



REST API存取與驗證功能 Active IQ Unified Manager

Active IQ Unified Manager

NetApp
January 15, 2026

目錄

| | |
|---|---|
| REST API存取與驗Active IQ Unified Manager 證功能 | 1 |
| 驗證 | 3 |
| HTTP狀態代碼用於Active IQ Unified Manager | 3 |
| 使用API進行Active IQ Unified Manager 效能不穩定的建議 | 4 |
| 疑難排解記錄 | 5 |
| 工作物件非同步處理 | 5 |
| 使用工作物件說明的非同步要求 | 5 |
| 查詢與 API 要求相關的工作物件 | 5 |
| 非同步要求的步驟 | 6 |
| 您好API伺服器 | 6 |

REST API存取與驗Active IQ Unified Manager 證功能

您可以使用任何REST用戶端或程式設計平台、利用基本的HTTP驗證機制來發出HTTP要求、藉此存取此靜態API Active IQ Unified Manager 。

要求與回應範例：

- 申請

```
GET
https://<IP
address/hostname>:<port_number>/api/v2/datacenter/cluster/clusters
```

- 回應

```
{
  "records": [
    {
      "key": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb:type=cluster,uuid=4c6bf721-2e3f-11e9-a3e2-00a0985badbb",
      "name": "fas8040-206-21",
      "uuid": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb",
      "contact": null,
      "location": null,
      "version": {
        "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17 10:28:33 UTC 2019",
        "generation": 9,
        "major": 5,
        "minor": 0
      },
      "isSanOptimized": false,
      "management_ip": "10.226.207.25",
      "nodes": [
        {
          "key": "4c6bf721-2e3f-11e9-a3e2-00a0985badbb:type=cluster_node,uuid=12cf06cc-2e3a-11e9-b9b4-00a0985badbb",
          "uuid": "12cf06cc-2e3a-11e9-b9b4-00a0985badbb",
          "name": "fas8040-206-21-01",
          "_links": {
            "self": {
              "href": "/api/datacenter/cluster/nodes/4c6bf721-2e3f-11e9-
```

```

a3e2-00a0985badbb:type=cluster_node,uuid=12cf06cc-2e3a-11e9-b9b4-
00a0985badbb"
    }
  },
  "location": null,
  "version": {
    "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17
10:28:33 UTC 2019",
    "generation": 9,
    "major": 5,
    "minor": 0
  },
  "model": "FAS8040",
  "uptime": 13924095,
  "serial_number": "701424000157"
},
{
  "key": "4c6bf721-2e3f-11e9-a3e2-
00a0985badbb:type=cluster_node,uuid=1ed606ed-2e3a-11e9-a270-
00a0985bb9b7",
  "uuid": "1ed606ed-2e3a-11e9-a270-00a0985bb9b7",
  "name": "fas8040-206-21-02",
  "_links": {
    "self": {
      "href": "/api/datacenter/cluster/nodes/4c6bf721-2e3f-11e9-
a3e2-00a0985badbb:type=cluster_node,uuid=1ed606ed-2e3a-11e9-a270-
00a0985bb9b7"
    }
  },
  "location": null,
  "version": {
    "full": "NetApp Release Dayblazer__9.5.0: Thu Jan 17
10:28:33 UTC 2019",
    "generation": 9,
    "major": 5,
    "minor": 0
  },
  "model": "FAS8040",
  "uptime": 14012386,
  "serial_number": "701424000564"
}
],
"_links": {
  "self": {
    "href": "/api/datacenter/cluster/clusters/4c6bf721-2e3f-11e9-
a3e2-00a0985badbb:type=cluster,uuid=4c6bf721-2e3f-11e9-a3e2-

```

```
00a0985badbb"  
    }  
  }  
},
```

- 「IP位址/主機名稱」是API伺服器的IP位址或完整網域名稱（FQDN）。
- 連接埠443

443是預設的HTTPS連接埠。您可以視需要自訂HTTPS連接埠。

若要從網頁瀏覽器發出HTTP要求、您必須使用REST API瀏覽器外掛程式。您也可以使用諸如Curl和Perl等指令碼平台來存取REST API。

驗證

Unified Manager支援API的基本HTTP驗證配置。為了確保資訊流程安全（要求和回應）、REST API只能透過HTTPS存取。API伺服器會為所有用戶端提供自我簽署的SSL憑證、以便進行伺服器驗證。此憑證可由自訂憑證（或CA憑證）取代。

您必須設定使用者對API伺服器的存取權限、才能叫用REST API。使用者可以是本機使用者（儲存在本機資料庫中的使用者設定檔）或LDAP使用者（如果您已將API伺服器設定為透過LDAP驗證）。您可以登入Unified Manager管理主控台使用者介面來管理使用者存取。

HTTP狀態代碼用於Active IQ Unified Manager

在執行API或疑難排解問題時、您應該注意Active IQ Unified Manager 到由資訊技術API使用的各種HTTP狀態代碼和錯誤代碼。

下表列出與驗證相關的錯誤代碼：

| HTTP狀態代碼 | 狀態代碼標題 | 說明 |
|----------|---------|--|
| 200 | 好的 | 成功執行同步 API 呼叫時傳回。 |
| 201. | 已建立 | 透過同步呼叫（例如 Active Directory 組態）來建立新資源。 |
| 202.02 | 已接受 | 成功執行非同步呼叫以進行資源配置功能（例如建立 LUN 和檔案共用）時傳回。 |
| 400 | 無效要求 | 表示輸入驗證失敗。使用者必須修正輸入、例如要求內容中的有效金鑰。 |
| 401. | 未獲授權的要求 | 您無權檢視資源/未獲授權。 |

| HTTP狀態代碼 | 狀態代碼標題 | 說明 |
|----------|---------|---|
| 403. | 禁止的要求 | 禁止存取您嘗試存取的資源。 |
| 404.. | 找不到資源 | 找不到您要聯絡的資源。 |
| 405 | 不允許使用方法 | 不允許使用方法。 |
| 429 | 要求太多 | 當使用者在特定時間內傳送過多要求時傳回。 |
| 500 | 內部伺服器錯誤 | 內部伺服器錯誤。無法從伺服器取得回應。此內部伺服器錯誤可能是永久性的、也可能不是永久性的。例如、如果您執行「Get」（取得）或「Get all」（取得全部）作業並收到此錯誤、建議您至少重複此作業五次。如果是永久性錯誤、則傳回的狀態代碼仍為 500。如果作業成功、則傳回的狀態代碼為 200。 |

使用API進行Active IQ Unified Manager 效能不穩定的建議

在Active IQ Unified Manager 使用API時、您應該遵循某些建議的實務做法。

- 所有回應內容類型必須採用下列格式、才能有效執行：

```
application/json
```

- API版本編號與產品版本編號無關。您應該使用Unified Manager執行個體可用的最新API版本。如需Unified Manager API版本的詳細資訊、請參閱Active IQ Unified Manager 「REST API版本管理功能」一節。
- 使用Unified Manager API更新陣列值時、您必須更新整個值字串。您無法將值附加至陣列。您只能取代現有的陣列。
- 您可以將篩選器運算子（例如pipe (|) 和萬用字元 (*) 用於所有查詢參數、但兩倍值除外、例如、度量API中的IOPS和效能。
- 使用篩選運算子萬用字元 (*) 和管道 (|) 的組合、避免查詢物件。它可能會擷取不正確的物件數。
- 使用篩選值時、請確定該值不包含任何「？」字元。這是為了降低SQL注入的風險。
- 請注意、任何API的「Get」（全部）要求最多會傳回1000筆記錄。即使您將「max_Records」參數設定為1000以上的值來執行查詢、也只會傳回1000筆記錄。
- 若要執行管理功能、建議您使用Unified Manager UI。

疑難排解記錄

系統記錄可讓您分析失敗的原因、並疑難排解執行API時可能發生的問題。

從下列位置擷取記錄、以疑難排解與API呼叫相關的問題。

| 記錄位置 | 使用 |
|---|---|
| <code>/var/log/osci/access_log.log</code> | 包含所有API呼叫詳細資料、例如叫用API的使用者名稱、開始時間、執行時間、狀態和URL。 您可以使用此記錄檔來檢查常用的API、或疑難排解任何GUI工作流程。您也可以根據執行時間、使用它來擴充分析。 |
| <code>/var/log/ocum/ocumserver.log</code> | 包含所有API執行記錄。 您可以使用此記錄檔來疑難排解及偵錯API呼叫。 |
| <code>/var/log/osci/server.log</code> | 包含所有Wildfly伺服器部署及啟動/停止服務相關記錄。 您可以使用此記錄檔來找出開始、停止或部署Wildfly伺服器期間發生任何問題的根本原因。 |
| <code>/var/log/osci/au.log</code> | 包含擷取單位相關記錄。 您可以在建立、修改或刪除ONTAP 任何物件時、使用此記錄檔、但Active IQ Unified Manager 這些物件不會反映在整個過程中。 |

工作物件非同步處理

支援支援「jobs」API、可擷取執行其他API時所執行工作的相關資訊。Active IQ Unified Manager您必須瞭解使用工作物件進行非同步處理的方式。

有些 API 呼叫（尤其是用於新增或修改資源的呼叫）可能需要比其他呼叫更長的時間才能完成。Unified Manager會以非同步方式處理這些長時間執行的要求。

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。

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：

- 正常
- 警告
- "party_f失敗"
- 錯誤

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：立即傳回目前的工作狀態。
- 長時間輪詢要求：當工作狀態移至「正常」、「錯誤」或「部分失敗」時

非同步要求的步驟

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、等待工作物件達到終端機狀態「正常、錯誤」或「部分失敗」
5. 確認工作的終端狀態、並擷取工作結果。

您好API伺服器

`_Hello API server_` 是示範如何Active IQ Unified Manager 使用簡單的REST用戶端、在靜態中叫用REST API的範例程式。範例程式會以Json格式提供API伺服器的基本詳細資料（伺服器僅支援「application/json」格式）。

使用的URI是：「`https://<hostname>/api/datacenter/svm/svms.`」此範例程式碼採用下列輸入參數：

- API伺服器IP位址或FQDN
- 選用：連接埠號碼（預設：443）
- 使用者名稱
- 密碼
- 回應格式（「application/json」）

若要叫用REST API、您也可以使用其他指令碼、例如JERSEY和REST-Easy來撰寫Java REST用戶端Active IQ Unified Manager 以供使用。您應該瞭解下列有關範例程式碼的考量事項：

- 使用HTTPS連線Active IQ Unified Manager 來叫用指定的REST URI
- 忽略Active IQ Unified Manager 由供應的憑證
- 在交握期間跳過主機名稱驗證
- 使用「`javax.net.ssl.HttpURLConnection`」建立URI連線

- 使用第三方程式庫（「org.apache.commons.codec.binary.Base64」）來建構HTTP基本驗證中使用的Base64編碼字串

若要編譯及執行範例程式碼、您必須使用Java編譯器1.8或更新版本。

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.URL;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.SSLSession;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;
import org.apache.commons.codec.binary.Base64;

public class HelloApiServer {

    private static String server;
    private static String user;
    private static String password;
    private static String response_format = "json";
    private static String server_url;
    private static String port = null;

    /*
     * * The main method which takes user inputs and performs the *
    necessary steps
     * to invoke the REST URI and show the response
    */ public static void main(String[] args) {
        if (args.length < 2 || args.length > 3) {
            printUsage();
            System.exit(1);
        }
        setUserArguments(args);
        String serverBaseUrl = "https://" + server;
        if (null != port) {
            serverBaseUrl = serverBaseUrl + ":" + port;
        }
        server_url = serverBaseUrl + "/api/datacenter/svm/svms";
        try {
            HttpsURLConnection connection =
getAllTrustingHttpsURLConnection();
            if (connection == null) {
```

```

        System.err.println("FATAL: Failed to create HTTPS
connection to URL: " + server_url);
        System.exit(1);
    }
    System.out.println("Invoking API: " + server_url);
    connection.setRequestMethod("GET");
    connection.setRequestProperty("Accept", "application/" +
response_format);
    String authString = getAuthorizationString();
    connection.setRequestProperty("Authorization", "Basic " +
authString);
    if (connection.getResponseCode() != 200) {
        System.err.println("API Invocation Failed : HTTP error
code : " + connection.getResponseCode() + " : "
+ connection.getResponseMessage());
        System.exit(1);
    }
    BufferedReader br = new BufferedReader(new
InputStreamReader((connection.getInputStream())));
    String response;
    System.out.println("Response:");
    while ((response = br.readLine()) != null) {
        System.out.println(response);
    }
    connection.disconnect();
} catch (Exception e) {
    e.printStackTrace();
}
}

/* Print the usage of this sample code */ private static void
printUsage() {
    System.out.println("\nUsage:\n\tHelloApiServer <hostname> <user>
<password>\n");
    System.out.println("\nExamples:\n\tHelloApiServer localhost admin
mypassword");
    System.out.println("\tHelloApiServer 10.22.12.34:8320 admin
password");
    System.out.println("\tHelloApiServer 10.22.12.34 admin password
");
    System.out.println("\tHelloApiServer 10.22.12.34:8212 admin
password \n");
    System.out.println("\nNote:\n\t(1) When port number is not
provided, 443 is chosen by default.");
}
}

```

```

/* * Set the server, port, username and password * based on user
inputs. */ private static void setUserArguments(
    String[] args) {
    server = args[0];
    user = args[1];
    password = args[2];
    if (server.contains(":")) {
        String[] parts = server.split(":");
        server = parts[0];
        port = parts[1];
    }
}

/*
 * * Create a trust manager which accepts all certificates and * use
this trust
 * manager to initialize the SSL Context. * Create a
URLConnection for this
 * SSL Context and skip * server hostname verification during SSL
handshake. * *
 * Note: Trusting all certificates or skipping hostname verification *
is not
 * required for API Services to work. These are done here to * keep
this sample
 * REST Client code as simple as possible.
 */ private static HttpURLConnection
getAllTrustingHttpsURLConnection() {
    HttpURLConnection conn =
null;
    try {
        /* Creating a trust manager that does not
validate certificate chains */
        TrustManager[]
trustAllCertificatesManager = new
TrustManager[]{new
X509TrustManager() {
    public X509Certificate[] getAcceptedIssuers(){return null;}
    public void checkClientTrusted(X509Certificate[]
certs, String authType){}
    public void checkServerTrusted(X509Certificate[]
certs, String authType){}
}};
        /* Initialize the
SSLContext with the all-trusting trust manager */
        SSLContext sslContext = SSLContext.getInstance("TLS");
        sslContext.init(null, trustAllCertificatesManager, new
SecureRandom());
        HttpURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory(
));
        URL url = new URL(server_url);
        conn =
(HttpURLConnection) url.openConnection();
        /* Do not perform an
actual hostname verification during SSL Handshake.
Let all
hostname pass through as verified.*/
        conn.setHostnameVerifier(new HostnameVerifier() {
            public

```

```

boolean verify(String host, SSLSession session) {
return true; } }); } catch (Exception e)
{ e.printStackTrace(); } return conn; }

/*
 * * This forms the Base64 encoded string using the username and
password *
 * provided by the user. This is required for HTTP Basic
Authentication.
 */ private static String getAuthorizationString() {
String userPassword = user + ":" + password;
byte[] authEncodedBytes =
Base64.encodeBase64(userPassword.getBytes());
String authString = new String(authEncodedBytes);
return authString;
}
}

```

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。