



# Active IQ Unified Manager での REST API へのアクセスおよび認証

## Active IQ Unified Manager 9.13

NetApp  
October 16, 2025

# 目次

Active IQ Unified Manager での REST API へのアクセスおよび認証 .....	1
認証 .....	3
Active IQ Unified Manager で使用される HTTP ステータスコード .....	3
Active IQ Unified Manager で API を使用する際の推奨事項です .....	4
トラブルシューティング用のログ .....	5
ジョブオブジェクトの非同期プロセス .....	6
ジョブオブジェクトを使用して記述された非同期要求 .....	6
API 要求に関連付けられたジョブオブジェクトの照会 .....	6
非同期要求の手順 .....	7
Hello API server 」と入力します .....	7

# Active IQ Unified Manager での REST API へのアクセスおよび認証

Active IQ Unified Manager REST APIには、基本のHTTP認証メカニズムを使用して問題 HTTP要求を実行できるRESTクライアントまたはプログラミングプラットフォームを使用してアクセスできます。

要求と応答の例：

• \* リクエスト \*

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

• \* 応答 \*

```
{
  "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 address/hostname* は、APIサーバのIPアドレスまたは完全修飾ドメイン名（FQDN）です。
- ポート443

443 は、デフォルトの HTTPS ポートです。必要に応じて、HTTPS ポートをカスタマイズできます。

Webブラウザから問題 HTTP要求を行うには、REST APIブラウザプラグインを使用する必要があります。cURL や Perl などのスクリプトプラットフォームを使用して、REST API にアクセスすることもできます。

## 認証

Unified Manager では、API の基本的な HTTP 認証方式がサポートされます。情報の流れ（要求と応答）をセキュリティで保護するために、REST API には HTTPS 経由でのみアクセスできます。API サーバは、サーバ検証のためにすべてのクライアントに自己署名 SSL 証明書を提供します。この証明書は、カスタム証明書（または CA 証明書）で置き換えることができます。

REST API を呼び出すには、API サーバへのユーザアクセスを設定する必要があります。ユーザには、ローカルユーザ（ローカルデータベースに格納されているユーザプロファイル）または LDAP ユーザ（LDAP 経由で認証するように API サーバを設定している場合）を指定できます。Unified Manager Administration Console のユーザインターフェイスにログインして、ユーザアクセスを管理できます。

## Active IQ Unified Manager で使用される HTTP ステータスコード

API の実行時や問題のトラブルシューティング時には、Active IQ Unified Manager API で使用されるさまざまな HTTP ステータスコードとエラーコードについて理解しておく必要があります。

次の表に、認証に関連するエラーコードを示します。

HTTP ステータスコード	ステータスコードタイトル	説明
200です	わかりました	同期 API 呼び出しの実行に成功した場合に返されます。
201年だ	作成済み	Active Directory の設定など、同期呼び出しによって新しいリソースが作成されたことを示します。

HTTP ステータスコード	ステータスコードタイトル	説明
202です	承認済み	LUN やファイル共有の作成など、プロビジョニング機能の非同期呼び出しの実行が成功したときに返されます。
400だ	無効な要求です	入力検証に失敗したことを示します。ユーザは、要求の本文に有効なキーを指定するなど、入力を修正する必要があります。
401	不正な要求です	リソースを表示する権限がありません / 権限がありません。
403です	禁止された要求です	アクセスしようとしていたリソースへのアクセスは禁止されています。
404です	リソースが見つかりません	アクセスしようとしているリソースが見つかりません。
405です	メソッドを使用できません	メソッドを使用できません
429	要求が多すぎます	指定した時間内にユーザが送信した要求が多すぎる場合に返されます。
500ドル	内部サーバエラー。	内部サーバエラー。サーバから応答を取得できませんでした。この内部サーバエラーは、永続的な場合とそうでない場合があります。たとえば、を実行するとします GET または GET ALL このエラーが表示された場合は、この処理を5回以上繰り返すことを推奨します。永続的なエラーの場合、引き続きステータスコード 500 が返されます。処理が成功すると、ステータスコード 200 が返されます。

## Active IQ Unified Manager で API を使用する際の推奨事項です

Active IQ Unified Manager で API を使用するときは、特定の推奨される方法に従ってください。

- 有効に実行するには、すべての応答コンテンツタイプが次の形式である必要があります。

- API のバージョン番号は、製品のバージョン番号とは関係ありません。Unified Manager インスタンスで使用可能な最新バージョンの API を使用する必要があります。Unified Manager API のバージョンの詳細については、「Active IQ Unified Manager での ST API のバージョン管理」を参照してください。
- Unified Manager API を使用して配列値を更新する場合は、値の文字列全体を更新する必要があります。配列に値を付加することはできません。既存の配列のみを交換できます。
- パイプ (|) やワイルドカード (+\*+) などのフィルタ演算子をすべてのクエリパラメータに使用できます。ただし、指標 API の IOPS やパフォーマンスなど、二重値は除きます。
- フィルタ演算子のワイルドカード (\*) とパイプ (|) を組み合わせてオブジェクトを照会しないでください。間違った数のオブジェクトが取得される可能性があります。
- フィルタに値を使用する場合は、値に何も含まれていないことを確認してください？を押します。これは、SQL インジェクションのリスクを軽減するためです。
- を参照してください GET (all) API に対する要求は、最大1000レコードを返します。を設定してクエリを実行した場合も同様です max\_records パラメータを1000より大きい値に設定すると、1000レコードのみが返されます。
- 管理機能を実行する場合は、Unified Manager UI を使用することを推奨します。

## トラブルシューティング用のログ

システムログを使用して、API の実行中に発生する可能性のある障害の原因を分析し、問題のトラブルシューティングを行うことができます。

API 呼び出しに関連する問題のトラブルシューティングを行うには、次の場所からログを取得します。

ログの場所	使用
/var/log/ocie/access_log.log	<p>API を呼び出しているユーザ名、開始時刻、実行時間、ステータス、URL など、API 呼び出しに関するすべての詳細情報が含まれます。</p> <p>このログファイルを使用して、頻繁に使用される API を確認したり、GUI ワークフローのトラブルシューティングを行ったりできます。また、実行時間に基づいて分析の規模を調整することもできます。</p>
/var/log/ocum/ocumserver.log	<p>すべての API 実行ログが含まれます。</p> <p>このログファイルを使用して、API 呼び出しのトラブルシューティングとデバッグを行うことができます。</p>

ログの場所	使用
/var/log/ocie/server.log	<p>すべての Wildfly サーバ構成と、サービスの開始 / 停止に関連するログが含まれています。</p> <p>このログファイルを使用して、Wildfly サーバの開始、停止、または導入時に発生する問題のルート原因を見つけることができます。</p>
/var/log/ocie/au.log	<p>Acquisition Unit 関連のログが含まれます。</p> <p>このログファイルは、ONTAP でオブジェクトを作成、変更、または削除したときに使用できますが、Active IQ Unified Manager REST API には反映されません。</p>

## ジョブオブジェクトの非同期プロセス

Active IQ Unified Manager は、を提供します `jobs` 他のAPIの実行中に実行されたジョブに関する情報を取得するAPI。非同期処理がジョブオブジェクトを使用してどのように動作するかを理解しておく必要があります。

一部の API 呼び出し、特にリソースの追加や変更を使用される API 呼び出しは、他の呼び出しよりも完了に時間がかかることがあります。Unified Manager は、これらの長時間実行されている要求を非同期的に処理します。

### ジョブオブジェクトを使用して記述された非同期要求

非同期的に実行される API 呼び出しを行うと、HTTP 応答コード 202 が返されます。この応答コードは、要求が正常に検証され受け入れられたものの、まだ完了していないことを示します。要求はバックグラウンドタスクとして処理され、クライアントへの最初の HTTP 応答後も引き続き実行されます。応答には、要求に対応するジョブオブジェクトと、その一意の識別子が含まれます。

### API 要求に関連付けられたジョブオブジェクトの照会

HTTP 応答で返されるジョブオブジェクトには、いくつかのプロパティが含まれています。状態プロパティを照会して、要求が正常に完了したかどうかを確認できます。ジョブオブジェクトは次のいずれかの状態になります。

- NORMAL
- WARNING
- PARTIAL\_FAILURES
- ERROR

ジョブオブジェクトをポーリングするときに、タスクの終了状態（成功または失敗）を検出するために使用できる 2 つの方法があります。

- 標準のポーリング要求：現在のジョブの状態がすぐに返されます。

- ログポーリング要求：ジョブの状態がになったとき NORMAL, ERROR, または PARTIAL\_FAILURES.

## 非同期要求の手順

非同期 API 呼び出しを完了する大まかな手順を次に示します。

1. 問題：非同期 API 呼び出し。
2. 要求が正常に受け取られたことを示す HTTP 応答 202 を受信します。
3. 応答の本文からジョブオブジェクトの識別子を抽出します。
4. ループ内で、ジョブオブジェクトが終了状態になるまで待ちます NORMAL, ERROR, または PARTIAL\_FAILURES.
5. ジョブの終了状態を確認し、ジョブの結果を取得します。

## Hello API server」と入力します

`_Hello API server_` は、シンプルな REST クライアントを使用して Active IQ Unified Manager で REST API を呼び出す方法を示すサンプルプログラムです。このサンプルプログラムでは、APIサーバに関する基本的な詳細情報をJSON形式で提供しています（サーバでのみサポートされます） `application/json` 形式）。

使用するURIは次のとおりです。 <https://<hostname>/api/datacenter/svm/svms>。このサンプルコードでは、次の入力パラメータを使用します。

- API サーバの IP アドレスまたは FQDN
- オプション：ポート番号（デフォルト：443）
- ユーザ名
- パスワード
- 応答の形式 (`application/json`)

REST API を呼び出すために、Jersey や RESTEasy など他のスクリプトを使用して Active IQ Unified Manager 用の Java REST クライアントを作成することもできます。サンプルコードに関して、次の点に注意する必要があります。

- Active IQ Unified Manager への HTTPS 接続を使用して、指定した REST URI を呼び出します
- Active IQ Unified Manager から提供される証明書を無視します
- ハンドシェイク中にホスト名の検証をスキップします
- 使用 `javax.net.ssl.HttpURLConnection` URI接続の場合
- サードパーティのライブラリを使用します (`org.apache.commons.codec.binary.Base64`) を使用して、HTTPベーシック認証で使用されるBase64エンコード文字列を作成します

サンプルコードをコンパイルして実行するには、Java コンパイラ 1.8 以降を使用する必要があります。

```
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.HttpURLConnection;
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 {
            HttpURLConnection 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
    HttpsURLConnection 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 HttpsURLConnection
    getAllTrustingHttpsURLConnection() {
        HttpsURLConnection 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());
            HttpsURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory(
            ));
            URL url = new URL(server_url);
            conn =
            (HttpsURLConnection) 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;
    }
}
```

## 著作権に関する情報

Copyright © 2025 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。