	必要	Description
resource	資源	是	要驗證的資源
profile	正式	否	要驗證之設定檔的正式 URL
mode	code	否	驗證模式：create 或 update

具有查詢參數的直接資源

參數	Type	必要	Description
profile	正式	否	要驗證之設定檔的正式 URL
mode	code	否	驗證模式：create 或 update

範例

具有 ID 和參數承載之資源的 POST 請求

```
POST [base]/Patient/example-patient/$validate
```

```
Content-Type: application/fhir+json
```

```
{
  "resourceType": "Parameters",
  "parameter": [
    {
      "name": "resource",
      "resource": {
        "resourceType": "Patient",
        "id": "example-patient",
        "name": [
          {
            "family": "Smith",
            "given": ["John"]
          }
        ],
        "gender": "male",
        "birthDate": "1990-01-01"
      }
    },
    {
      "name": "profile",
      "valueCanonical": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
    },
    {
      "name": "mode",
      "valueString": "create"
    }
  ]
}
```

資源類型和參數承載的 POST 請求

```
POST [base]/Patient/$validate
```

```
Content-Type: application/fhir+json
```

```
{
  "resourceType": "Parameters",
  "parameter": [
```

```

{
  "name": "resource",
  "resource": {
    "resourceType": "Patient",
    "name": [
      {
        "family": "Doe",
        "given": ["Jane"]
      }
    ],
    "gender": "female",
    "birthDate": "1985-05-15"
  }
},
{
  "name": "profile",
  "valueCanonical": "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient"
},
{
  "name": "mode",
  "valueString": "update"
}
]
}

```

具有 ID 和直接資源承載的資源 POST 請求

POST [base]/Patient/example-patient/\$validate?profile=<http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient>&mode=create

Content-Type: application/fhir+json

```

{
  "resourceType": "Patient",
  "id": "example-patient",
  "name": [
    {
      "family": "Smith",
      "given": ["John"]
    }
  ],
  "gender": "male",
  "birthDate": "1990-01-01"
}

```

```
}
```

資源類型和直接資源承載的 POST 請求

```
POST [base]/Patient/$validate?profile=http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient&mode=create
```

```
Content-Type: application/fhir+json
```

```
{
  "resourceType": "Patient",
  "id": "example-patient",
  "name": [
    {
      "family": "Smith",
      "given": ["John"]
    }
  ],
  "gender": "male",
  "birthDate": "1990-01-01"
}
```

回應範例

操作會傳回具有驗證結果的 OperationOutcome 資源：

```
{
  "resourceType": "OperationOutcome",
  "issue": [
    {
      "severity": "information",
      "code": "informational",
      "diagnostics": "Validation successful"
    }
  ]
}
```

驗證錯誤的回應範例

```
{
  "resourceType": "OperationOutcome",
  "issue": [
```

```

{
  "severity": "error",
  "code": "required",
  "details": {
    "text": "Missing required element"
  },
  "diagnostics": "Patient.identifier is required by the US Core Patient profile",
  "location": [
    "Patient.identifier"
  ]
},
{
  "severity": "warning",
  "code": "code-invalid",
  "details": {
    "text": "Invalid code value"
  },
  "diagnostics": "The provided gender code is not from the required value set",
  "location": [
    "Patient.gender"
  ]
}
]
}

```

Behavior (行為)

\$validate 操作：

1. 根據 FHIR 規格和基本資源定義驗證資源
2. 提供 profile 參數時，檢查是否符合指定的設定檔
3. 根據指定的模式進行驗證 (create 或 update)
4. 傳回詳細的驗證結果，包括錯誤、警告和資訊性訊息
5. 不執行任何儲存操作 - 僅驗證
6. 在可執行驗證時傳回 HTTP 200 OK，無論是否發現驗證問題

驗證模式

- create：將資源驗證為正在建立（新資源）
- 更新：將資源驗證為正在更新（現有資源）

錯誤處理

操作會傳回：

- 200 OK：已成功執行驗證（無論驗證結果為何）
- 400 錯誤的請求：無效的請求格式或參數
- 找不到 404：找不到資源類型或設定檔

如需 `$validate` 操作規格的詳細資訊，請參閱 [FHIR R4 資源\\$validate](#) 文件。

的合規參考 AWS HealthLake

AWS HealthLake 提供旨在協助您追蹤和報告 API 用量的功能，以符合 CMS (Centers for Medicare & Medicaid Services) 互通性需求。這些功能可讓您依 CMS 指定的類別來分類 API 交易，並自動擷取用量指標以供合規報告之用。

了解您的合規責任

使用 AWS HealthLake 及其 CMS 互通性端點本身不足以達成 CMS 合規。您負責：

- 根據您的特定使用案例和法規義務，將 API 工作流程正確映射至適當的 CMS 類別端點
- 實作符合 CMS 要求的適當身分驗證和授權控制
- 確保您的 FHIR 資源和資料交換符合適用的 CMS 法規和實作指南
- 設定和監控 CloudWatch 指標，以支援您的合規報告需求
- 了解哪些 CMS 規則適用於您的組織，並實作適當的技術和操作控制

AWS HealthLake 提供基礎設施和工具以支援您的合規工作，但您必須根據您的特定法規要求適當地使用這些功能。只要透過這些端點路由 API 呼叫，您的應用程式就不會自動符合 CMS 法規。

主題

- [CMS 合規功能](#)

CMS 合規功能

AWS HealthLake 提供功能，協助您符合 CMS (Centers for Medicare & Medicaid Services) 互通性和合規要求。這些功能可讓您依 CMS 類別追蹤 API 用量，然後基於合規目的報告用量指標。

主題

- [CMS 互通性端點](#)
- [CMS 合規的增強型 CloudWatch 指標](#)

CMS 互通性端點

概觀

HealthLake 提供四個對應於 CMS 指定 API 類別的 CMS 互通性端點。HealthLake 資料存放區的基礎 URL 不會變更。這些端點僅提供針對 CMS 報告目的分類和追蹤 API 呼叫的方法。

用途

這些互通性端點的主要目的是讓客戶能夠：

- 依 CMS 類別輕鬆追蹤 API 交易
- 自動報告 CMS 合規的用量指標
- 以最少的變更維護現有的 FHIR 工作流程

無論您使用互通性端點或標準 FHIR 端點，所有 API 呼叫的功能都相同 - 唯一的區別是如何在 CloudWatch 指標中分類交易。

支援的 CMS 互通性端點

CMS 類別	互通性端點	使用範例
病患存取	/patientaccess/v2/r4	baseURL/patientaccess/v2/r4/Patient/123
提供者存取	/provideraccess/v2/r4	baseURL/provideraccess/v2/r4/Observation?patient=123
付款人到付款人	/payertopayerdx/v2/r4	baseURL/payertopayerdx/v2/r4/Practitioner/456
先前的驗證服務	/priorauthservice/v2/r4	baseURL/priorauthservice/v2/r4/ExplanationOfBenefit?patient=789

互通性端點的運作方式

標準 HealthLake API 呼叫：

```
baseUrl/resourceType/[id]  
baseUrl/resourceType?[parameters]
```

使用 CMS 互通性端點：

```
baseUrl/interoperability-endpoint/resourceType/[id]  
baseUrl/interoperability-endpoint/resourceType?[parameters]
```

互通性端點路徑只會插入您的基本 URL 和資源類型之間。互通性端點路徑之後的所有內容都與您目前的 API 呼叫完全相同。

使用範例

範例 1：病患存取 API

目前的 API 呼叫（仍然有效）：

```
curl -X GET \  
  https://healthlake.us-east-1.amazonaws.com/datastore/abc123/r4/Patient/123 \  
  -H "Authorization: Bearer <token>" \  
  -H "Content-Type: application/fhir+json"
```

使用病患存取互通性端點（用於 CMS 追蹤）：

```
curl -X GET \  
  https://healthlake.us-east-1.amazonaws.com/datastore/abc123/patientaccess/v2/r4/  
  Patient/123 \  
  -H "Authorization: Bearer <token>" \  
  -H "Content-Type: application/fhir+json"
```

重點：

- 保留基本 URL：https://healthlake.us-east-1.amazonaws.com/datastore/abc123
- 插入的互通性端點：/patientaccess/v2/r4
- 資源路徑不變：/Patient/123

- 兩個呼叫都會傳回相同的回應
- CloudWatch URIType=patient-access中的 會自動追蹤互通性端點呼叫
- POST、PUT、PATCH、DELETE 操作的運作方式相同。
- 請求內文保持不變。

端點轉譯參考

互通性端點	轉換為	CMS 類別
baseURL/patientaccess/v2/r4/Patient	baseURL/r4/Patient	病患存取
baseURL/provideraccess/v2/r4/Observation	baseURL/r4/Observation	提供者存取
baseURL/payertopayerdx/v2/r4/Practitioner/456	baseURL/r4/Practitioner/456	付款人對付款人資料交換
baseURL/priorauthservice/v2/r4/ExplanationOfBenefit?patient=789	baseURL/r4/ExplanationOfBenefit?patient=789	預先授權

重要說明

- 沒有功能差異：互通性端點和標準 FHIR 端點會傳回相同的回應，並支援相同的操作
- 基本 URL 未變更：您的 HealthLake 資料存放區端點保持不變
- 簡易整合：在您的基本 URL 和資源類型之間插入互通性端點路徑
- 自動追蹤：CloudWatch 指標會依使用的互通性端點自動分類呼叫
- 回溯相容：沒有互通性端點的現有 API 呼叫會繼續正常運作

CMS 合規的增強型 CloudWatch 指標

概觀

當您使用 CMS 互通性端點時，HealthLake 會自動發出具有額外維度的增強型 CloudWatch 指標，以支援 CMS 報告需求。這些指標會依呼叫者身分、應用程式和 CMS 特定的 URI 類型追蹤 API 用量，而不需要任何其他組態。

運作方式

當您使用互通性端點進行 API 呼叫時：

```
# This call...
curl https://healthlake.us-east-1.amazonaws.com/datastore/abc123/patientaccess/v2/r4/Patient/123

# Automatically generates metrics with:
# - URIType: "patient-access"
# - Sub: extracted from your bearer token (SMART on FHIR datastores only)
# - ClientId: extracted from your bearer token (SMART on FHIR datastores only)
# - Plus all standard dimensions (DatastoreId, Operation, etc.)
```

不需要額外的程式碼或組態。只需使用互通性端點，即可自動擷取增強型指標。

Note

對於 FHIR 資料存放區上的非 SMART，仍會擷取 URIType 維度，讓您可以依 CMS 類別追蹤 API 用量。只有在 FHIR 身分驗證上使用 SMART 搭配包含這些宣告的承載字符時，才能使用 Sub 和 ClientId 維度。

新的指標維度

除了現有的維度 (DatastoreId、DatastoreType、Operation) 之外，使用互通性端點時也會自動新增下列維度：

維度	Description	範例數值	來源
URIType	CMS 合規類別	patient-access , provider-access ,	從互通性端點路徑自動確定

維度	Description	範例數值	來源
		payer-to-payer , prior-authorization	
子	來電者身分	使用者/實體識別符	從承載字符sub宣告擷取
ClientId	應用程式識別符	portal_app , ehr_system	從承載字符client_id 宣告擷取

可用的指標

所有現有的 HealthLake 指標現在都包含使用互通性端點時的額外維度：

- CallCount - API 呼叫總數
- 延遲 - API 回應時間，以毫秒為單位
- UserErrors - 4xx 用戶端錯誤計數
- SystemErrors - 5xx 伺服器錯誤的計數
- 限流 - 限流請求的計數
- SuccessfulRequests - 成功 API 呼叫的計數

在 CloudWatch 中查詢指標

CloudWatch Insights 查詢範例

依應用程式查詢所有病患存取 API 呼叫：

```
SELECT SUM(CallCount)
FROM "AWS/HealthLake"
WHERE DatastoreId = '75c1cf9b0d71cd38fec8f7fb317c4c1a'
AND URIType = 'patient-access'
GROUP BY ClientId
```

的支援參考 AWS HealthLake

下列支援參考資料可供使用 AWS HealthLake。

Note

所有原生 HealthLake 動作和資料類型會在單獨的參考中描述。如需詳細資訊，請參閱 [AWS HealthLake API 參考](#)。

主題

- [AWS HealthLake 端點和配額](#)
- [HealthLake 的 Synthea 預先載入資料類型](#)
- [AWS HealthLake 範例專案](#)
- [故障診斷 AWS HealthLake](#)
- [搭配 AWS SDK 使用 HealthLake](#)

AWS HealthLake 端點和配額

下列主題包含 AWS HealthLake 服務端點和配額的相關資訊。

主題

- [服務端點](#)
- [Service Quotas](#)

服務端點

服務端點是識別主機和連接埠做為 Web 服務進入點的 URL。每個 Web 服務請求都會包含一個端點。大多數 AWS 服務為特定區域提供端點，以實現更快的連線。下表列出的服務端點 AWS HealthLake。

區域名稱	區域	端點	通訊協定
美國東部 (俄亥俄)	us-east-2	healthlake.us-east-2.amazonaws.com	HTTPS
		healthlake-fips.us-east-2.amazonaws.com	HTTPS
美國東部 (維吉尼亞 北部)	us-east-1	healthlake.us-east-1.amazonaws.com	HTTPS
		healthlake-fips.us-east-1.amazonaws.com	HTTPS

區域名稱	區域	端點	通訊協定
美國西部 (奧勒岡)	us-west-2	healthlake.us-west-2.amazonaws.com	HTTPS
		healthlake-fips.us-west-2.amazonaws.com	HTTPS
亞太區域 (孟買)	ap-south-1	healthlake.ap-south-1.amazonaws.com	HTTPS
亞太地區 (悉尼)	ap-southeast-2	healthlake.ap-southeast-2.amazonaws.com	HTTPS
加拿大 (中部)	ca-central-1	healthlake.ca-central-1.amazonaws.com	HTTPS
歐洲 (愛爾蘭)	eu-west-1	healthlake.eu-west-1.amazonaws.com	HTTPS
歐洲 (倫敦)	eu-west-2	healthlake.eu-west-2.amazonaws.com	HTTPS

Service Quotas

服務配額定義為您 AWS 帳戶中資源、動作和項目的最大值。

Note

對於可調整配額，您可以使用 [Service Quotas 主控台](#) 請求增加配額。如需詳細資訊，請參閱「Service Quotas 使用者指南」中的 [請求提高配額](#)。

同步寫入 API 速率會隨著承載大小按比例增加，每增加 1KB 會消耗額外的容量（例如，4KB 承載會使用 4 倍的寫入容量）。將選用 `x-amz-fhir-history-consistency-level` 標頭設定為 `strong` 會使每個資源的寫入容量耗用 `strong` 加倍。

套件內的資源會遵循以 1 KB 承載大小為基礎的標準讀取/寫入限制。套件類型交易使用的寫入容量是類型批次的兩倍，這表示批次套件每秒可以處理兩倍於交易套件的資源。

下表列出 的預設配額 AWS HealthLake。

名稱	預設	可調整	說明
每個帳戶的作用中訂閱數量	每個受支援的區域：100	是	每個帳戶的作用中訂閱資源數目上限。
每個資料存放區的作用中訂閱資源數量	每個受支援的區域：50	是	每個資料存放區的作用中訂閱資源數量上限。
每個帳戶的作用中 SubscriptionTopic 數量	每個受支援的區域：100	是	每個帳戶的作用中 SubscriptionTopic 資源數目上限。
每個資料存放區的作用中 SubscriptionTopic 資源數量	每個受支援的區域：50	是	每個資料存放區的作用中 SubscriptionTopic 資源數量上限。
醫學筆記中的字元數	每個受支援的區域：10,000	否	DocumentReference 資源類型 (POST/PUT 請求) 中個別醫療備註中的字元數上限。
並行 StartFHIRExportJob 任務的數量	每個受支援的區域：1	否	並行 StartFHIRExportJob 任務上限。
並行 StartFHIRImportJob 任務的數量	每個受支援的區域：1	否	並行 StartFHIRImportJob 任務上限。
每個帳戶的資料存放區數量	每個受支援的區域：20	是	每個帳戶的預設作用中資料存放區數量上限。
StartFHIRImportJob 中的檔案數量	每個受支援的區域：1,000,000	是	StartFHIRImportJob 中的檔案數目上限。
每個套件的資源數量	每個受支援的區域：500	否	套件請求中允許的資源數目上限。

名稱	預設	可調整	說明
每個帳戶使用 DELETE 的 CancelFHIRExportJob 請求率	每個受支援的區域：1	否	每個帳戶每秒可使用 DELETE 提出的 CancelFHIRExportJob 請求數目上限。
每個帳戶的 CreateFHIRDatastore 請求率	每個受支援的區域：1	否	您可以為每個帳戶每分鐘提出的 CreateFHIRDatastore 請求數目上限。
每個帳戶的 DeleteFHIRDatastore 請求率	每個受支援的區域：1	否	您可以為每個帳戶每分鐘提出的 DeleteFHIRDatastore 請求數目上限。
每個帳戶的 DescribeFHIRBulkDeleteJob 請求率	每個受支援的區域：10	<u>是</u>	每個帳戶每秒可以提出的 DescribeFHIRBulkDeleteJob 請求數目上限。
每個資料存放區的 DescribeFHIRBulkDeleteJob 請求率	每個受支援的區域：10	<u>是</u>	每個資料存放區每秒可以提出的 DescribeFHIRBulkDeleteJob 請求數目上限。
每個帳戶的 DescribeFHIRDatastore 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 DescribeFHIRDatastore 請求數目上限。
每個帳戶的 DescribeFHIRExportJob 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 DescribeFHIRExportJob 請求數目上限。

名稱	預設	可調整	說明
每個帳戶使用 GET 的 DescribeFHIRExportJob 請求速率	每個受支援的區域：10	否	每個帳戶每秒可使用 GET 提出的 DescribeFHIRExportJob 請求數目上限。
每個帳戶的 DescribeFHIRImportJob 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 DescribeFHIRImportJob 請求數目上限。
每個帳戶的探索請求率	每個受支援的區域：10	否	每個帳戶每分鐘可以提出的探索請求數目上限。
每個帳戶的 GET 請求率	每個支援的區域：6,000	<u>是</u>	您可以為每個帳戶每秒提出的 GET 請求數目上限。
每個資料存放區的 GET 請求速率	每個受支援的區域：3,000	<u>是</u>	您可以為每個資料存放區每秒提出的 GET 請求數目上限。在 8/21/2023 之前建立的資料存放區限制為每秒 100 個請求。
每個帳戶的 GetCapabilities 請求速率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 GetCapabilities 請求數目上限。
每個資料存放區的 GetExportedFile 請求率	每個受支援的區域：10	否	您可以為每個資料存放區每秒提出的 GetExportedFile 請求數目上限。
每個帳戶的 ListFHIRDatastores 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 ListFHIRDatastores 請求數目上限。

名稱	預設	可調整	說明
每個帳戶的 ListFHIRExportJobs 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 ListFHIRExportJobs 請求數目上限。
每個帳戶的 ListFHIRImportJobs 請求率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 ListFHIRImportJobs 請求數目上限。
每個帳戶的 ListTagsForResource 請求速率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 ListTagsForResource 請求數目上限。
每個帳戶的 SearchEverything 請求速率	每個受支援的區域：10	<u>是</u>	您可以為每個帳戶每秒提出的 SearchEverything 請求數目上限。
每個資料存放區的 SearchEverything 請求速率	每個受支援的區域：10	<u>是</u>	您可以為每個資料存放區每秒提出的 SearchEverything 請求數目上限。
每個帳戶的 StartFHIRBulkDeleteJob 請求率	每個受支援的區域：10	<u>是</u>	您可以為每個帳戶每秒提出的 StartFHIRBulkDeleteJob 請求數目上限。
每個資料存放區的 StartFHIRBulkDeleteJob 請求率	每個受支援的區域：10	<u>是</u>	您可以為每個資料存放區每秒提出的 StartFHIRBulkDeleteJob 請求數目上限。
每個帳戶的 StartFHIRBulkMemberMatchJob 請求率	每個受支援的區域：10	<u>是</u>	您可以為每個帳戶每秒提出的 StartFHIRBulkMemberMatchJob 請求數目上限。

名稱	預設	可調整	說明
每個資料存放區的 StartFHIR BulkMemberMatchJob 請求率	每個受支援的區域：10	<u>是</u>	您可以為每個資料存放區每秒提出的 StartFHIR BulkMemberMatchJob 請求數目上限。
每個帳戶的 StartFHIRExportJob 請求率	每個受支援的區域：1	否	您可以為每個帳戶每秒提出的 StartFHIRExportJob 請求數目上限。
每個帳戶使用 GET 的 StartFHIR ExportJob 請求速率	每個受支援的區域：1	否	每個帳戶每秒可使用 GET 提出的 StartFHIR ExportJob 請求數目上限。
每個帳戶使用 POST 的 StartFHIR ExportJob 請求率	每個受支援的區域：1	否	每個帳戶每秒可使用 POST 提出的 StartFHIR ExportJob 請求數目上限。
每個帳戶的 StartFHIRImportJob 請求率	每個受支援的區域：25	否	您可以為每個帳戶每秒提出的 StartFHIRImportJob 請求數目上限。
每個帳戶的 TagResource 請求速率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 TagResource 請求數目上限。
每個帳戶的 UntagResource 請求速率	每個受支援的區域：10	否	您可以為每個帳戶每秒提出的 UntagResource 請求數目上限。

名稱	預設	可調整	說明
每個帳戶的 ValidateResource 請求速率	每個受支援的區域：2,000	<u>是</u>	您可以為每個帳戶每秒提出的 ValidateResource 請求數目上限。在 8/21/2023 之前建立的資料存放區限制為每秒 300 個請求。
每個資料存放區的 ValidateResource 請求速率	每個受支援的區域：1,000	<u>是</u>	您可以為每個資料存放區每秒提出的 ValidateResource 請求數目上限。在 8/21/2023 之前建立的資料存放區限制為每秒 300 個請求。
每個帳戶的 WRITE 請求率	每個支援的區域：6,000	<u>是</u>	您可以為每個帳戶每秒提出的 CREATE UPDATE DELETE 請求數目上限。
每個資料存放區的 WRITE 請求速率	每個受支援的區域：3,000	<u>是</u>	您可以為每個資料存放區每秒提出的 CREATE UPDATE DELETE 請求數目上限。在 8/21/2023 之前建立的資料存放區限制為每秒 300 個請求。
每個帳戶使用 GET 的搜尋請求率	每個受支援的區域：200	<u>是</u>	每個帳戶每秒可使用 GET 提出的搜尋請求數目上限。
每個資料存放區使用 GET 的搜尋請求速率	每個受支援的區域：100	<u>是</u>	每個資料存放區每秒可使用 GET 提出的搜尋請求數目上限。

名稱	預設	可調整	說明
每個帳戶使用 POST 的搜尋請求率	每個受支援的區域：200	<u>是</u>	每個帳戶每秒可使用 POST 提出的搜尋請求數目上限。
每個資料存放區使用 POST 的搜尋請求率	每個受支援的區域：100	<u>是</u>	每個資料存放區每秒可使用 POST 提出的搜尋請求數目上限。
個別匯入檔案的大小	每個支援的區域：50 GB	否	StartFHIRImportJob 中包含之個別檔案的大小上限（以 GB 為單位）。
每個資料存放區的佇列非同步套件交易總數	每個受支援的區域：500	<u>是</u>	任何指定時間每個資料存放區的佇列非同步套件交易數目上限。
每個資料存放區的佇列大量匯出任務總數	每個受支援的區域：25	<u>是</u>	任何指定時間每個資料存放區的佇列大量匯出任務數目上限。
每個資料存放區的佇列大量匯入任務總數	每個受支援的區域：25	<u>是</u>	任何指定時間每個資料存放區的佇列大量匯入任務數目上限。
匯入任務大小總計	每個支援的區域：5,000 GB	<u>是</u>	匯入任務中包含之所有檔案的大小上限（以 GB 為單位）。

HealthLake 的 Synthea 預先載入資料類型

HealthLake 僅支援 SYNTHEA 作為預先載入的資料類型。[Synthea](#) 是一種合成患者產生器，可建立 Patient 醫療歷史記錄的模型。它託管在開放原始碼 Git 儲存庫中，該儲存庫允許 HealthLake 產生 FHIR R4-compliant 資源，Bundle 讓使用者可以在不使用實際患者資料的情況下測試模型。

下列資源類型可在預先載入的 HealthLake 資料存放區中使用。如需使用 Synthea 資料預先載入 HealthLake 資料存放區的詳細資訊，請參閱 [建立 HealthLake 資料存放區](#)。

Note

若要檢視 HealthLake 支援的 FHIR R4 資源的完整清單，請參閱 [HealthLake 的 FHIR R4 支援的資源類型](#)。

HealthLake 支援的 Synthea FHIR 資源類型

AllergyIntolerance	Location
CarePlan	MedicationAdministration
CareTeam	MedicationRequest
取得	觀察
條件	組織
裝置	病患
DiagnosticReport	從業人員
遇到	PractitionerRole
ExplanationofBenefit	程序
ImagingStudy	證明
預防	

AWS HealthLake 範例專案

若要進一步分析，您可以搭配其他 AWS 服務使用 HealthLake，如下列部落格文章範例所示。

HealthLake 整合式分析

- [人口運作狀態應用程式 AWS HealthLake – 第 1 部分：使用 Amazon Quick 進行分析和監控](#)。
- [使用具有 AWS HealthLake 標準化資料的 Amazon SageMaker AI 建置預測性疾病模型](#)。

- [使用 AWS AI 服務建立認知搜尋和運作狀態知識圖表。](#)

HealthLake 事件監控

- [Amazon EventBridge 與 整合 AWS HealthLake。](#)

故障診斷 AWS HealthLake

下列主題提供使用、AWS CLI AWS SDKs 或 HealthLake 主控台時可能遇到的錯誤和問題的疑難排解建議。如果您發現本節未列出的問題，請使用此頁面右側的提供意見回饋按鈕進行報告。

主題

- [資料存放區動作](#)
- [匯入動作](#)
- [FHIR APIs](#)
- [NLP 整合](#)
- [SQL 整合](#)

資料存放區動作

問題：當我嘗試建立 HealthLake 資料存放區時，會收到下列錯誤：

```
AccessDeniedException: Insufficient Lake Formation permission(s): Required Database on Catalog
```

2022 年 11 月 14 日，HealthLake 更新了建立新資料存放區所需的 IAM 許可。如需詳細資訊，請參閱[設定 IAM 使用者或角色以使用 HealthLake \(IAM 管理員\)](#)。

問題：使用 AWS SDKs 建立 HealthLake 資料存放區時，資料存放區建立狀態會傳回例外狀況或未知狀態。

如果您的 DescribeFHIRDatastore 或 ListFHIRDatastores API 呼叫傳回例外狀況或未知的資料存放區狀態，請將 SDK 更新 AWS 為最新版本。

匯入動作

問題：如果我的資料不是 FHIR R4 格式，我是否仍然可以使用 HealthLake？

只有 FHIR R4 格式的資料可以匯入 HealthLake 資料存放區。如需可協助將現有運作狀態資料轉換為 FHIR R4 格式的合作夥伴清單，請參閱 [AWS HealthLake 合作夥伴](#)。

問題：為什麼我的 FHIR 匯入任務失敗？

成功的匯入任務將產生具有 .ndjson 格式結果（輸出日誌）的資料夾，不過，個別記錄可能無法匯入。發生這種情況時，會產生第二個 FAILURE 資料夾，其中包含無法匯入的記錄清單。如需詳細資訊，請參閱 [使用 匯入 FHIR 資料 AWS HealthLake](#)。

若要分析匯入任務失敗的原因，請使用 DescribeFHIRImportJob API 來分析 JobProperties。建議使用下列項目：

- 如果狀態為 FAILED 且訊息存在，則失敗與任務參數有關，例如輸入資料大小或輸入檔案的數量超出 HealthLake 配額。
- 如果匯入任務狀態為 COMPLETED_WITH_ERRORS，請檢查資訊清單檔案 manifest.json，以取得哪些檔案未成功匯入的資訊。
- 如果匯入任務狀態為 FAILED 且訊息不存在，請前往任務輸出位置以存取資訊清單檔案 manifest.json。

對於每個輸入檔案，都有失敗輸出檔案，其中包含任何無法匯入之資源的輸入檔案名稱。回應包含對應於輸入資料位置的行號 (lineId)、FHIR 回應物件 (UpdateResourceResponse) 和回應的狀態碼 (statusCode)。

範例輸出檔案可能類似下列內容：

```
{"lineId":3, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"1 validation error detected:
Value 'Patient123' at 'resourceType' failed to satisfy constraint: Member must satisfy
regular expression pattern: [A-Za-z]{1,256}"}]}, "statusCode":400}
{"lineId":5, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"This property must be an
simple value, not a com.google.gson.JsonArray","location":["/EffectEvidenceSynthesis/
name"]}, {"severity":"error","code":"processing","diagnostics":"Unrecognised
property '@telecom',"location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
property '@gender',"location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
property '@birthDate',"location":["/EffectEvidenceSynthesis"]},
{"severity":"error","code":"processing","diagnostics":"Unrecognised
```

```

    property '@address',"location":["/EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Unrecognised
  property '@maritalStatus',"location":["/EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Unrecognised
  property '@multipleBirthBoolean',"location":["/EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Unrecognised
  property '@communication',"location":["/EffectEvidenceSynthesis"]}],
{"severity":"warning","code":"processing","diagnostics":"Name should be usable as an
  identifier for the module by machine processing applications such as code generation
  [name.matches('[A-Z]([A-Za-z0-9_]){0,254}')]","location":["EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
  StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.status':
  minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Profile
  http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
  Element 'EffectEvidenceSynthesis.population': minimum required
  = 1, but only found 0","location":["EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Profile
  http://hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis,
  Element 'EffectEvidenceSynthesis.exposure': minimum required =
  1, but only found 0","location":["EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Profile http://
  hl7.org/fhir/StructureDefinition/EffectEvidenceSynthesis, Element
  'EffectEvidenceSynthesis.exposureAlternative': minimum required
  = 1, but only found 0","location":["EffectEvidenceSynthesis"]}],
{"severity":"error","code":"processing","diagnostics":"Profile http://hl7.org/fhir/
  StructureDefinition/EffectEvidenceSynthesis, Element 'EffectEvidenceSynthesis.outcome':
  minimum required = 1, but only found 0","location":["EffectEvidenceSynthesis"]}],
{"severity":"information","code":"processing","diagnostics":"Unknown
  extension http://synthetichealth.github.io/synthea/disability-adjusted-
  life-years","location":["EffectEvidenceSynthesis.extension[3]"]}],
{"severity":"information","code":"processing","diagnostics":"Unknown extension
  http://synthetichealth.github.io/synthea/quality-adjusted-life-years","location":
  ["EffectEvidenceSynthesis.extension[4]"]}]}, "statusCode":400}
{"lineId":7, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"2 validation errors detected:
  Value at 'resourceId' failed to satisfy constraint: Member must satisfy regular
  expression pattern: [A-Za-z0-9-.]{1,64}; Value at 'resourceId' failed to satisfy
  constraint: Member must have length greater than or equal to 1"}]}}, "statusCode":400}
{"lineId":9, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Missing required id field in
  resource json"}]}}, "statusCode":400}

```

```
{"lineId":15, UpdateResourceResponse:{"jsonBlob":
{"resourceType":"OperationOutcome","issue":
[{"severity":"error","code":"processing","diagnostics":"Invalid JSON found in input
file"}]}, "statusCode":400}
```

上述範例顯示輸入檔案中對應輸入行的第 3、4、7、9、15 行失敗。對於每行，說明如下：

- 在第 3 行，回應說明輸入檔案第 3 行 resourceType 所提供的是無效的。
- 在第 5 行，回應說明輸入檔案的第 5 行中存在 FHIR 驗證錯誤。
- 在第 7 行，回應說明 resourceId 提供做為輸入時發生驗證問題。
- 在第 9 行，回應說明輸入檔案必須包含有效的資源 ID。
- 在第 15 行，輸入檔案的回應是檔案不是有效的 JSON 格式。

FHIR APIs

問題：如何實作 FHIR RESTful APIs 的授權？

決定 [資料存放區授權策略](#) 要使用的。

若要使用 建立 SigV4 授權 適用於 Python (Boto3) 的 AWS SDK，請建立類似下列範例的指令碼。

```
import boto3
import requests
import json
from requests_auth_aws_sigv4 import AWSSigV4

# Set the input arguments
data_store_endpoint = 'https://healthlake.us-east-1.amazonaws.com/datastore/<datastore
id>/r4/'
resource_path = "Patient"
requestBody = {"resourceType": "Patient", "active": True, "name": [{"use":
"official", "family": "Dow", "given": ["Jen"]}, {"use": "usual", "given":
["Jen"]}], "gender": "female", "birthDate": "1966-09-01"}
region = 'us-east-1'

#Frame the resource endpoint
resource_endpoint = data_store_endpoint+resource_path
session = boto3.session.Session(region_name=region)
client = session.client("healthlake")

# Frame authorization
```

```
auth = AWSSigV4("healthlake", session=session)

# Call data store FHIR endpoint using SigV4 auth

r = requests.post(resource_endpoint, json=requestBody, auth=auth, )
print(r.json())
```

問題：為什麼針對使用客戶受管 KMS 金鑰加密的資料存放區使用 FHIR RESTful APIs 時收到 *AccessDenied* 錯誤？

使用者或角色需要客戶受管金鑰和 IAM 政策的許可，才能存取資料存放區。使用者必須擁有使用客戶受管金鑰所需的 IAM 許可。如果使用者撤銷或淘汰授予 HealthLake 使用客戶受管 KMS 金鑰的許可，HealthLake 將傳回 *AccessDenied* 錯誤。

HealthLake 必須具備存取客戶資料的許可、加密匯入資料存放區的新 FHIR 資源，以及在請求時解密 FHIR 資源。如需詳細資訊，請參閱 [故障診斷 AWS KMS 許可](#)。

問題：使用 10MB 文件的 FHIR *POST* API 操作向 HealthLake 傳回 *413 Request Entity Too Large* 錯誤。

AWS HealthLake 具有 5MB 的同步建立和更新 API 限制，以避免增加延遲和逾時。您可以使用大量匯入 API，使用 Binary 資源類型擷取最多 164MB 的大型文件。

NLP 整合

問題：如何開啟 HealthLake 整合的自然語言處理功能？

截至 2022 年 11 月 14 日，HealthLake 資料存放區的預設行為已變更。

目前的資料存放區：所有目前的 HealthLake 資料存放區都將停止使用 base64 編碼 DocumentReference 資源上的自然語言處理 (NLP)。這表示將不會使用 NLP 分析新 DocumentReference 資源，也不會根據資源類型中的文字產生新 DocumentReference 資源。對於現有 DocumentReference 資源，透過 NLP 產生的資料和資源仍會保留，但不會在 2023 年 2 月 20 日之後更新。

新資料存放區：2023 年 2 月 20 日之後建立的 HealthLake 資料存放區將不會對 base64 編碼 DocumentReference 的資源執行自然語言處理 (NLP)。

若要開啟 HealthLake NLP 整合，請使用 [建立支援案例 AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。若要進一步了解如何建立案例和案例管理，請參閱支援《使用者指南》中的 [建立支援案例和案例管理](#)。

問題：>如何尋找整合式 NLP 無法處理`DocumentReference`的資源？

如果`DocumentReference`資源無效，HealthLake 會提供表示驗證錯誤的延伸，而不是在整合的醫療 NLP 輸出中提供。若要尋找在 NLP 處理期間導致驗證錯誤`DocumentReference`的資源，您可以使用 HealthLake 的 FHIR search 函數搭配搜尋索引鍵 `cm-decoration-status` 和搜尋值 `VALIDATION_ERROR`。此搜尋會列出導致驗證錯誤的所有`DocumentReference`資源，以及描述錯誤性質的錯誤訊息。這些`DocumentReference`資源中具有驗證錯誤的延伸欄位結構將類似下列範例。

```
"extension": [
  {
    "extension": [
      {
        "url": "http://healthlake.amazonaws.com/aws-cm/status/",
        "valueString": "VALIDATION_ERROR"
      },
      {
        "url": "http://healthlake.amazonaws.com/aws-cm/message/",
        "valueString": "Resource led to too many nested objects after NLP
operation processed the document. 10937 nested objects exceeds the limit of 10000."
      }
    ],
    "url": "http://healthlake.amazonaws.com/aws-cm/"
  }
]
```

Note

如果 NLP 裝飾建立超過 10,000 個巢狀物件，`VALIDATION_ERROR`也可能發生。發生這種情況時，文件必須在處理之前分割成較小的文件。

SQL 整合

問題：新增資料湖管理員`permissions error: lakeformation:PutDataLakeSettings`時，為什麼要取得 Lake Formation？

如果您的 IAM 使用者或角色包含 `AWSLakeFormationDataAdmin` AWS 受管政策，則無法新增資料湖管理員。您會收到包含下列內容的錯誤：

```
User arn:aws:sts::111122223333:assumed-role/lakeformation-admin-user is not authorized to perform: lakeformation:PutDataLakeSettings on resource: arn:aws:lakeformation:us-east-2:111122223333:catalog:111122223333 with an explicit deny in an identity-based policy
```

AdministratorAccess 需要 AWS 受管政策，才能將 IAM 使用者或角色新增為 AWS Lake Formation 資料湖管理員。如果您的 IAM 使用者或角色也包含 AWSLakeFormationDataAdmin 動作將會失敗。AWSLakeFormationDataAdmin AWS 受管政策包含 AWS Lake Formation API 操作的明確拒絕PutDataLakeSetting。即使是具有 AWS 使用 AdministratorAccess受管政策之完整存取權的管理員，也可以受到AWSLakeFormationDataAdmin政策的限制。

問題：如何遷移現有的 HealthLake 資料存放區以使用 Amazon Athena SQL 整合？

在 2022 年 11 月 14 日之前建立的 HealthLake 資料存放區可正常運作，但無法使用 SQL 在 Athena 中查詢。若要使用 Athena 查詢預先存在的資料存放區，您必須先將其遷移至新的資料存放區。

將 HealthLake 資料遷移至新的資料存放區

1. 建立新的資料存放區。
2. 從預先存在的資料匯出至 Amazon S3 儲存貯體。
3. 從 Amazon S3 儲存貯體將資料匯入新的資料存放區。

Note

將資料匯出至 Amazon S3 儲存貯體會產生額外費用。額外費用取決於您匯出的資料大小。

問題：為 SQL 整合建立新的 HealthLake 資料存放區時，資料存放區狀態不會從 變更*Creating*。

如果您嘗試建立新的 HealthLake 資料存放區，且資料存放區狀態並未從建立中變更，則需要更新 Athena 才能使用 AWS Glue Data Catalog。如需詳細資訊，請參閱《Amazon Athena [AWS 使用者指南](#)》中的[step-by-step升級至 Glue Data Catalog](#)。Amazon Athena

成功升級 後 AWS Glue Data Catalog，您可以建立 HealthLake 資料存放區。

若要移除舊的 HealthLake 資料存放區，請使用 建立支援案例[AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。若要進一步了解，請參閱支援《使用者指南》中的[建立支援案例和案例管理](#)。

問題：Athena 主控台在將資料匯入新的 HealthLake 資料存放區後無法運作

將資料匯入新的 HealthLake 資料存放區後，資料可能無法立即使用。這是為了讓資料有時間擷取到 Apache Iceberg 資料表。請稍後重試。

問題：如何將 Athena 中的搜尋結果連接到其他 AWS 服務？

當您將 Athena 的搜尋結果與其他 AWS 服務共用時，當您使用 `json_extract[1]` 做為 SQL 搜尋查詢的一部分時，可能會發生問題。若要修正此問題，您必須更新為 `CATVAR`。

嘗試建立儲存結果、資料表（靜態）或檢視（動態）時，您可能會遇到此問題。

搭配 AWS SDK 使用 HealthLake

AWS 軟體開發套件 (SDKs) 適用於許多熱門的程式設計語言。每個 SDK 都提供 API、程式碼範例和說明文件，讓開發人員能夠更輕鬆地以偏好的語言建置應用程式。

SDK 文件	代碼範例
適用於 C++ 的 AWS SDK	適用於 C++ 的 AWS SDK 程式碼範例
AWS CLI	AWS CLI 程式碼範例
適用於 Go 的 AWS SDK	適用於 Go 的 AWS SDK 程式碼範例
適用於 Java 的 AWS SDK	適用於 Java 的 AWS SDK 程式碼範例
適用於 JavaScript 的 AWS SDK	適用於 JavaScript 的 AWS SDK 程式碼範例
適用於 Kotlin 的 AWS SDK	適用於 Kotlin 的 AWS SDK 程式碼範例
適用於 .NET 的 AWS SDK	適用於 .NET 的 AWS SDK 程式碼範例
適用於 PHP 的 AWS SDK	適用於 PHP 的 AWS SDK 程式碼範例
AWS Tools for PowerShell	AWS Tools for PowerShell 程式碼範例
適用於 Python (Boto3) 的 AWS SDK	適用於 Python (Boto3) 的 AWS SDK 程式碼範例
適用於 Ruby 的 AWS SDK	適用於 Ruby 的 AWS SDK 程式碼範例

SDK 文件	代碼範例
適用於 Rust 的 AWS SDK	適用於 Rust 的 AWS SDK 程式碼範例
適用於 SAP ABAP 的 AWS SDK	適用於 SAP ABAP 的 AWS SDK 程式碼範例
適用於 Swift 的 AWS SDK	適用於 Swift 的 AWS SDK 程式碼範例

可用性範例

找不到所需的內容嗎？請使用本頁面底部的提供意見回饋連結申請程式碼範例。

AWS HealthLake 版本

下表顯示發佈功能和更新的時機 AWS HealthLake。如需版本的詳細資訊，請參閱連結主題。

變更	描述	日期
\$export 和 \$davinci-data-export 的安全性和標籤篩選	\$export 和 \$davinci-data-export 操作現在支援 <code>_security</code> 和 <code>_tag</code> 查詢參數，以依 <code>meta.security</code> 和 <code>meta.tag</code> 編碼值篩選匯出的資源。這可使用標準 FHIR <code>system code</code> 格式啟用多租戶匯出和精細存取控制。	2026 年 4 月 30 日
\$bulk-member-match 操作	<p>AWS HealthLake 現在支援以非同步方式處理多個成員比對請求 <code>\$bulk-member-match</code> 的操作。此操作可讓醫療保健組織在單一大量請求中使用人口統計和涵蓋範圍資訊，在不同的醫療保健系統中有效率地比對數百個成員的唯一識別符。</p> <ul style="list-style-type: none"> • 每個請求最多可處理 500 個成員，每個資料存放區最多可處理 5 個並行操作 • 分類為 <code>MatchedMembers</code>、<code>NonMatchedMembers</code> 和 <code>ConsentConstrainedMembers</code> 群組的結果 	2026 年 4 月 1 日

- 整合與 `$davinci-data-export` 以進行端對端大量資料工作流程 end-to-end

如需詳細資訊，請參閱[the section called “\\$bulk-member-match”](#)。

[非同步套件交易](#)

AWS HealthLake 現在支援非同步 Bundle 類型 transaction，可讓您提交具有最多 500 個資源的交易。HealthLake 會將交易排入佇列進行處理，並立即傳回輪詢 URL 以檢查狀態並擷取結果。如需詳細資訊，請參閱[非同步套件交易](#)。

2026 年 3 月 24 日

[套件操作中的修補程式支援](#)

HealthLake 現在支援[套件修補程式](#)，同時針對交易和批次套件啟用 FHIRPath 和 JSON 修補程式，以及針對修補程式 API 啟用 FHIRPath。此功能可讓客戶直接在 Bundle 操作中修補資源，而不需要完整資源替換、減少承載大小、簡化整合工作流程，以及啟用更快速的資料擷取。

2026 年 3 月 20 日

\$davinci-data-export 的 DaVinci PDex 匯出類型 davinci-data-export	<p>\$davinci-data-export 操作現在支援提供者存取、Payer-to-Payer和成員存取 APIs PDex 匯出類型。</p> <ul style="list-style-type: none"> • ExplanationOfBenefit 資源的設定檔型包含邏輯 • 使用 <code>_includeE OB2xWoFinancial</code> 參數進行財務資料轉換 • 臨床和宣告資料的 5 年時間篩選條件 	2026 年 3 月 20 日
\$export 和 \$davinci-data-export 中的 <code>_until</code> 參數	匯出操作的時間篩選參數	2026 年 2 月 26 日
_include 搜尋參數	HealthLake 現在支援包含： <code>*</code> 和包含： <code>iterate</code>	2026 年 2 月 26 日
CMS 互通性端點	此功能可讓您依 CMS 類別追蹤 API 用量，然後基於合規目的報告用量指標。	2026 年 2 月 26 日
套件訊息類型支援	對具有訊息類型的 FHIR 套件資源的有限支援	2026 年 2 月 26 日

新增對新 IGs 支援

AWS HealthLake 已擴展其對 CMS 0057F 的 FHIR 實作指南 (IGs) 支援：

2026 年 2 月 26 日

- 支援 CARIN 藍色按鈕 2.0.0 和 2.1.0
- 支援 Da Vinci Payer Data Exchange 2.0.0 和 2.1.0
- 支援 Da Vinci Payer Data Exchange (PDex) 美國藥物配方 2.0.1 和 2.1.0
- 支援 Da Vinci 臨床資料交換 (CDex) 2.1.0
- 支援 Da Vinci 預先授權支援 (PAS) FHIR IG 2.1.0

\$submit 操作

\$submit 此操作可讓您以電子方式提交預先授權請求給付款人以進行核准。

2026 年 2 月 26 日

\$questionnaire-package 操作

\$questionnaire-package 操作會擷取完整的套件，其中包含 FHIR 問卷及其轉譯和處理問卷所需的所有相依性。

2026 年 2 月 26 日

\$inquire 操作

\$inquire 操作可讓您檢查先前提交的預先授權請求的狀態。

2026 年 2 月 26 日

\$member-remove 操作

\$member-remove 操作可讓您從 FHIR 成員屬性清單（群組資源）中移除成員 AWS HealthLake。

2025 年 11 月 12 日

\$member-match 操作	AWS HealthLake 現在支援病患資源的 \$member-match 操作，讓醫療保健組織能夠使用人口統計和涵蓋範圍資訊，跨不同的醫療保健系統尋找成員的唯一識別符。	2025 年 11 月 12 日
\$member-add 操作	FHIR \$member-add 操作會將成員（病患）新增至群組資源，特別是成員屬性清單。	2025 年 11 月 12 日
\$davinci-data-export 操作	\$davinci-data-export 操作是一種非同步 FHIR 操作，可啟用從中匯出成員屬性清單資料 AWS HealthLake。	2025 年 11 月 12 日
\$confirm-attribution-list 操作	向生產者指出消費者不再需要對屬性清單進行變更，透過移除非作用中成員並將狀態變更為「最終」來完成屬性清單。	2025 年 11 月 12 日
\$attribution-status 操作	擷取特定成員的屬性狀態，傳回包含與病患相關所有屬性資源的套件。	2025 年 11 月 12 日
FHIR 訂閱	HealthLake 支援 FHIR 訂閱，可讓您在特定醫療保健資料變更發生時接收即時通知，並建置事件驅動型工作流程。	2025 年 10 月 30 日
區域擴展到蒙特婁加拿大	HealthLake 可在加拿大（蒙特婁）區域使用。如需詳細資訊，請參閱 服務端點 。	2025 年 10 月 17 日

[新增對新 IGs 支援](#)

AWS HealthLake 已擴展其 FHIR 實作指南 (IGs) 對下列加拿大市場的支援：

2025 年 10 月 17 日

- CA Core+Canadian 基準 FHIR 描述檔定義核心資料元素和限制，以跨加拿大醫療保健系統進行互通性。
- CA : eReC Pan-Canadian eReferral-eConsultStandardized FHIR 規格，適用於加拿大各地醫療保健供應商之間的電子推薦和諮詢。
- 病患摘要 加拿大版 (PS-CA) 加拿大國際病患摘要 (IPS) 的調整，用於跨照護設定共用重要的病患健康資訊。
- Ontario Digital Health Drug Repository Ontario 特定的 FHIR 設定檔，用於在省級醫療系統中進行標準化藥物和處方資料交換。

[區域特定的 IG 支援](#)

HealthLake 現在支援區域特定的 IGs。如需詳細資訊，請參閱[設定檔驗證](#)。

2025 年 10 月 8 日

[修補程式操作](#)

HealthLake 允許使用 JSON 修補程式操作修改 FHIR 資源的特定元素，而無需更新整個資源。

2025 年 8 月 18 日

[\\$驗證操作](#)

HealthLake 可針對規格和設定檔驗證 FHIR 資源，而無需執行儲存操作，並傳回詳細的驗證結果。

2025 年 8 月 18 日

\$purge 操作	HealthLake 包含從資料存放區永久刪除病患隔室內所有資源的功能。	2025 年 8 月 18 日
\$lookup 操作	HealthLake 提供程式碼和系統識別符，可讓您擷取 CodeSystem 中特定概念的詳細資訊。	2025 年 8 月 18 日
\$expand 操作	HealthLake 現在允許擴展 ValueSet 資源，以擷取客戶擷取的 ValueSets 中包含的代碼完整清單。	2025 年 8 月 18 日
\$erase 操作	HealthLake 現在提供從資料存放區永久刪除特定資源及其所有歷史版本。	2025 年 8 月 18 日
\$document 操作	HealthLake 透過將合成資源與其所有參考的資源綁定到單一文件套件中，支援產生完整的臨床文件。	2025 年 8 月 18 日
: 搜尋修改器下方	HealthLake 引入搜尋階層低於術語系統中指定 URI 的 URI 值。	2025 年 8 月 8 日
條件式刪除	HealthLake 現在支援 FHIR 條件式刪除，允許醫療保健組織根據搜尋條件而非邏輯 FHIR ID 刪除現有資源。如需詳細資訊，請參閱 根據條件刪除 FHIR 資源 。	2025 年 7 月 7 日

新增對新 IGs 支援

AWS HealthLake 已擴展其 FHIR 實作指南 (IGs) 對下列項目的支援：

2025 年 7 月 7 日

- US Core 7.0.0，指定如何使用 FHIR 實作 USCDI 4.0 標準
- UK Core 2.0.1 實作指南，提供全英國 FHIR 實作指引

區域擴展到愛爾蘭都柏林

HealthLake 可在歐洲（都柏林）區域使用。如需詳細資訊，請參閱[服務端點](#)。

2025 年 6 月 9 日

套件類型交易

HealthLake 現在支援 FHIR Bundle 類型「交易」，可讓醫療保健組織提交多個資源做為單一原子操作。這可讓資料交換和整合工作流程更有效率。例如，醫療保健供應商現在可以在單一交易中更新患者記錄、藥物清單和預約，從而降低複雜性和潛在的錯誤。如需詳細資訊，請參閱[綁定 FHIR 資源](#)。

2025 年 4 月 28 日

新增對新 IGs 支援

AWS HealthLake AWS HealthLake 已擴展其 FHIR 實作指南 (IGs) 對下列項目的支援：

2025 年 4 月 28 日

- NCQA HEDIS® 實作指南 (0.3.1)：支援醫療有效性資料和資訊集 (HEDIS) 的品質測量和報告。
- 國際病患摘要 (IPS) (2.0.0)：啟用基本健康資訊的交換，以支援病患持續照護。
- 品質指標 (5.0.0)：支援品質指標定義和資料的表示和交換。
- 基因體報告 (3.0.0)：促進基因體資料和報告的交換。

等冪性金鑰

HealthLake 現在支援 FHIR POST 操作的等冪性金鑰，提供強大的機制，以確保資源建立期間的資料完整性。如需詳細資訊，請參閱[冪等和並行](#)。

2025 年 4 月 18 日

FHIR 歷史記錄一致性

HealthLake 現在支援透過新 `x-amz-fhir-history-consistency-level` 標頭為已啟用[歷史記錄](#)的資料存放區提供強大的一致性。設為「強」時，FHIR 搜尋結果會包含所有索引記錄，無論更新狀態為何。如需詳細資訊，請參閱[FHIR 搜尋一致性層級](#)。

2025 年 4 月 18 日

[Etag 和 'if-match'](#)

HealthLake 現在提供 eTag 支援，允許用戶端使用「If-Match」標頭，以確保等冪更新。這有助於透過防止在並行更新期間意外覆寫來維護資料完整性。這在大量醫療保健環境中特別有用，其中多個系統可能會嘗試同時更新相同的記錄。如需詳細資訊，請參閱 [AWS HealthLake 中的 ETag](#)。

2025 年 4 月 18 日

[套件中的條件式 PUTs](#)

HealthLake 現在支援 FHIR 套件的條件式更新，讓醫療保健組織更靈活地管理和更新其資料。用戶端現在可以指定條件，在 Bundle 交易中有條件地建立、更新或刪除資源。這可簡化系統之間的資料同步程序，並減少對複雜用戶端邏輯的需求。如需詳細資訊，請參閱 [Bundles 中的條件式 PUTs](#)。

2025 年 4 月 18 日

[FHIR V2 範圍上的 SMART](#)

HealthLake 支援在 FHIR V2 範圍上建立、讀取、更新、刪除和搜尋 FHIR 資源的 SMART。如需詳細資訊，請參閱 [HealthLake 的 FHIR 資源範圍上的 SMART](#)。

2025 年 1 月 22 日

- 01/22/2025 之後建立的所有 HealthLake 資料存放區都可以使用 FHIR V2 範圍上的 SMART。如果您的資料存放區在此日期之前建立，您可以提交支援票證，以在 FHIR V2 範圍上啟用 SMART。使用 [建立案例AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。

[FHIR 美國核心設定檔，6.1.0 版](#)

HealthLake 支援 FHIR US Core Profile 的 6.1.0 版。如需詳細資訊，請參閱 [HealthLake 的 FHIR 設定檔驗證](#)。

2025 年 1 月 22 日

[FHIR \\$export 與 GET](#)

HealthLake 透過 \$export 支援 FHIRGET。如需詳細資訊，請參閱[使用 FHIR 匯出 HealthLake 資料\\$export](#)。

2025 年 1 月 22 日

[重構開發人員指南與經過測試的程式碼範例](#)

HealthLake 推出重構開發人員指南，其中包含原生 AWS CLI 和 AWS SDK 動作的測試程式碼範例。此外，程序現在適用於所有支援的 FHIR API 互動。如需詳細資訊，請參閱[程式碼範例](#)和管理 [FHIR 資源](#)。

2024 年 12 月 18 日

[FHIR history和vread互動](#)

2024 年 10 月 25 日

HealthLake 支援擷取特定資源歷史記錄的 FHIR history 互動，以及執行特定版本資源讀取 vread 的互動。如需詳細資訊，請參閱[讀取 FHIR 資源歷史記錄](#)。

- FHIR 資源預設為 history 在 10/25/2024 之後建立的所有 HealthLake 資料存放區啟用。如果您的資料存放區在此日期之前建立，您可以提交支援票證以啟用 FHIR history 互動。使用 建立案例[AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。

[FHIR Patient/\\$everything](#) 操作

HealthLake 支援 FHIR Patient/\$everything 操作來搜尋Patient資源及其所有相關資源。使用此操作，您可以存取病患的整個記錄或大量下載Patient資料。如需詳細資訊，請參閱[使用 取得患者資料Patient/\\$everything](#)。

2024 年 2 月 27 日

- FHIR Patient/\$everything 預設為在 02/27/2024 之後建立的所有 HealthLake 資料存放區。如果您的資料存放區在此日期之前建立，您可以提交支援票證以啟用 Patient/\$everything 操作。使用 [建立案例AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。

[FHIR VerificationResult](#) 資源

HealthLake 支援 FHIR VerificationResult 資源類型，以描述一或多個元素的驗證需求、來源、狀態和日期。如需詳細資訊，請參閱[HealthLake 的 FHIR R4 資源類型](#)。

2023 年 12 月 9 日

[FHIR \\$export操作](#)

HealthLake 支援從 HealthLake 資料存放區大量匯出運作狀態資料的 FHIR \$export 操作。如需詳細資訊，請參閱 [使用 FHIR 匯出 HealthLake 資料 \\$export](#)。

2023 年 6 月 1 日

- FHIR \$export 預設為啟用於 2023 年 6 月 1 日之後建立的所有 HealthLake 資料存放區。如果您的資料存放區在此日期之前建立，您可以提交支援票證以啟用 \$export 操作。使用 [建立案例 AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。
- 在 06/01/23 之前建立的 HealthLake 資料存放區僅支援整個系統匯出 \$export 的任務請求。
- 在 06/01/23 之前建立的 HealthLake 資料存放區不支援 \$export 使用資料存放區端點上的 GET 請求來取得 FHIR 狀態。

[FHIR 支援的 SMART](#)

HealthLake 新增對 FHIR 授權上的 SMART 的支援。如需詳細資訊，請參閱 [的 FHIR 支援上的 SMART AWS HealthLake](#)。

2023 年 5 月 31 日

[FHIR 設定檔驗證](#)

HealthLake 支援 FHIR 設定檔驗證，以使用基本資源類型的限制條件和/或延伸來定義特定資源類型定義。如需詳細資訊，請參閱[設定檔驗證](#)。

2023 年 5 月 31 日

[亞太區域（孟買）區域](#)

HealthLake 可在亞太區域（孟買）區域使用。如需詳細資訊，請參閱[服務端點](#)。

2023 年 4 月 4 日

[預設關閉自然語言處理](#)

HealthLake 已於 2023 年 2 月 20 日關閉所有資料存放區的整合自然語言處理 (NLP)。您可以提交支援票證，以開啟整合的 NLP 功能。使用建立案例[AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。若要進一步了解整合式 NLP，請參閱[將 NLP 與 HealthLake 整合](#)。

-

2023 年 2 月 20 日

[Amazon Athena 的 SQL 索引和查詢](#)

HealthLake 支援使用 Amazon Athena 使用 SQL 查詢 FHIR 資料。如需詳細資訊，請參閱[使用 Amazon Athena 查詢 HealthLake 資料](#)。

2022 年 11 月 14 日

- SQL 查詢功能預設為在 11/14/2022 之後建立的所有 HealthLake 資料存放區啟用。如果您的資料存放區在此日期之前建立，您可以提交支援票證以啟用 SQL 查詢功能。使用 [建立案例AWS Support Center Console](#)。若要建立您的案例，請登入您的 AWS 帳戶，然後選擇建立案例。
- 使用 SQL 查詢功能時，必須更新存取 HealthLake 的 IAM 設定。若要在 Athena 中建立 HealthLake 資料存放區並授予存取權，您必須將 AWSLakeFormationDataAdmin 受管政策新增至 IAM 使用者、群組或角色。您可以使用 AWSLakeFormationDataAdmin 政策來建立資料湖管理員，並授予 Athena 中資料存放區的存取權。如需詳細資訊，請參閱[設定 IAM 使用者或角色](#)。

總匯入任務大小增加	HealthLake 會將StartFHIR ImportJob 請求Total import job size的更新為500 GB。如需詳細資訊，請參閱 Service Quotas 。	2022 年 10 月 3 日
FHIR Bundle 資源	HealthLake 支援 FHIR Bundle 資源類型，可同時處理多個 FHIR 資源。如需詳細資訊，請參閱 綁定 FHIR 資源 。	2022 年 8 月 5 日
FHIR 互動的配額更新	HealthLake 會更新 FHIR 資源管理互動的配額。如需詳細資訊，請參閱 Service Quotas 。	2022 年 7 月 16 日
FHIR _include搜尋參數	HealthLake 新增對 FHIR _include搜尋參數的支援，以傳回search請求中的其他資源。如需詳細資訊，請參閱 進階搜尋參數 。	2022 年 7 月 16 日
AWS HealthLake 已全面推出	HealthLake 通常在所有支援的區域提供。如需詳細資訊，請參閱 服務端點 。	2021 年 7 月 15 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。