



REST APIによる自動化

SnapCenter Software 6.0

NetApp
September 23, 2024

目次

| | |
|--|----|
| REST APIによる自動化 | 1 |
| REST APIの概要 | 1 |
| SnapCenter REST APIに標準でアクセスする方法 | 1 |
| 基盤となるREST Webサービス | 1 |
| 基本的な動作特性 | 2 |
| API要求を制御する入力変数 | 4 |
| API応答の解釈 | 7 |
| SnapCenter ServerとプラグインでサポートされるREST API | 10 |
| Swagger API Webページを使用してREST APIにアクセスする方法 | 18 |
| REST APIの使用を開始する | 18 |

REST APIによる自動化

REST APIの概要

REST APIを使用して、SnapCenterのいくつかの管理処理を実行できます。REST APIはSwagger Webページから利用できます。

<SnapCenter_IP_address_or_name>のドキュメントを表示したり、<SnapCenter_port>呼び出しを手動で問題したりするには、Swagger Webページ (`https://:/swagger/`) にアクセスします。

REST APIをサポートするプラグインは次のとおりです。

- Microsoft SQL Server用プラグイン
- SAP HANAデータベース向けプラグイン
- カスタムプラグイン
- Oracleデータベース向けプラグイン

SnapCenter REST APIに標準でアクセスする方法

SnapCenter REST APIには、RESTクライアントをサポートする任意のプログラミング言語を使用して直接アクセスできます。一般的な言語には、Python、PowerShell、Javaなどがあります。

基盤となるREST Webサービス

Representational State Transfer (REST) は、分散Webアプリケーションを作成するための形式です。WebサービスAPIの設計に適用すると、サーバベースのリソースを公開してその状態を管理するための一連のテクノロジー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、SnapCenterを管理するための柔軟な基盤を提供します。

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際の初期設計タスクには、次のものがあります。

システムまたはサーバ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、または管理エンティティがあります。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の最初のタスクの1つは、リソースを特定することです。

リソースの状態および関連する状態操作の定義

リソースは常に有限数の状態のいずれかにあります。状態、および状態の変化に影響を与えるために使用され

る関連操作は、明確に定義する必要があります。

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および利用可能にする必要があります。リソースが配置され、識別されるエンドポイントでは、Uniform Resource Identifier (URI) が使用されます。

URIは、ネットワーク内の各リソースに一意的な名前を作成するための一般的なフレームワークを提供します。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ため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メソッドは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、関連する一連の要求と応答を1つのトランザクションの一部として関連付けるには、要求と応答の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

Webサービスのクライアントとサーバの間で情報を構造化して転送する方法はいくつかありますが、最も一般的な方法は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、リソースを記述する状態情報の転送に使用されます。SnapCenter REST APIは、JSONを使用して、各HTTP要求と応答の本文で伝送されるデータをフォーマットします。

基本的な動作特性

RESTでは共通のテクノロジーとベストプラクティスが確立されますが、各APIの詳細は設計の選択内容によって異なります。

要求と応答のAPIトランザクション

すべてのREST API呼び出しは、SnapCenterサーバシステムへのHTTP要求として実行され、クライアントへの関連する応答が生成されます。この要求と応答のペアはAPIトランザクションとみなされます。

APIを使用する前に、要求の制御に使用できる入力変数と応答出力の内容を理解しておく必要があります。

CRUD操作のサポート

SnapCenter REST APIで使用できる各リソースは、CRUDモデルに基づいてアクセスされます。

- 作成
- 読み取り

- 更新
- 削除

一部のリソースでは、一部の処理のみがサポートされます。

オブジェクトID

各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割り当てられます。ほとんどの場合、識別子は128ビットUUIDです。これらの識別子は、特定のSnapCenterサーバ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呼び出しを発行すると、関連付けられたIDのURLがHTTP応答のlocationヘッダーで呼び出し元に返されます。識別子を抽出して以降の呼び出しでリソースインスタンスを参照する際に使用できます。



オブジェクトIDの内容と内部構造はいつでも変更できます。識別子は、関連するオブジェクトを参照する場合にのみ、該当するAPI呼び出しで使用してください。

オブジェクトのインスタンスとコレクション

リソースパスとHTTPメソッドに応じて、特定のオブジェクトインスタンスまたはオブジェクトのコレクションにAPI呼び出しを適用できます。

同期操作と非同期操作

SnapCenterは、クライアントから受信したHTTP要求を同期または非同期で実行します。

同期処理

SnapCenterは要求をただちに実行し、成功した場合はHTTPステータスコード200または201で応答します。

GETメソッドを使用する要求はすべて、常に同期的に実行されます。また、POSTを使用する要求は、2秒以内に完了すると予想される場合は同期的に実行されるように設計されています。

非同期処理

非同期要求が有効な場合、SnapCenterは要求を処理するバックグラウンドタスクと、タスクをアンカーするジョブオブジェクトを作成します。HTTPステータスコード202が、ジョブオブジェクトとともに呼び出し元に返されます。成功または失敗を判断するには、ジョブの状態を取得する必要があります。

POSTおよびDELETEメソッドを使用する要求は、完了までに2秒以上かかると予想される場合に非同期で実行されるように設計されています。

セキュリティ

REST APIで提供されるセキュリティは、主にSnapCenterで使用できる既存のセキュリティ機能に基づいています。APIで使用されるセキュリティは次のとおりです。

トランスポート層セキュリティ

SnapCenterサーバとクライアントの間でネットワーク経由で送信されるすべてのトラフィックは、通常、SnapCenter設定に基づいてTLSを使用して暗号化されます。

HTTP認証

HTTPレベルでは、APIトランザクションにベーシック認証が使用されます。base64文字列にユーザ名とパスワードを含むHTTPヘッダーが各要求に追加されます。

API要求を制御する入力変数

API呼び出しの処理方法は、HTTP要求で設定されたパラメータと変数を使用して制御できます。

HTTP メソッド

SnapCenter REST APIでサポートされるHTTPメソッドを次の表に示します。



RESTエンドポイントごとにすべてのHTTPメソッドを使用できるわけではありません。

| HTTPメソッド | 説明 |
|----------|--|
| 取得 | リソースインスタンスまたはコレクションのオブジェクトプロパティを取得します。 |
| 投稿 | 指定した入力に基づいて新しいリソースインスタンスを作成します。 |
| 削除 | 既存のリソースインスタンスを削除します。 |
| PUT | 既存のリソースインスタンスを変更します。 |

要求ヘッダー

HTTP要求には複数のヘッダーを含める必要があります。

コンテンツタイプ

要求の本文にJSONが含まれている場合は、このヘッダーを `application/json` に設定する必要があります。

同意する

このヘッダーは、`application/json` に設定してください。

許可

ベーシック認証は、base64文字列としてエンコードされたユーザ名とパスワードで設定する必要があります。

リクエストの本文

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。

- JSONオブジェクトと入力変数
- 空

オブジェクトのフィルタリング

GETを使用するAPI呼び出しを実行するときに、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リングできます。たとえば、次のように完全に一致する値を指定できます。

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

完全一致に加えて、値の範囲内の一連のオブジェクトを返すための他の演算子も使用できます。SnapCenter REST APIでは、次の表に示すフィルタ演算子がサポートされます。

| 運用者 | 説明 |
|-----|-----------------|
| = | 等しい |
| < | より小さい |
| > | 次の値より大きい |
| ← | 以下 |
| >= | 以上 |
| 更新 | または |
| なんだ | 等しくない |
| * | すべてに一致するワイルドカード |

また、クエリの一部として **null** キーワードまたはその negation ***!null*** を使用して、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コレクションを返すこともできます。



通常、設定されていないフィールドはクエリの照合から除外されます。

特定のオブジェクトフィールドの要求

デフォルトでは、GETを使用してAPI呼び出しを実行すると、オブジェクトを一意に識別する属性のみが返されます。このフィールドの最小セットは、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。クエリパラメータを使用すると、次の方法で追加のオブジェクトプロパティを選択でき `fields` ます。

共通または標準のフィールド

fields=* を指定すると、最もよく使用されるオブジェクトフィールドが取得されます。これらのフィールドは通常、ローカルサーバメモリに保持されるか、アクセスするための処理をほとんど必要としません。これらは、URLパスキー (UUID) を指定してGETを使用したあとにオブジェクトに対して返されるプロパティと同じです。

すべてのフィールド

fields=* を指定すると 'アクセスするために追加のサーバ処理が必要なフィールドも含め' すべてのオブジェクトフィールドが取得されます

カスタムフィールドの選択

fields=<field_name> を使用すると、必要なフィールドを正確に指定できます。複数のフィールドを要求する場合は、値をカンマで区切ってスペースなしで指定する必要があります。



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。一連の共通フィールドまたはすべてのフィールドは、必要に応じて取得する必要があります。共通として分類されるフィールドで、**fields=*** を使用して返されるフィールドは、ネットアップの内部パフォーマンス分析に基づいて決定されます。フィールドの分類は、今後のリリースで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のソート

リソースコレクション内のレコードは、オブジェクトによって定義されたデフォルトの順序で返されます。次のように、フィールド名とソート方向を指定したクエリパラメータを使用して順序を変更できます `order_by`。

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

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

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

- ソートフィールドを指定しても方向を指定しない場合、値は昇順でソートされます。
- 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。

コレクション内のオブジェクトを取得するときのページネーション

GET を使用して API 呼び出しを発行し、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と、SnapCenter は 2 つの制約に基づいてできるだけ多くのオブジェクトを返します。これらの各制約は、リクエストの追加のクエリパラメータを使用して制御できます。特定の GET 要求に対して最初に到達した制約によって要求が終了するため、返されるレコード数が制限されます。



すべてのオブジェクトについての処理が完了する前に要求が終了した場合、次のレコードのバッチを取得するために必要なリンクが応答に含まれます。

オブジェクト数の制限

デフォルトでは、SnapCenter は GET 要求に対して最大 10,000 個のオブジェクトを返します。この制限は、`_max_records_query` パラメータを使用して変更できます。例：

```
max_records=20
```

実際に返されるオブジェクトの数は、関連する時間の制約やシステム内のオブジェクトの総数に基づいて、有効な最大数よりも少なくなることがあります。

オブジェクトの読み出しに使用する時間の制限

デフォルトでは、SnapCenterはGET要求に許可された時間内にできるだけ多くのオブジェクトを返します。デフォルトのタイムアウトは15秒です。この制限は、`_return_timeout_query` パラメータを使用して変更できます。例：

```
return_timeout=5
```

実際に返されるオブジェクトの数は、関連するオブジェクト数の制約やシステム内のオブジェクトの総数に基づいて、有効な最大数よりも少なくなることがあります。

結果セットの絞り込み

必要に応じて、これらの2つのパラメータを追加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、指定した時間が経過すると生成されたEMSイベントが最大10個返されます。

```
time⇒ 2018-04-04T15:41:29.140265Z&max_records=10
```

複数の要求を発行してオブジェクトをページングすることができます。以降の各API呼び出しでは、最後の結果セットの最新のイベントに基づいて新しい時間値を使用する必要があります。

サイズのプロパティ

一部のAPI呼び出しおよび特定のクエリパラメータで使用される入力値は数値です。バイト単位で整数を指定する代わりに、必要に応じて次の表に示すサフィックスを使用できます。

| サフィックス | 説明 |
|--------|-------------------------------|
| KB | KBキロバイト（1024バイト）またはキビバイト |
| MB | MBメガバイト（KB x 1024バイト）またはメビバイト |
| GB | GBギガバイト（MB x 1024バイト）またはギビバイト |
| TB | テラバイト（GB x 1024バイト）またはテビバイト |
| PB | ペタバイト（TB x 1024バイト）またはペビバイト |

API応答の解釈

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判断し、必要に応じて追加のデータを取得する必要があります。

HTTPステータスコード

SnapCenter REST APIで使用されるHTTPステータスコードを次に示します。

| コード | 説明 |
|-----|---|
| 200 | OK は、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 | オブジェクトが作成されました。応答のlocationヘッダーには、オブジェクトの一意的識別子が含まれています。 |
| 202 | 承認バックグラウンドジョブで要求の実行が開始されましたが、まだ完了していません。 |
| 400 | 要求が正しくありません。要求の入力が認識されていないか、適切ではありません。 |
| 401 | 権限のないユーザ認証に失敗しました。 |
| 403 | 認証（RBAC）エラーにより、アクセスが禁止されています。 |
| 404 | 要求で参照されているリソースが見つかりません。 |
| 405 | メソッドが許可されていません要求内の HTTP メソッドはリソースに対してサポートされていません |
| 409 | 競合先に別のオブジェクトを作成する必要があるか、要求された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 | 内部エラーサーバーで一般的な内部エラーが発生しました。 |

応答ヘッダー

SnapCenterによって生成されるHTTP応答には、いくつかのヘッダーが含まれています。

場所

オブジェクトが作成されると、新しいオブジェクトへの完全なURL（オブジェクトに割り当てられた一意の識別子を含む）が格納されます。

コンテンツタイプ

通常は `application/json`。

応答の本文

API要求の結果として返される応答の本文の内容は、オブジェクト、処理タイプ、および要求の成否によって異なります。応答は常にJSON形式で表示されます。

単一のオブジェクト

要求に基づいて一連のフィールド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を使用すると、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、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す一貫した形式が使用され `num_records` ます。たとえば、特定のクラスタで定義されているノードを取得できます。

ジョブオブジェクト

API呼び出しが非同期で処理されると、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。たとえば、クラスタ構成の更新に使用されるPATCH要求は非同期で処理され、ジョブオブジェクトが返されます。

エラーオブジェクト

エラーが発生した場合は、常にErrorオブジェクトが返されます。たとえば、クラスタに対して定義されていないフィールドを変更しようとする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、応答の本文に空のJSONオブジェクトが含まれていることがあります。

エラー

エラーが発生した場合は、応答の本文でエラーオブジェクトが返されます。

形式

エラーオブジェクトの形式は次のとおりです。

```
"error": {
  "message": "<string>",
  "code": <integer>[,
  "target": "<string>"]
}
```

コード値を使用して一般的なエラータイプまたはカテゴリを特定し、メッセージを使用して特定のエラーを特定できます。該当する場合は、エラーに関連する特定のユーザ入力ターゲットフィールドに表示されます。

一般的なエラーコード

次の表に、一般的なエラーコードを示します。特定のAPI呼び出しには、追加のエラーコードが含まれることがあります。

| コード | 説明 |
|-----|---|
| 409 | 同じ識別子のオブジェクトがすでに存在します。 |
| 400 | フィールドの値が無効であるか、値が指定されていないか、追加のフィールドが指定されています。 |
| 400 | この処理はサポートされていません。 |

| コード | 説明 |
|-----|-------------------------|
| 405 | 指定した識別子のオブジェクトが見つかりません。 |
| 403 | 要求を実行する権限が拒否されました。 |
| 409 | リソースが使用中です。 |

SnapCenter ServerとプラグインでサポートされるREST API

SnapCenter REST APIで使用できるリソースは、SnapCenter APIドキュメントページに表示されるカテゴリ別に分類されています。各リソースの簡単な説明と基本リソースパスを次に示します。また、必要に応じて使用上のその他の考慮事項も示します。

認証

このAPIを使用して、SnapCenterサーバにログインできます。このAPIは、後続の要求の認証に使用されるユーザ認証トークンを返します。

ドメイン

APIを使用してさまざまな処理を実行できます。

- SnapCenter内のすべてのドメインを取得する
- 特定のドメインの詳細を取得する
- ドメインの登録または登録解除
- ドメインを変更する

ジョブ

APIを使用してさまざまな処理を実行できます。

- SnapCenterですべてのジョブを取得する
- ジョブのステータスの取得
- ジョブをキャンセルまたは停止する

設定

APIを使用してさまざまな処理を実行できます。

- クレデンシャルの登録、変更、削除
- SnapCenterサーバに登録されているクレデンシャル情報を表示します。
- 通知の設定
- Eメール通知を送信するように現在設定されているSMTPサーバに関する情報を取得し、SMTPサーバの名前、受信者の名前、および送信者の名前を表示します。
- SnapCenterサーバログインの多要素認証（MFA）設定を表示します。

- SnapCenterサーバログインのMFAを有効または無効にして設定する
- MFAのセットアップに必要な構成ファイルを作成

ホスト

APIを使用してさまざまな処理を実行できます。

- すべてのSnapCenterホストを照会
- SnapCenterから1つ以上のホストを削除する
- 名前によるホストの取得
- ホスト上のすべてのリソースを取得する
- リソースIDを使用してリソースを取得する
- プラグイン設定の詳細を取得する
- プラグインホストの設定
- Plug-in for Microsoft SQL Serverホストのすべてのリソースを取得する
- Plug-in for Oracle Databaseホストのすべてのリソースを取得
- カスタムアプリケーションホスト用プラグインのすべてのリソースを取得する
- Plug-in for SAP HANAホストのすべてのリソースを取得する
- インストールされているプラグインの取得
- 既存のホストにプラグインをインストールする
- ホストパッケージのアップグレード
- 既存のホストからプラグインを削除する
- ホストにプラグインを追加する
- ホストの追加または変更
- Linuxホストのシグネチャの取得
- Linuxホストのシグネチャの登録
- ホストをメンテナンスモードまたは本番モードにする
- ホストでプラグインサービスを開始または再起動する
- ホストの名前を変更する

リソース

APIを使用してさまざまな処理を実行できます。

- すべてのリソースを取得
- リソースIDを使用してリソースを取得する
- Plug-in for Microsoft SQL Serverホストのすべてのリソースを取得する
- Plug-in for Oracle Databaseホストのすべてのリソースを取得

- カスタムアプリケーションホスト用プラグインのすべてのリソースを取得する
- Plug-in for SAP HANAホストのすべてのリソースを取得する
- キーを使用したMicrosoft SQL Serverリソースの取得
- キーを使用したカスタムリソースの取得
- カスタムアプリケーションホスト用プラグインのリソースを変更する
- キーを使用してカスタムアプリケーションホスト用プラグインのリソースを削除する
- キーを使用したSAP HANAリソースの取得
- Plug-in for SAP HANAホストのリソースを変更する
- キーを使用してPlug-in for SAP HANAホストのリソースを削除する
- キーを使用したOracleリソースの取得
- Oracleアプリケーションボリュームリソースを作成する
- Oracleアプリケーションボリュームリソースを変更する
- キーを使用してOracleアプリケーションボリュームリソースを削除する
- Oracleリソースのセカンダリの詳細を取得する
- Plug-in for Microsoft SQL Serverを使用したMicrosoft SQL Serverリソースのバックアップ
- Plug-in for Oracle Databaseを使用したOracleリソースのバックアップ
- カスタムアプリケーション用プラグインを使用してカスタムリソースをバックアップする
- SAP HANAデータベースの設定
- Oracleデータベースの設定
- SQLデータベースのバックアップをリストアする
- Oracleデータベースバックアップのリストア
- カスタムアプリケーションのバックアップのリストア
- カスタムプラグインリソースを作成する
- SAP HANAリソースを作成する
- カスタムアプリケーション用のプラグインを使用してカスタムリソースを保護
- Plug-in for Microsoft SQL Serverを使用したMicrosoft SQL Serverリソースの保護
- 保護されたMicrosoft SQL Serverリソースを変更する
- Microsoft SQL Serverリソースの保護の解除
- Plug-in for Oracle Databaseを使用したOracleリソースの保護
- 保護されているOracleリソースを変更する
- Oracleリソースの保護の解除
- カスタムアプリケーション用プラグインを使用したバックアップからのリソースのクローニング
- Plug-in for Oracle Databaseを使用したバックアップからのOracleアプリケーションボリュームのクローニング
- Plug-in for Microsoft SQL Serverを使用したバックアップからのMicrosoft SQL Serverリソースのクローニ

ング

- Microsoft SQL Serverリソースのクローンライフサイクルを作成する
- Microsoft SQL Serverリソースのクローンライフサイクルを変更する
- Microsoft SQL Serverリソースのクローンライフサイクルを削除する
- 既存のMicrosoft SQL ServerデータベースをローカルディスクからNetApp LUNに移動する
- Oracleデータベースのクローン仕様ファイルの作成
- Oracleリソースのクローン更新ジョブをオンデマンドで開始する
- クローン仕様ファイルを使用して、バックアップからOracleリソースを作成する
- データベースをセカンダリレプリカにリストアし、データベースを可用性グループに結合します。
- Oracleアプリケーションボリュームリソースを作成する

バックアップ

APIを使用してさまざまな処理を実行できます。

- バックアップの名前、タイプ、プラグイン、リソース、または日付を指定してバックアップの詳細を取得する
- すべてのバックアップを取得
- バックアップの詳細を取得
- バックアップの名前変更または削除
- Oracleバックアップのマウント
- Oracleバックアップのアンマウント
- Oracleバックアップをカタログ化
- Oracleバックアップのカタログから削除
- ポイントインタイムリカバリを実行するためにマウントに必要なすべてのバックアップを取得します。

クローン

APIを使用してさまざまな処理を実行できます。

- Oracleデータベースのクローン仕様ファイルの作成、表示、変更、削除
- Oracleデータベースのクローン階層を表示します。
- クローンの詳細を取得
- すべてのクローンを取得
- クローンの削除
- IDによるクローンの詳細の取得
- Oracleリソースのクローン更新ジョブをオンデマンドで開始する
- クローン仕様ファイルを使用して、バックアップからOracleリソースをクローニングする

クローンスプリット

APIを使用してさまざまな処理を実行できます。

- クローンリソースのクローンスプリット処理を見積もります。
- クローンスプリット処理のステータスの取得
- クローンスプリット処理の開始または停止

リソースグループ

APIを使用してさまざまな処理を実行できます。

- すべてのリソースグループの詳細を取得する
- リソースグループを名前で取得
- カスタムアプリケーション用プラグインのリソースグループを作成する
- Plug-in for Microsoft SQL Serverのリソースグループを作成する
- Oracleデータベース用プラグインのリソースグループを作成する
- カスタムアプリケーション用プラグインのリソースグループを変更する
- Plug-in for Microsoft SQL Serverのリソースグループを変更する
- Plug-in for Oracle Databaseのリソースグループを変更する
- Plug-in for Microsoft SQL Serverのリソースグループのクローンライフサイクルを作成、変更、削除する
- リソースグループのバックアップ
- リソースグループをメンテナンスモードまたは本番モードにする
- リソースグループを削除する

ポリシー

APIを使用してさまざまな処理を実行できます。

- ポリシーの詳細を取得
- 名前によるポリシーの詳細の取得
- ポリシーを削除する
- 既存のポリシーのコピーを作成する
- カスタムアプリケーション用プラグインのポリシーを作成または変更する
- Plug-in for Microsoft SQL Serverのポリシーを作成または変更する
- Plug-in for Oracle Database用のポリシーの作成または変更
- Plug-in for SAP HANA Databaseのポリシーを作成または変更する

ストレージ

APIを使用してさまざまな処理を実行できます。

- すべての共有を取得
- 名前による共有の取得
- 共有を作成または削除する
- ストレージの詳細を取得
- 名前によるストレージの詳細の取得
- ストレージの作成、変更、削除
- ストレージクラスタ上のリソースを検出する
- ストレージクラスタ上のリソースの取得

共有

APIを使用してさまざまな処理を実行できます。

- 共有の詳細を取得する
- すべての共有の詳細を取得する
- ストレージ上の共有を作成または削除する
- 名前による共有の取得

プラグイン

APIを使用してさまざまな処理を実行できます。

- ホストのすべてのプラグインを一覧表示する
- キーを使用したMicrosoft SQL Serverリソースの取得
- キーを使用してカスタムリソースを変更する
- キーを使用してカスタムリソースを削除する
- キーを使用したSAP HANAリソースの取得
- キーを使用してSAP HANAリソースを変更する
- キーを使用してSAP HANAリソースを削除する
- キーを使用したOracleリソースの取得
- キーを使用してOracleアプリケーションボリュームリソースを変更する
- キーを使用してOracleアプリケーションボリュームリソースを削除する
- Microsoft SQL Server用プラグインとキーを使用してMicrosoft SQL Serverリソースをバックアップする
- Plug-in for Oracle Databaseとキーを使用してOracleリソースをバックアップする
- カスタムアプリケーション用のプラグインとキーを使用して、カスタムアプリケーションリソースをバックアップする

- キーを使用してSAP HANAデータベースを設定
- キーを使用してOracleデータベースを設定する
- キーを使用したカスタムアプリケーションのバックアップのリストア
- カスタムプラグインリソースを作成する
- SAP HANAリソースを作成する
- Oracleアプリケーションボリュームリソースを作成する
- カスタムアプリケーション用のプラグインを使用してカスタムリソースを保護
- Plug-in for Microsoft SQL Serverを使用したMicrosoft SQL Serverリソースの保護
- 保護されたMicrosoft SQL Serverリソースを変更する
- Microsoft SQL Serverリソースの保護の解除
- Plug-in for Oracle Databaseを使用したOracleリソースの保護
- 保護されているOracleリソースを変更する
- Oracleリソースの保護の解除
- カスタムアプリケーション用プラグインを使用したバックアップからのリソースのクローニング
- Plug-in for Oracle Databaseを使用したバックアップからのOracleアプリケーションボリュームのクローニング
- Plug-in for Microsoft SQL Serverを使用したバックアップからのMicrosoft SQL Serverリソースのクローニング
- Microsoft SQL Serverリソースのクローンライフサイクルを作成する
- Microsoft SQL Serverリソースのクローンライフサイクルを変更する
- Microsoft SQL Serverリソースのクローンライフサイクルを削除する
- Oracleデータベースのクローン仕様ファイルの作成
- Oracleリソースのクローンライフサイクルをオンデマンドで開始
- クローン仕様ファイルを使用して、バックアップからOracleリソースをクローニングする

レポート

APIを使用してさまざまな処理を実行できます。

- それぞれのプラグインのバックアップ、リストア、クローニング処理のレポートを取得する
- スケジュールの追加、実行、削除、変更
- スケジュール済みレポートのデータを取得する

アラート

APIを使用してさまざまな処理を実行できます。

- すべてのアラートを取得
- IDによるアラートの取得

- 複数のアラートの削除またはIDによるアラートの削除

RBAC

APIを使用してさまざまな処理を実行できます。

- ユーザ、グループ、およびロールの詳細を取得する
- ユーザの追加または削除
- ロールへのユーザの割り当て
- ロールからユーザの割り当てを解除
- ロールの作成、変更、削除
- ロールへのグループの割り当て
- ロールからのグループの割り当て解除
- グループの追加または削除
- 既存のロールのコピーを作成する
- ユーザまたはグループへのリソースの割り当てまたは割り当て解除

構成

APIを使用してさまざまな処理を実行できます。

- 構成設定の表示
- 設定の変更

証明書の設定

APIを使用してさまざまな処理を実行できます。

- SnapCenterサーバまたはプラグインホストの証明書ステータスの表示
- SnapCenterサーバまたはプラグインホストの証明書設定を変更する

リポジトリ

APIを使用してさまざまな処理を実行できます。

- リポジトリバックアップの取得
- リポジトリに関する設定情報を表示する
- SnapCenterリポジトリの保護とリストア
- SnapCenterリポジトリの保護を解除する
- リポジトリの再構築とフェイルオーバー

バージョン

このAPIを使用してSnapCenterのバージョンを確認できます。

Swagger API Webページを使用してREST APIにアクセスする方法

REST APIはSwagger Webページから利用できます。Swagger Web ページにアクセスして SnapCenter サーバ REST API を表示したり、API を手動で問題呼び出ししたりできます。REST API を使用して、SnapCenter サーバの管理やデータ保護処理を行うことができます。

REST APIを実行するSnapCenterサーバの管理IPアドレスまたはドメイン名を確認しておく必要があります。

REST APIクライアントを実行するために特別な権限は必要ありません。すべてのユーザがSwagger Webページにアクセスできます。REST API経由でアクセスされるオブジェクトに対する権限は、REST APIにログインするためのトークンを生成したユーザに基づきます。

手順

1. ブラウザで、「\ `https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/swagger/`」の形式でWebページにアクセスするためのURLを入力します。



REST API URL に、+、.、%、& の文字が含まれていないことを確認してください。

2. SwaggerのExplore *フィールドに、Swagger APIドキュメントが自動的に表示されない場合は、次のように入力します。`https://<SnapCenter_IP_address_or_name>:<SnapCenter_port>/Content/swagger/SnapCenter.yaml`
3. [* Explore] をクリックします。

APIリソースタイプまたはカテゴリのリストが表示されます。

4. APIリソースタイプをクリックすると、そのリソースタイプのAPIが表示されます。

SnapCenter REST API の実行時に予期しない動作が発生した場合は、ログファイルを使用して原因を特定し、問題を解決することができます。SnapCenter ユーザー・インターフェイスからログ・ファイルをダウンロードするには、* Monitor * > * Logs * > * Download * をクリックします。

REST APIの使用を開始する

SnapCenter REST APIはすぐに使用を開始できます。APIにアクセスすると、ライブセットアップの複雑なワークフロープロセスでAPIを使用する前に、ある程度の情報を確認できます。

Hello world

システムで簡単なコマンドを実行して、SnapCenter REST APIの使用を開始し、使用可能かどうかを確認できます。

開始する前に

- システムでCurlユーティリティが使用可能であることを確認します。
- SnapCenterサーバのIPアドレスまたはホスト名
- SnapCenter REST APIにアクセスする権限を持つアカウントのユーザ名とパスワード。



クレデンシャルに特殊文字が含まれている場合は、使用しているシェルに基づいてCurlが許容できる形式にする必要があります。たとえば、各特殊文字の前にバックスラッシュを挿入したり、文字列全体を一重引用符で囲むことができます `username:password`。

ステップ

コマンドラインインターフェイスで、次のコマンドを実行してプラグイン情報を取得します。

```
curl -X GET -u username:password -k  
"https://<ip_address>/api/hosts?fields=IncludePluginInfo"
```

例：

```
curl -X GET -u admin:password -k  
"'https://10.225.87.97/api/hosts?fields=IncludePluginInfo'"
```

著作権に関する情報

Copyright © 2024 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。