概念
ONTAP Select

NetApp
February 03, 2026

This PDF was generated from https://docs.netapp.com/zh-tw/ontap-select-9141/concept_api_rest.html on February 03, 2026. Always check docs.netapp.com for the latest.

目錄

概念 1
REST Web服務基礎 1
架構與傳統限制 1
資源和狀態表示 1
URI端點 1
HTTP 訊息 1
JSON格式化 1
如何存取部署API 2
部署公用程式原生使用者介面 2
部署線上文件頁面ONTAP Select 2
自訂程式 2
部署API版本管理 2
基本營運特性 2
Hypervisor主機與ONTAP Select 非節點 3
物件識別碼 3
要求識別碼 3
同步和非同步呼叫 3
確認已完成長時間執行的工作 3
安全性 4
要求及回應API交易 4
控制API要求的輸入變數 4
解讀API回應 6
使用工作物件進行非同步處理 7
使用工作物件說明的非同步要求 7
查詢與 API 要求相關的工作物件 7
發出非同步要求的一般程序 8

概念

REST Web服務基礎

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Web服務API的設計時、它會建立一套技術和最佳實務做法、以揭露伺服器型資源並管理其狀態。它使用主流傳輸協定和標準、為部署和管理ONTAP Select 等叢集提供靈活的基礎。

架構與傳統限制

REST由Roy Fielding在博士課程中正式表達 "論文" 於2000年在UC爾灣舉行。它透過一組限制來定義架構風格、這些限制共同改善了網路型應用程式和基礎傳輸協定。這些限制會使用無狀態通訊協定、根據用戶端/伺服器架構來建立RESTful Web服務應用程式。

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。

用戶端與伺服器之間會交換訊息、以根據一般CRUD（建立、讀取、更新及刪除）模式來存取及變更資源狀態。

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動詞（例如GET和POST）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與一筆交易建立關聯、則必須在隨要求/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（以及與部署REST API搭配使用的選項）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。

如何存取部署API

由於REST Web服務固有的靈活度、ONTAP Select 因此可以透過多種不同的方式存取《支援》API。

部署公用程式原生使用者介面

存取API的主要方法是ONTAP Select 透過「功能不整合」網路使用者介面。瀏覽器會呼叫API、並根據使用者介面的設計重新格式化資料。您也可以透過部署公用程式命令列介面存取API。

部署線上文件頁面ONTAP Select

使用瀏覽器時、「支援功能」線上文件頁面可提供替代存取點。ONTAP Select除了提供直接執行個別API呼叫的方法之外、此頁面也包含API的詳細說明、包括每個呼叫的輸入參數和其他選項。API呼叫分為多個不同的功能區域或類別。

自訂程式

您可以使用多種不同的程式設計語言和工具來存取Deploy API。熱門選項包括Python、Java和Curl。使用API的程式、指令碼或工具會做為REST Web服務用戶端。使用程式設計語言可讓您更深入瞭解API、並提供自動化ONTAP Select 部署的機會。

部署API版本管理

隨附於Rashdeploy的REST API ONTAP Select 會指派版本編號。API版本編號與部署版本編號無關。您應該注意部署版本隨附的API版本、以及這會如何影響API的使用。

部署管理公用程式的目前版本包含REST API的第3版。舊版的部署公用程式包括下列API版本：

部署**2.8**及更新版本
: 包含REST API第3版的更新版本。ONTAP Select

部署**2.7.2**及更早版本
: 包含REST API第2版的更新版本。ONTAP Select

REST API的第2版和第3版不相容。如果您從包含API第2版的舊版升級至部署2.8或更新版本、則必須更新任何直接存取API的現有程式碼、以及使用命令列介面的任何指令碼。

基本營運特性

REST建立一套通用的技術和最佳實務做法、但每個API的詳細資料可能會因設計選項而異。在使用API之前、您應該先瞭解ONTAP Select 到「更新部署API」的詳細資料和操作特性。

Hypervisor主機與ONTAP Select 非節點

Hypervisor主機_是裝載ONTAP Select 一個整套虛擬機器的核心硬體平台。當在Hypervisor主機上部署及啟用某部支援的虛擬機器時、該虛擬機器被視為_這個節點_。ONTAP Select ONTAP Select在部署REST API的第3版中、主機和節點物件是分開且獨立的。這可建立一對多關係、讓一或多ONTAP Select 個支援節點可在同一個Hypervisor主機上執行。

物件識別碼

每個資源執行個體或物件在建立時都會指派一個唯一的識別碼。這些識別碼在ONTAP Select 特定的例子中是全域唯一的。發出建立新物件執行個體的 API 呼叫後、會將相關的 ID 值傳回給中的呼叫者 `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。

物件識別碼的內容和內部結構可隨時變更。當您參照相關的物件時、只能視需要在適用的API呼叫上使用識別碼。

要求識別碼

每個成功的API要求都會指派一個唯一的識別碼。識別碼會傳回至 `request-id` 相關 HTTP 回應的標頭。您可以使用要求識別碼、統稱為單一特定API要求回應交易的活動。例如、您可以根據要求ID擷取交易的所有事件訊息。

同步和非同步呼叫

伺服器執行從用戶端接收的HTTP要求的主要方法有兩種：

- 同步
  伺服器會立即執行要求、並以 200 、 201 或 204 的狀態碼回應。
- 非同步
  伺服器接受要求、並以狀態代碼 202 回應。這表示伺服器已接受用戶端要求、並開始執行背景工作以完成要求。最終成功或失敗無法立即取得、必須透過額外的API呼叫來判斷。

確認已完成長時間執行的工作

一般而言、任何可能需要較長時間才能完成的作業、都會使用非同步處理
伺服器的背景工作。使用部署 REST API 時、每個背景工作都會以錨定
工作物件、可追蹤工作並提供資訊、例如目前狀態。工作物件、
在建立背景工作之後、 HTTP 回應會傳回其唯一識別碼、

您可以直接查詢「工作」物件、以判斷相關聯的API呼叫是否成功。
如需其他資訊、請參閱_使用工作物件進行非同步處理_。

除了使用工作物件之外、還有其他方法可以判斷的成功或失敗
申請、包括：

- 事件訊息
  您可以使用傳回原始回應的要求 ID 、擷取與特定 API 呼叫相關的所有事件訊息。事件訊息通常包含成功或失敗的指示、也可在偵錯錯誤條件時使用。
- 資源狀態
  有幾個資源會維持一個狀態或狀態值、您可以查詢該值、以間接判斷要求的成功或失敗。

安全性

部署API使用下列安全技術：

- 傳輸層安全性
  透過網路在部署伺服器和用戶端之間傳送的所有流量都會透過 TLS 加密。不支援在未加密的通道上使用HTTP傳輸協定。支援TLS 1.2版。
- HTTP 驗證
  每項 API 交易都會使用基本驗證。每個要求都會新增一個HTTP標頭、其中包含基礎64字串中的使用者名稱和密碼。

要求及回應API交易

每個部署API呼叫都會以HTTP要求的形式執行、以供部署虛擬機器產生與用戶端相關的回應。此要求/回應配對被視為API交易。在使用部署API之前、您應該先熟悉可用於控制要求的輸入變數、以及回應輸出的內容。

控制API要求的輸入變數

您可以透過HTTP要求中設定的參數來控制API呼叫的處理方式。

要求標頭

您必須在HTTP要求中包含多個標頭、包括：

- 內容類型
  如果要求主體包含 JSON 、則必須將此標頭設定為 application/json 。
- 接受
  如果回應本文將包含 JSON 、則必須將此標頭設定為 application/json 。
- 授權
  必須使用以 base64 字串編碼的使用者名稱和密碼來設定基本驗證。

申請本文

申請本文的內容會因特定通話而有所不同。HTTP要求本文包含下列其中一項：

- 具有輸入變數的Json物件（例如新叢集的名稱）
- 空白

篩選物件

發出使用Get的API呼叫時、您可以根據任何屬性來限制或篩選傳回的物件。例如、您可以指定要符合的確切值：

```
<field>=<query value>
```

除了完全符合的項目、還有其他運算子可以傳回一系列值的一組物件。支援下列篩選操作員。ONTAP Select

營運者	說明
=	等於
<	小於
>	大於
&l;=	小於或等於
>=	大於或等於
	或
！	不等於
*	貪婪的萬用字元

您也可以使用null關鍵字或其否定（！null）做為查詢的一部分、根據是否設定特定欄位來傳回一組物件。

選取物件欄位

根據預設、使用Get發出API呼叫時、只會傳回唯一識別物件的屬性。這組最小欄位可做為每個物件的金鑰、而且會根據物件類型而有所不同。您可以使用「欄位查詢」參數、以下列方式選取其他物件屬性：

- 經濟實惠的欄位
  指定 `fields=*` 擷取在本機伺服器記憶體中維護的物件欄位、或只需少量處理即可存取。
- 昂貴的領域
  指定 `fields=**` 擷取所有物件欄位、包括需要額外伺服器處理才能存取的欄位。
- 自訂欄位選擇
  使用 `fields=FIELDNAME` 指定您要的確切欄位。要求多個欄位時、必須使用不含空格的逗號分隔值。

最佳實務做法是、務必找出您想要的特定欄位。您只能在需要時擷取一組廉價或昂貴的欄位。價格低廉且昂貴的分類是由NetApp根據內部效能分析所決定。特定欄位的分類可隨時變更。

排序輸出集中的物件

資源集合中的記錄會以物件定義的預設順序傳回。您可以使用 order_by 查詢參數變更訂單、並使用欄位名稱和排序方向、如下所示：

```
order_by=<field name> asc|desc
```

例如、您可以依遞增順序、以遞減順序排序類型欄位、然後依ID排序：

```
order_by=type desc, id asc
```

包含多個參數時、您必須以逗號分隔欄位。

分頁

使用Get存取同一類型物件的集合時發出API呼叫、預設會傳回所有相符的物件。如有需要、您可以使用main_Records查詢參數搭配要求來限制傳回的記錄數。例如：

```
max_records=20
```

如有需要、您可以將此參數與其他查詢參數合併、以縮小結果集範圍。例如、下列項目最多會傳回指定時間之後產生的 10 個系統事件：

`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

您可以針對事件（或任何物件類型）發出多個分頁要求。每次後續的API呼叫都應根據最後結果集中的最新事件、使用新的時間值。

解讀API回應

每個API要求都會對用戶端產生回應。您可以檢查回應以判斷是否成功、並視需要擷取其他資料。

HTTP狀態代碼

部署REST API所使用的HTTP狀態代碼如下所述。

程式碼	意義	說明
200	好的	表示未建立新物件的通話成功。
201	已建立	已成功建立物件；位置回應標頭包含物件的唯一識別碼。
202.02	已接受	執行要求的背景工作已開始執行、但作業尚未完成。
400	錯誤要求	無法辨識或不適當的要求輸入。
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。
405	不允許使用方法	資源不支援要求中的HTTP動詞。
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。
501..	未實作	URI已知但無法執行要求。

回應標頭

部署伺服器產生的HTTP回應中包含數個標頭、包括：

- 要求識別碼
  每個成功的 API 要求都會指派唯一的要求識別碼。
- 位置
  建立物件時、位置標頭會包含新物件的完整 URL 、包括唯一物件識別碼。

回應本文

與API要求相關的回應內容會因物件、處理類型、以及要求的成功或失敗而有所不同。回應本文會以Json呈現。

- 單一物件
  單一物件可根據要求傳回一組欄位。例如、您可以使用「Get」（取得）、使用唯一識別碼擷取叢集的選定內容。
- 多個物件
  可從資源集合傳回多個物件。在任何情況下、都會使用一致的格式 `num_records` 指出包含物件執行個體陣列的記錄和記錄數。例如、您可以擷取在特定叢集中定義的所有節點。

- 工作物件
  如果API呼叫以非同步方式處理、則會傳回工作物件、以固定背景工作。例如、用於部署叢集的POST要求會以非同步方式處理、並傳回工作物件。
- 錯誤物件
  如果發生錯誤、一律會傳回錯誤物件。例如、當您嘗試建立已存在名稱的叢集時、會收到錯誤訊息。
- 空白
  在某些情況下、不會傳回任何資料、回應本文是空的。例如、使用DELETE刪除現有主機之後、回應本文為空白。

使用工作物件進行非同步處理

有些部署API呼叫（尤其是建立或修改資源的呼叫）可能需要比其他呼叫更長的時間才能完成。以非同步方式部署這些長時間執行的要求。ONTAP Select

使用工作物件說明的非同步要求

在非同步執行 API 呼叫之後、 HTTP 回應代碼 202 表示該要求已成功驗證並接受、但尚未完成。此要求會以背景工作的形式處理、並在對用戶端的初始 HTTP 回應之後繼續執行。回應包括繫留要求的工作物件、包括其唯一識別碼。

您應該參閱ONTAP Select 「非同步部署」線上文件頁面、以判斷哪些API呼叫是以非同步方式運作。

查詢與 **API** 要求相關的工作物件

HTTP回應中傳回的工作物件包含數個內容。您可以查詢狀態內容、以判斷要求是否成功完成。工作物件可以處於下列其中一種狀態：

- 已佇列
- 執行中
- 成功
- 故障

輪詢工作物件以偵測工作的終端機狀態時、您可以使用兩種技巧：成功或失敗：

- 標準輪詢要求
  目前工作狀態會立即傳回
- 長輪詢要求
  只有在發生下列其中一種情況時、才會傳回工作狀態：
  - 狀態變更的時間比輪詢要求上提供的日期時間值還要晚
  - 逾時值已過期（1至120秒）

標準輪詢和長輪詢使用相同的API呼叫來查詢工作物件。不過、長輪詢要求包含兩個查詢參數： `poll_timeout` 和 `last_modified`。

您應該永遠使用長輪詢來減少部署虛擬機器上的工作負載。

發出非同步要求的一般程序

您可以使用下列高階程序來完成非同步 API 呼叫：

1. 發出非同步 API 呼叫。
2. 接收 HTTP 回應 202 、表示已成功接受要求。
3. 從回應本文擷取工作物件的識別碼。
4. 在迴圈內、在每個週期中執行下列步驟：
   a. 以長時間輪詢要求取得工作的目前狀態
   b. 如果工作處於非終端機狀態（佇列中、執行中）、請再次執行迴圈。
5. 當工作達到終端機狀態（成功、失敗）時停止。

版權資訊

Copyright © 2026 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。