



# 核心**REST**實作

## Astra Automation

NetApp  
December 01, 2023

# 目錄

核心REST實作 .....	1
REST Web服務 .....	1
資源與收藏 .....	1
HTTP詳細資料 .....	3
URL格式 .....	5

# 核心REST實作

## REST Web服務

代表性狀態傳輸（REST）是建立分散式Web應用程式的風格。當套用到網路服務API的設計時、它會建立一套主流技術和最佳實務做法、以揭露伺服器型資源並管理其狀態。REST為應用程式開發提供一致的基礎、但每個API的詳細資料可能會因特定設計選項而異。在使用Astra Control REST API進行即時部署之前、您應該先瞭解其特性。

### 資源和狀態表示

資源是網路型系統的基本元件。建立REST Web服務應用程式時、早期的設計工作包括：

- 識別系統或伺服器型資源

每個系統都會使用及維護資源。資源可以是檔案、商業交易、程序或管理實體。根據REST Web服務設計應用程式的首要任務之一、就是識別資源。

- 資源狀態和相關狀態作業的定義

資源永遠處於有限的狀態之一。必須清楚定義狀態、以及用來影響狀態變更的相關作業。

### URI端點

每個REST資源都必須使用明確定義的定址方案來定義和提供。資源所在及識別的端點使用統一資源識別元（URI）。URI提供一般架構、可為網路中的每個資源建立唯一名稱。統一資源定位器（URL）是一種與Web服務搭配使用的URI、用於識別及存取資源。資源通常會以階層式結構公開、類似檔案目錄。

### HTTP 訊息

超文字傳輸傳輸協定（HTTP）是Web服務用戶端和伺服器用來交換有關資源的要求和回應訊息的傳輸協定。在設計Web服務應用程式時、HTTP方法會對應至資源及對應的狀態管理動作。HTTP為無狀態。因此、若要將一組相關的要求和回應建立關聯、以做為一筆交易的一部分、則必須在隨要求和回應資料流一起提供的HTTP標頭中加入額外資訊。

### JSON格式化

雖然資訊可透過多種方式在Web服務用戶端和伺服器之間進行結構化和傳輸、但最受歡迎的選項是JavaScript物件標記法（Json）。Json是以純文字表示簡單資料結構的產業標準、用於傳輸描述資源的狀態資訊。Astra Control REST API使用Json格式化在每個HTTP要求和回應的本文中所攜帶的資料。

## 資源與收藏

Astra Control REST API可讓您存取資源執行個體和資源執行個體集合。



從概念上來說、REST 資源\*類似於以物件導向程式設計（**OOP**）語言和系統所定義的\*物件。有時這些術語會互換使用。但一般而言、在外部REST API的內容中使用「資源」是偏好的、而「物件」則是用於儲存在伺服器上的對應狀態執行個體資料。

## Astra資源的屬性

Astra Control REST API符合RESTful設計原則。每個Astra資源執行個體都是根據明確定義的資源類型來建立。一組相同類型的資源執行個體稱為\*集合\*。API呼叫會對個別資源或資源集合起作用。

### 資源類型

Astra Control REST API隨附的資源類型具有下列特性：

- 每種資源類型都是使用架構來定義（通常是在Json中）
- 每個資源架構都包含資源類型和版本
- 資源類型是全域唯一的

### 資源執行個體

透過Astra Control REST API提供的資源執行個體具有下列特性：

- 資源執行個體是根據單一資源類型建立
- 資源類型會使用「媒體類型」值來表示
- 執行個體由Astra服務維護的狀態資料組成
- 每個執行個體都可透過專屬且長效的URL存取
- 如果資源執行個體可以有多种表示形式、則可以使用不同的媒體類型來要求所需的表示形式

### 資源集合

透過Astra Control REST API提供的資源集合具有下列特性：

- 單一資源類型的資源執行個體集稱為集合
- 資源集合具有獨特且長久存在的URL

### 執行個體識別碼

每個資源執行個體都會在建立時指派一個識別碼。此識別碼為128位元UUIDv4值。指派的UUIDv4值是全域唯一且不可變的。在發出API呼叫以建立新執行個體之後、會將具有相關ID的URL傳回給中的呼叫者 Location HTTP回應的標頭。您可以擷取識別碼、並在參照資源執行個體時用於後續通話。



資源識別碼是用於集合的主要金鑰。

## Astra資源的通用架構

每個Astra Control資源都是使用通用結構來定義。

### 通用資料

每個Astra資源都包含下表所示的關鍵值。

金鑰	說明
類型	一種全域唯一的資源類型、稱為*資源類型*。
版本	稱為*資源版本*的版本識別碼。
ID	全域唯一識別碼、稱為*資源識別碼*。
中繼資料	包含各種資訊的Json物件、包括使用者和系統標籤。

#### 中繼資料物件

每個Astra資源隨附的中繼資料Json物件包含下表所示的索引鍵值。

金鑰	說明
標籤	與資源相關聯之用戶端指定標籤的Json陣列。
建立時間戳記	Json字串包含時間戳記、指出資源建立的時間。
修改時間戳記	Json字串包含ISO-8601格式化時間戳記、指出上次變更資源的時間。
建立者	Json字串、包含建立資源之使用者ID的UUIDv4識別碼。如果資源是由內部系統元件所建立、且與建立實體沒有相關聯的UUID、則會使用* null * UUID。

#### 資源狀態

所選資源a state 用於協調生命週期轉換和控制存取的價值。

## HTTP詳細資料

Astra Control REST API使用HTTP和相關參數來處理資源執行個體和集合。HTTP實作的詳細資料如下。

### API交易與CRUD模式

Astra Control REST API實作交易模式、提供定義完善的作業和狀態轉換。

#### 要求及回應API交易

每個REST API呼叫都會以HTTP要求的形式執行、以供Astra服務使用。每個要求都會產生與用戶端相關的回應。此要求回應配對可視為API交易。

#### 支援CRUD營運模式

每個透過Astra Control REST API提供的資源執行個體和集合、都是根據\* CRUD\*模型來存取。共有四項作業、每項作業對應至單一HTTP方法。這些作業包括：

- 建立
- 讀取
- 更新
- 刪除

對於部分Astra資源、僅支援其中一部分作業。您應該檢閱 "[線上API參考](#)" 以取得特定API呼叫的詳細資訊。

## HTTP方法

下表顯示API支援的HTTP方法或動詞。

方法	CRUD	說明
取得	讀取	擷取資源執行個體或集合的物件屬性。當與集合一起使用時、這被視為* list* 作業。
貼文	建立	根據輸入參數建立新的資源執行個體。長期URL會傳回於中 Location 回應標頭：
放入	更新	使用隨附的Json要求本文來更新整個資源執行個體。無法使用者修改的關鍵值會保留下來。
刪除	刪除	刪除現有的資源執行個體。

## 要求和回應標頭

下表摘要說明Astra Control REST API所使用的HTTP標頭。



請參閱 "[RFC 7232.](#)" 和 "[RFC 7233](#)" 以取得更多資訊。

標頭	類型	使用注意事項
接受	申請	如果值為「/」或未提供、application/json 傳回內容類型回應標頭。如果該值設為Astra資源媒體類型、則內容類型標頭會傳回相同的媒體類型。
授權	申請	含有使用者API金鑰的承載權杖。
內容類型	回應	根據傳回 Accept 要求標頭：
ETag	回應	隨附於RFC 7232.定義的成功。此值是整個Json資源之MD5值的十六進位表示法。
如果相符	申請	如3.1 RFC 72332節所述實作的先決條件要求標頭、並支援* PUT *要求。
IF修改日期	申請	如第3.4節RFC 72332所述實作的先決條件要求標頭、並支援* PUT *要求。
如果沒有修改、自此	申請	如第3.4節RFC 72332所述實作的先決條件要求標頭、並支援* PUT *要求。
位置	回應	包含新建立資源的完整URL。

## 查詢參數

下列查詢參數可用於資源集合。請參閱 "[使用集合](#)" 以取得更多資訊。

查詢參數	說明
包括	包含讀取集合時應傳回的欄位。
篩選器	表示讀取集合時、資源必須符合的欄位。

查詢參數	說明
訂單者	決定讀取集合時傳回資源的排序順序。
限制	限制讀取集合時傳回的資源上限。
跳過	設定讀取集合時要經過和跳過的資源數目。
數	指出中繼資料物件是否應傳回資源總數。

## HTTP 狀態代碼

Astra Control REST API使用的HTTP狀態代碼如下所述。



Astra Control REST API也使用\*問題詳細資料做為HTTP API\*標準。請參閱 ["診斷與支援"](#) 以取得更多資訊。

程式碼	意義	說明
200	好的	表示未建立新資源執行個體的通話成功。
201.	已建立	已成功建立物件、且位置回應標頭包含該物件的唯一識別碼。
204.	無內容	雖然未傳回任何內容、但要求仍成功。
400	錯誤要求	無法辨識或不適當的要求輸入。
401.	未獲授權	使用者未獲授權、必須經過驗證。
403.	禁止	由於授權錯誤、存取遭拒。
404.04	找不到	要求中提及的資源不存在。
409.	衝突	建立物件的嘗試失敗、因為物件已經存在。
500	內部錯誤	伺服器發生一般內部錯誤。
503/50	服務無法使用	由於某些原因、服務尚未準備好處理要求。

## URL格式

用於透過REST API存取資源執行個體或集合的URL一般結構由數個值組成。此結構反映基礎物件模型和系統設計。

帳戶為root

每個REST端點的資源路徑根目錄為Astra帳戶。因此URL中的所有路徑都以開頭 `/account/{account_id}` 其中 `account_id` 為帳戶的唯一UUIDv4值。內部架構反映出所有資源存取都以特定帳戶為基礎的設計。

端點資源類別

Astra資源端點分為三種不同類別：

- 核心 (`/core`)
- 託管應用程式 (`/k8s`)
- 拓撲 (`/topology`)

請參閱 "資源" 以取得更多資訊。

#### 類別版本

這三種資源類別各有一個全域版本、可控制所存取資源的版本。根據慣例和定義、移至資源類別的新主要版本（例如 /v1 至 /v2）將會在API中引入中斷變更。

#### 資源執行個體或集合

根據資源執行個體或集合是否被存取、路徑中可以使用資源類型和識別碼的組合。

#### 範例

- 資源路徑

根據上述結構、端點的典型路徑為：`/accounts/{account_id}/core/v1/users`。

- 完整URL

對應端點的完整URL為：

`https://astra.netapp.io/accounts/{account_id}/core/v1/users`。



## 版權資訊

Copyright © 2023 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

## 商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。