概念
ONTAP Select

NetApp
January 29, 2026

目次

概念

ONTAP Selectクラスタの導入と管理のための REST Web サービス基盤

Representational State Transfer（REST）は、分散Webアプリケーションの作成に使用される形式です。WebサービスAPIの設計においては、これによってサーバベースのリソースの公開とその状態の管理に関する一連のテクノロジとベストプラクティスが確立されます。主流のプロトコルと標準を使用して、 ONTAP Selectクラスタを導入および管理するための柔軟な基盤を提供します。

建築と古典的な制約

RESTはロイ・フィールディング博士によって正式に発表された。 "論文" 2000年にカリフォルニア大学アーバイン校で。発表されました。一連の制約を通してアーキテクチャスタイルを定義し、Webベースのアプリケーションとその基盤となるプロトコルを総合的に改善します。これらの制約により、ステートレスな通信プロトコルを用いたクライアント/サーバーアーキテクチャに基づくRESTful Webサービスアプリケーションが構築されます。

リソースと状態の表示

リソースはWebベース システムの基本コンポーネントです。REST Webサービス アプリケーションを作成する場合、設計の早い段階で次の作業を行います。

- システムまたはサーバーベースのリソースの識別 すべてのシステムはリソースを使用し、維持します。リソースには、ファイル、ビジネス トランザクション、プロセス、管理エンティティなどがあります。REST Webサービスに基づいてアプリケーションを設計する際に行う最初の作業の1つは、リソースを識別することです。
- リソースの状態と関連する状態操作の定義 リソースは常に有限の数の状態のいずれかになります。状態は明確に定義する必要があり、状態の変化に作用する操作も明確に定義する必要があります。

一般的な CRUD (作成、読み取り、更新、削除) モデルに従ってリソースの状態にアクセスし、変更するために、クライアントとサーバーの間でメッセージが交換されます。

URIエンドポイント

すべてのRESTリソースは、明確に定義されたアドレス指定方式を使用して定義、提供される必要があります。リソースが置かれているエンドポイントは、Uniform Resource Identifier（URI）で識別されます。URIは、ネットワークの各リソースに一意の名前を作成するための一般的なフレームワークです。Uniform Resource Locator（URL）は、リソースを識別してアクセスするためにWebサービスで使用されるURIの一種です。リソースは、通常、ファイル ディレクトリに似た階層構造で公開されます。

HTTPメッセージ

Hypertext Transfer Protocol（HTTP）は、Webサービスのクライアントとサーバがリソースに関する要求と応答のメッセージを交換する際に使用するプロトコルです。Web サービス アプリケーションの設計の一環として、HTTP 動詞 (GET や POST など) がリソースおよび対応する状態管理アクションにマッピングされます。

HTTPはステートレスです。したがって、関連する要求と応答のセットを 1 つのトランザクションに関連付けるには、要求/応答データ フローで伝送される HTTP ヘッダーに追加情報を含める必要があります。

JSONの形式

情報はさまざまな方法で構造化してクライアントとサーバー間で転送できますが、最も一般的なオプション (Deploy REST API で使用されるオプション) は JavaScript Object Notation (JSON) です。JSONは、単純なデータ構造をプレーン テキストで表現するための業界標準であり、リソースについての状態情報の転送に使用されます。

ONTAP Select Deploy API にアクセスする方法

REST Web サービスの本質的な柔軟性により、 ONTAP Select Deploy API にはさまざまな方法でアクセスできます。

ONTAP Select Deployに含まれるREST APIにはバージョン番号が割り当てられています。APIのバージョン番号は、Deployのリリース番号とは無関係です。DeployONTAP Select 9.17.1 Deploy 管理ユーティリティには、REST API バージョン 3 が含まれています。

デプロイユーティリティのネイティブユーザーインターフェース

APIにアクセスする主な方法は、 ONTAP Select DeployのWebユーザインターフェイスを使用することです。ブラウザはAPIを呼び出し、ユーザインターフェイスの設計に従ってデータを再フォーマットします。また、Deployユーティリティのコマンドラインインターフェイスからも APIにアクセスできます。

ONTAP Select Deploy オンラインドキュメントページ

ONTAP Select Deployのオンラインドキュメントページは、ブラウザ使用時の代替アクセスポイントとしてご利用いただけます。個々のAPI呼び出しを直接実行する方法に加えて、このページには、各呼び出しの入力パラメータやその他のオプションを含む、APIの詳細な説明も記載されています。API呼び出しは、複数の異なる機能領域またはカテゴリに分類されています。

カスタムプログラム

Deploy APIには、様々なプログラミング言語とツールからアクセスできます。Python、Java、cURLなどが一般的です。APIを使用するプログラム、スクリプト、またはツールは、REST Webサービスクライアントとして機能します。プログラミング言語を使用することで、APIをより深く理解し、 ONTAP Selectの導入を自動化する機会が得られます。

ONTAP Select Deploy API の基本的な動作特性

RESTで共通のテクノロジとベストプラクティスは確立されますが、各APIの詳細は設計内容に応じて異なる場合があります。API を使用する前に、 ONTAP Select Deploy API の詳細と動作特性を理解しておく必要があります。

ハイパーバイザーホストとONTAP Selectノード

_ハイパーバイザーホスト_は、 ONTAP Select仮想マシンをホストするコアハードウェアプラットフォームで

す。ONTAPONTAP Select仮想マシンがハイパーバイザーホストに導入され、アクティブになると、その仮想マシンは_ONTAP Selectノード_とみなされます。DeployREST APIバージョン3では、ホストオブジェクトとノードオブジェクトは別個の独立したオブジェクトです。これにより、1対多の関係が可能になり、1つ以上のONTAP Selectノードを同じハイパーバイザーホスト上で実行できます。

オブジェクトID

リソース インスタンスまたはオブジェクトには、作成時に一意の識別子がそれぞれ割り当てられます。これらの識別子は、 ONTAP Select Deployの特定のインスタンス内でグローバルに一意です。新しいオブジェクトインスタンスを作成するAPI呼び出しを発行した後、関連付けられたID値が呼び出し元に返されます。`location` HTTP 応答のヘッダー。リソース インスタンスを以降の呼び出しで参照する際は、この識別子を抽出して使用できます。

> オブジェクト識別子の内容と内部構造は変更になることがあります。識別子を使用するのは、該当するAPI呼び出しで関連付けられているオブジェクトを参照するために必要な場合だけにしてください。

リクエスト識別子

成功したAPIリクエストにはそれぞれ固有の識別子が割り当てられます。識別子は `request-id`関連する HTTP 応答のヘッダー。リクエストIDを使用すると、特定のAPIリクエスト・レスポンス・トランザクションのアクティビティをまとめて参照できます。例えば、リクエストIDに基づいて、あるトランザクションのすべてのイベントメッセージを取得できます。

同期呼び出しと非同期呼び出し

サーバーがクライアントから受信した HTTP 要求を実行する主な方法は 2 つあります。

- 同期 サーバーは要求を直ちに実行し、ステータス コード 200、201、または 204 で応答します。
- 非同期：サーバーはリクエストを受け入れ、ステータスコード202で応答します。これは、サーバーがクライアントからのリクエストを受け入れ、リクエストを完了するためのバックグラウンドタスクを開始したことを示します。最終的な成功または失敗はすぐには確認できず、追加のAPI呼び出しを通じて判断する必要があります。

長時間実行ジョブの完了を確認する

通常、完了までに長時間かかる操作は、サーバー側でバックグラウンドタスクを使用して非同期的に処理されます。DeployREST APIでは、すべてのバックグラウンドタスクはジョブオブジェクトによってアンカーされ、このオブジェクトはタスクを追跡し、現在の状態などの情報を提供します。バックグラウンドタスクが作成されると、一意の識別子を含むジョブオブジェクトがHTTPレスポンスで返されます。

ジョブオブジェクトを直接クエリすることで、関連するAPI呼び出しの成功または失敗を確認できます。詳細については、「ジョブオブジェクトを使用した非同期処理」を参照してください。

Job オブジェクトを使用する以外にも、次のような方法でリクエストの成功または失敗を判断できます。

- イベントメッセージ 元のレスポンスで返されたリクエストIDを使用して、特定のAPI呼び出しに関連付けられたすべてのイベントメッセージを取得できます。イベントメッセージには通常、成功または失敗の情報が含まれ、エラー状態のデバッグにも役立ちます。
- リソースの状態またはステータス リソースのいくつかは状態またはステータス値を維持しており、これを

照会することで、要求の成功または失敗を間接的に判断できます。

セキュリティ

Deploy API は次のセキュリティ テクノロジを使用します。

- トランスポート層セキュリティ（TLS） デプロイサーバーとクライアント間のネットワーク上で送信されるすべてのトラフィックは、TLSによって暗号化されます。暗号化されていないチャネルでのHTTPプロトコルの使用はサポートされていません。TLSバージョン1.2がサポートされています。
- HTTP認証：すべてのAPIトランザクションには基本認証が使用されます。ユーザー名とパスワードをBase64文字列で含むHTTPヘッダーがすべてのリクエストに追加されます。

ONTAP Selectのリクエストおよびレスポンス API トランザクション

すべてのDeploy API呼び出しは、Deploy仮想マシンへのHTTPリクエストとして実行され、クライアントへのレスポンスが生成されます。Deployこの要求と応答のペアでAPIトランザクションが構成されます。APIを使用する前に、リクエストを制御するために使用できる入力変数とレスポンス出力の内容について理解しておく必要があります。

API要求を制御する入力変数

HTTP リクエストに設定されたパラメータを通じて、API 呼び出しの処理方法を制御できます。

要求ヘッダー

HTTP リクエストには、次のようないくつかのヘッダーを含める必要があります。

- content-type リクエスト本文に JSON が含まれている場合、このヘッダーは application/json に設定する必要があります。
- accept レスポンス本文に JSON が含まれる場合、このヘッダーは application/json に設定する必要があります。
- 認証 基本認証は、base64 文字列でエンコードされたユーザー名とパスワードを使用して設定する必要があります。

リクエスト本文

要求の本文の内容は、それぞれの呼び出しに応じて異なります。HTTP要求の本文は、次のいずれかで構成されます。

- 入力変数（新しいクラスターの名前など）を含む JSON オブジェクト
- 空の

フィルターオブジェクト

GETを使用するAPI呼び出しを発行する際、返されるオブジェクトを任意の属性に基づいて制限またはフィルタできます。たとえば、一致する完全な値を指定できます。

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

完全一致に加えて、値の範囲に含まれるオブジェクトのセットを返すための演算子もいくつかありま す。ONTAPONTAP Select は、以下に示すフィルタリング演算子をサポートしています。

オペレーター	説明
=	等しい
<	小なり
>	より大きい
⇐	以下
>=	より大きいか等しい
|	または
!	等しくない
*	すべてに一致するワイルドカード

クエリの一部として null キーワードまたはその否定 (!null) を使用することで、特定のフィールドが設定されているかどうかに基づいてオブジェクトのセットを返すこともできます。

オブジェクトフィールドの選択

デフォルトでは、GETを使用してAPI呼び出しを実行すると、1つまたは複数のオブジェクトを一意に識別する属性のみが返されます。この最小のフィールド セットは各オブジェクトのキーとして機能し、オブジェクト タイプによって異なります。次の方法で、fields クエリ パラメータを使用して追加のオブジェクト プロパティを選択できます。

- 安価なフィールド指定 `fields=*`ローカル サーバー メモリに保持されているオブジェクト フィールド、またはアクセスにほとんど処理を必要としないオブジェクト フィールドを取得します。
- 高価なフィールドを指定する `fields=**`アクセスするために追加のサーバー処理を必要とするものも含め、すべてのオブジェクト フィールドを取得します。
- カスタムフィールドの選択使用 `fields=FIELDNAME`必要なフィールドを正確に指定します。複数のフィールドを要求する場合は、各値の間にスペースを入れずにカンマで区切る必要があります。

ベストプラクティスとして、必要なフィールドを常に個別に指定することを推奨します。必要な場合にのみ、安価なフィールドと高価なフィールドのセットを取得してください。安価なフィールドと高価なフィールドの分類は、 NetAppが社内のパフォーマンス分析に基づいて決定します。特定のフィールドの分類はいつでも変更される可能性があります。

出力セット内のオブジェクトを並べ替える

リソース コレクション内のレコードは、オブジェクトで定義されたデフォルトの順序で返されます。次のように、フィールド名と並べ替え方向を指定した order_by クエリ パラメータを使用して順序を変更できます。

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

たとえば、type フィールドを降順で並べ替え、次に id フィールドを昇順で並べ替えることができます。

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

複数のパラメータを指定する場合は、各フィールドをカンマで区切る必要があります。

ページネーション

GETを使用して同じタイプのオブジェクトのコレクションにアクセスするAPI呼び出しを発行すると、デフォルトでは一致するすべてのオブジェクトが返されます。必要に応じて、リクエストにmax_recordsクエリパラメータを指定して、返されるレコード数を制限できます。例えば：

```
max_records=20
```

必要に応じて、このパラメータを他のクエリパラメータと組み合わせて、結果セットを絞り込むことができます。たとえば、次の例では、指定された時間後に生成されたシステム イベントを最大 10 個返します。

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

複数のリクエストを発行して、イベント (または任意のオブジェクト タイプ) をページングできます。以降のAPI呼び出しでは、前回の結果セットの最後のイベントに基づいて新しい時間の値を使用する必要があります。

APIレスポンスを解釈する

各API要求でクライアントへの応答が生成されます。応答を調べて成功したかどうかを確認し、必要に応じて追加データを取得できます。

HTTPステータス コード

Deploy REST API で使用される HTTP ステータス コードについて以下に説明します。

コード	説明	説明
200	OK	新しいオブジェクトを作成しない呼び出しが成功したことを示します。
201	作成	オブジェクトが正常に作成されました。場所応答ヘッダーには、オブジェクトの一意の識別子が含まれます。
202	承認済み	要求を実行するために長時間実行されるバックグラウンド ジョブが開始されましたが、操作はまだ完了していません。
400	Bad request	要求の入力が認識されないか不適切です。
403	Forbidden	認証エラーのためアクセスが拒否されました。
404	Not found	要求で参照されているリソースが存在しません。
405	Method not allowed	リクエスト内の HTTP 動詞はリソースではサポートされていません。
409	対立	オブジェクトがすでに存在するため、オブジェクトの作成に失敗しました。
500	内部エラー	サーバで一般的な内部エラーが発生しました。
501	実装されていません	URI は既知ですが、要求を実行することができません。

応答ヘッダー

デプロイ サーバーによって生成される HTTP 応答には、次のようないくつかのヘッダーが含まれます。

- request-id 成功したすべての API リクエストには、一意のリクエスト ID が割り当てられます。

- 場所 オブジェクトが作成されると、場所ヘッダーには、一意のオブジェクト識別子を含む新しいオブジェクトへの完全な URL が含まれます。

応答の本文

APIリクエストに関連付けられたレスポンスの内容は、オブジェクト、処理タイプ、リクエストの成功または失敗によって異なります。レスポンス本文はJSON形式でレンダリングされます。

- 単一オブジェクト リクエストに基づいて、フィールドのセットを含む単一のオブジェクトを返すことができます。たとえば、GETで一意の識別子を使用して、クラスタの選択したプロパティを取得できます。
- 複数のオブジェクト リソースコレクションから複数のオブジェクトを返すことができます。いずれの場合も、一貫した形式が使用され、 `num_records`オブジェクト インスタンスの配列を含むレコードとレコードの数を示します。例えば、特定のクラスターに定義されているすべてのノードを取得できます。
- ジョブオブジェクト API呼び出しが非同期的に処理される場合、バックグラウンドタスクをアンカーするジョブオブジェクトが返されます。例えば、クラスターのデプロイに使用されるPOSTリクエストは非同期的に処理され、ジョブオブジェクトを返します。
- エラーオブジェクト エラーが発生した場合、常にエラーオブジェクトが返されます。例えば、既に存在する名前でクラスターを作成しようとするとエラーが発生します。
- 空 場合によっては、データが返されず、レスポンス本文が空になります。例えば、DELETEを使用して既存のホストを削除した場合などです。

ONTAP Selectのジョブオブジェクトを使用した非同期処理

一部のDeploy API呼び出し、特にリソースの作成または変更を行う呼び出しは、他の呼び出しよりも完了までに時間がかかる場合があります。ONTAPONTAP Select Deployは、これらの長時間実行され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。

どの API 呼び出しが非同期に動作するかを判断するには、 ONTAP Select Deploy のオンライン ドキュメント ページを参照する必要があります。

APIリクエストに関連付けられたジョブオブジェクトをクエリする

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。

- キューに登録
- 実行中
- 成功

- 失敗

ジョブ オブジェクトをポーリングしてタスクの最終状態（成功または失敗）を検出するには、2つの方法があります。

- 標準ポーリング要求現在のジョブ状態が直ちに返されます
- ロングポーリング要求ジョブ状態は、次のいずれかが発生した場合にのみ返されます。
  - 状態は、ポーリング要求で指定された日時値よりも最近に変更されました
  - タイムアウト値が経過しました（1～120秒）

標準ポーリングとロングポーリングは、同じAPI呼び出しを使用してジョブオブジェクトをクエリします。ただし、ロング ポーリング リクエストには、次の 2 つのクエリ パラメータが含まれます。 `poll_timeout`そして `last_modified`。

デプロイ仮想マシンのワークロードを軽減するには、常にロングポーリングを使用する必要があります。

非同期リクエストを発行するための一般的な手順

以下は、非同期API呼び出しを完了する手順の概要です。

1. 非同期API呼び出しを実行します。
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. ループ内では、各サイクルで次の操作を実行します。
   a. ロングポーリングリクエストでジョブの現在の状態を取得する
   b. ジョブが非終了状態（キューに入れられ、実行中）の場合、ループを再度実行します。
5. ジョブが終了状態 (成功、失敗) に達したら停止します。

著作権に関する情報

商標に関する情報