



REST 구현 세부 정보

ONTAP automation

NetApp
January 12, 2026

This PDF was generated from https://docs.netapp.com/ko-kr/ontap-automation/rest/operational_characteristics.html on January 12, 2026. Always check docs.netapp.com for the latest.

목차

REST 구현 세부 정보	1
ONTAP REST API의 작동 특성입니다	1
요청 및 응답 API 트랜잭션	1
CRUD 작업 지원	1
오브젝트 식별자	1
개체 인스턴스 및 컬렉션	1
동기 및 비동기 작업	1
보안	2
ONTAP REST API 요청에 대한 입력 변수입니다	2
HTTP 메서드	3
경로 변수	3
요청 헤더	3
요청 본문	3
오브젝트 필터링	4
특정 객체 필드를 요청하는 중입니다	4
출력 집합에서 오브젝트 정렬	5
컬렉션의 개체를 검색할 때 페이지 매김	5
크기 속성	6
ONTAP REST API 응답을 해석합니다	6
HTTP 상태 코드입니다	7
응답 헤더	7
응답 바디	7
HAL 연결	8
오류	8
ONTAP REST API를 통한 비동기 처리	9
요청 처리 방법 제어	9
API 요청과 관련된 작업 객체를 쿼리합니다	9
비동기 요청을 실행하는 일반 절차입니다	9
ONTAP REST API 개체 참조 및 액세스	10
오브젝트 액세스 경로	10
UUID를 사용하여 개체에 액세스	10
개체 속성을 사용하여 개체에 액세스	10
클러스터 및 SVM 컨텍스트 비교	11
패치 사용 및 개체 모음에서 삭제	11
ONTAP REST API를 통해 성능 메트릭에 액세스할 수 있습니다	11

REST 구현 세부 정보

ONTAP REST API의 작동 특성입니다

REST는 일반적인 기술과 모범 사례를 설정하지만 각 API의 세부 사항은 설계 선택에 따라 달라질 수 있습니다.

요청 및 응답 API 트랜잭션

모든 REST API 호출은 클라이언트에 대한 관련 응답을 생성하는 ONTAP 시스템에 대한 HTTP 요청으로 수행됩니다. 이 요청/응답 쌍은 API 트랜잭션으로 간주됩니다. API를 사용하기 전에 요청 및 응답 출력의 내용을 제어하는 데 사용할 수 있는 입력 변수에 대해 잘 알고 있어야 합니다.

CRUD 작업 지원

ONTAP REST API를 통해 사용 가능한 각 리소스는 CRUD 모델을 기반으로 액세스됩니다.

- 생성
- 읽기
- 업데이트
- 삭제

일부 리소스의 경우 일부 작업만 지원됩니다. 각 리소스에 대한 자세한 내용은 ONTAP 클러스터의 ONTAP API 설명서 페이지를 참조하십시오.

오브젝트 식별자

각 리소스 인스턴스 또는 개체는 만들 때 고유한 식별자가 할당됩니다. 대부분의 경우 식별자는 128비트 UUID입니다. 이러한 식별자는 특정 ONTAP 클러스터 내에서 전역적으로 고유합니다. 새 개체 인스턴스를 만드는 API 호출을 실행하면 연결된 ID 값이 있는 URL이 HTTP 응답의 위치 헤더에 있는 호출자에게 반환됩니다. 식별자를 추출하여 리소스 인스턴스를 참조할 때 후속 호출에 사용할 수 있습니다.



개체 식별자의 내용 및 내부 구조는 언제든지 변경할 수 있습니다. 관련 객체를 참조할 때는 필요한 경우 해당 API 호출에서만 식별자를 사용해야 합니다.

개체 인스턴스 및 컬렉션

API 호출은 리소스 경로 및 HTTP 메서드에 따라 특정 개체 인스턴스 또는 개체 컬렉션에 적용될 수 있습니다.

동기 및 비동기 작업

ONTAP는 클라이언트로부터 받은 HTTP 요청을 수행하는 두 가지 방법이 있습니다.

동기 처리

ONTAP는 요청을 즉시 수행하고 HTTP 상태 코드가 200 또는 201인 경우 응답한다.

`get`, `head` 및 `options` 메서드를 사용하는 모든 요청은 항상 동기적으로 수행됩니다. 또한 `POST`, `패치` 및 `삭제`를 사용하는 요청은 2초 이내에 완료될 것으로 예상되는 경우 동기적으로 실행되도록 설계되었습니다.

비동기 처리

비동기 요청이 유효한 경우 ONTAP는 요청을 처리하기 위한 백그라운드 작업과 작업을 고정하기 위한 작업 개체를 만듭니다. 202 HTTP 상태는 작업 객체와 함께 호출자에게 반환됩니다. 최종 성공 또는 실패를 확인하려면 작업의 상태를 검색해야 합니다.

`POST`, `패치` 및 `삭제` 메서드를 사용하는 요청은 완료하는 데 2초 이상 걸릴 것으로 예상되는 경우 비동기적으로 실행되도록 설계되었습니다.



를 클릭합니다 `return_timeout` 쿼리 매개 변수는 비동기 API 호출에 사용할 수 있으며 비동기 호출을 동기적으로 완료하도록 변환할 수 있습니다. 을 참조하십시오 ["작업 개체를 사용한 비동기 처리"](#)를 참조하십시오.

보안

REST API와 함께 제공되는 보안은 주로 ONTAP에서 사용할 수 있는 기존 보안 기능을 기반으로 합니다. 다음 보안은 API에서 사용됩니다.

전송 계층 보안

클라이언트와 ONTAP LIF 간에 네트워크를 통해 전송되는 모든 트래픽은 ONTAP 구성 설정에 따라 TLS를 사용하여 암호화됩니다.

클라이언트 인증

ONTAP System Manager 및 네트워크 관리 SDK에서 사용할 수 있는 인증 옵션과 동일한 인증 옵션을 ONTAP REST API에서도 사용할 수 있습니다.

HTTP 인증

예를 들어, HTTP 레벨에서 ONTAP REST API에 직접 액세스하는 경우 아래에 설명된 대로 두 가지 인증 옵션이 있습니다. 각 경우에 HTTP 권한 부여 헤더를 만들어 각 요청에 포함시켜야 합니다.

옵션을 선택합니다	설명
HTTP 기본 인증	ONTAP 사용자 이름과 암호는 콜론으로 연결됩니다. 문자열은 base64로 변환되고 요청 헤더에 포함됩니다.
OAuth 2.0 을 참조하십시오	ONTAP 9.14부터 외부 권한 부여 서버에서 액세스 토큰을 요청하여 요청 헤더에 베어러 토큰으로 포함할 수 있습니다.

OAuth 2.0에 대한 자세한 내용과 ONTAP에서 구현하는 방법은 을 참조하십시오 ["ONTAP OAuth 2.0 구축 개요"](#). 도 참조하십시오 ["워크플로우 사용을 준비하십시오"](#) 이 사이트의 아래.

ONTAP 인증

ONTAP는 역할 기반 인증 모델을 구현합니다. ONTAP REST API 또는 API 설명서 페이지에 액세스할 때 사용하는 계정에 적절한 권한이 있어야 합니다.

ONTAP REST API 요청에 대한 입력 변수입니다

HTTP 요청에 설정된 매개 변수와 변수를 통해 API 호출이 처리되는 방식을 제어할 수 있습니다.

HTTP 메서드

ONTAP REST API에서 지원하는 HTTP 메서드는 다음 표에 나와 있습니다.



모든 HTTP 메서드를 각 REST 끝점에서 사용할 수 있는 것은 아닙니다. 또한 컬렉션에서 패치와 삭제를 모두 사용할 수 있습니다. 자세한 내용은 [_개체 참조 및 액세스_](#)를 참조하십시오.

HTTP 메소드	설명
가져오기	리소스 인스턴스 또는 컬렉션의 개체 속성을 검색합니다.
게시	제공된 입력을 기반으로 새 리소스 인스턴스를 만듭니다.
패치	제공된 입력에 따라 기존 리소스 인스턴스를 업데이트합니다.
삭제	기존 리소스 인스턴스를 삭제합니다.
머리	GET 요청을 효과적으로 발급하지만 HTTP 헤더만 반환합니다.
옵션	특정 끝점에서 지원되는 HTTP 메서드를 결정합니다.

경로 변수

각 REST API 호출에 사용되는 끝점 경로에는 다양한 식별자가 포함될 수 있습니다. 각 ID는 특정 리소스 인스턴스에 해당합니다. 예를 들어 클러스터 ID 및 SVM ID가 있습니다.

요청 헤더

HTTP 요청에 여러 개의 헤더를 포함해야 합니다.

콘텐츠 유형

요청 본문에 JSON이 포함된 경우 이 헤더를 `application/json`으로 설정해야 합니다.

수락

이 헤더는 `application/hal+json`으로 설정해야 한다. 대신 `application/json`으로 설정하면 다음 레코드 배치를 검색하는 데 필요한 링크를 제외하고 HAL 링크는 반환되지 않습니다. 이 두 값 외에 다른 머리글이 있으면 응답에서 콘텐츠 형식 헤더의 기본값은 `application/hal+json`이 됩니다.

권한 부여

기본 인증은 `base64` 문자열로 인코딩된 사용자 이름과 암호로 설정되어야 합니다. 예를 들면 다음과 같습니다.

```
Authorization: Basic YWRtaW46cGV0ZXJzb24=.
```

요청 본문

요청 본문의 내용은 특정 호출에 따라 달라집니다. HTTP 요청 본문은 다음 중 하나로 구성됩니다.

- 입력 변수가 있는 JSON 개체입니다
- JSON 개체가 비어 있습니다

오브젝트 필터링

Get 메서드를 사용하여 API 호출을 실행할 때 쿼리 매개 변수를 사용하여 모든 특성을 기준으로 반환된 객체를 제한하거나 필터링할 수 있습니다.

쿼리 매개 변수 구문 분석 및 해석

하나 이상의 매개 변수 집합을 뒤에 오는 URL 문자열에 추가할 수 있습니다 ? 문자. 둘 이상의 매개 변수가 제공된 경우 쿼리 매개 변수는 을 기반으로 분할됩니다 & 문자. 매개 변수의 각 키와 값은 에서 분할됩니다 = 문자.

예를 들어 등호를 사용하여 정확히 일치하는 값을 지정할 수 있습니다.

<field>=<value>

보다 복잡한 쿼리의 경우 추가 연산자는 등호 뒤에 배치됩니다. 예를 들어 일부 값보다 크거나 같은 특정 필드를 기준으로 개체 집합을 선택하려면 쿼리는 다음과 같습니다.

<field>=>=<value>

필터링 연산자

위에 제공된 예제 외에 값 범위에 걸쳐 개체를 반환하는 추가 연산자를 사용할 수 있습니다. ONTAP REST API가 지원하는 필터링 연산자에 대한 요약은 아래 표와 같다.



설정되지 않은 필드는 일반적으로 일치하는 쿼리에서 제외됩니다.

운영자	설명
=	같음
를 누릅니다	보다 작음
를 누릅니다	보다 큼
=	보다 작거나 같음
>=	보다 크거나 같음
!	같지 않음
*	greedy 와일드카드

쿼리의 일부로 "null" 키워드 또는 "Negation"!null"을 사용하여 특정 필드가 설정되었는지 여부를 기준으로 개체 컬렉션을 반환할 수도 있습니다.

워크플로 예

이 사이트의 REST API 워크플로에 포함된 몇 가지 예가 아래에 나와 있습니다.

- ["디스크 나열"](#)

를 기준으로 필터링합니다 state 스페어 디스크를 선택하는 변수입니다.

특정 객체 필드를 요청하는 중입니다

기본적으로 Get을 사용하여 API 호출을 실행하면 HAL 자체 링크와 함께 개체나 개체를 고유하게 식별하는 특성만

반환됩니다. 이 최소 필드 집합은 각 개체의 키 역할을 하며 개체 유형에 따라 달라집니다. 다음과 같은 방법으로 '필드' 쿼리 매개 변수를 사용하여 추가 개체 속성을 선택할 수 있습니다.

- **공통 또는 표준 필드**

가장 일반적으로 사용되는 객체 필드를 검색하려면 필드= *를 지정합니다. 이러한 필드는 일반적으로 로컬 서버 메모리에 유지되거나 액세스에 필요한 처리가 거의 필요하지 않습니다. 이 속성은 URL 경로 키(UUID)로 GET을 사용한 후 개체에 대해 반환되는 속성과 동일합니다.

- **모든 필드**

액세스 시 추가 서버 처리가 필요한 필드를 포함하여 모든 객체 필드를 검색하려면 필드=***를 지정합니다.

- **사용자 정의 필드 선택**

필드=<field_name>'을 사용하여 원하는 필드를 정확하게 지정합니다. 여러 필드를 요청할 때는 공백 없이 쉼표를 사용하여 값을 구분해야 합니다.



가장 좋은 방법은 항상 원하는 특정 필드를 식별하는 것입니다. 필요한 경우 공통 필드 또는 모든 필드 집합만 검색해야 합니다. NetApp은 자체 성과 분석을 기반으로 공통으로 분류되어 있으며 'fields= *'를 사용하여 반환되는 필드를 결정합니다. 필드의 분류는 향후 릴리스에서 변경될 수 있습니다.

출력 집합에서 오브젝트 정렬

리소스 컬렉션의 레코드는 개체에서 정의한 기본 순서로 반환됩니다. 다음과 같이 필드 이름 및 정렬 방향을 가진 "ORDER_BY" 쿼리 매개 변수를 사용하여 순서를 변경할 수 있습니다.

'ORDER_BY=<필드 이름>ASC|desc'

예를 들어 유형 필드를 내림차순으로 정렬한 다음 ID를 오름차순으로 정렬할 수 있습니다.

order_by=desc, id asc를 입력합니다

다음 사항에 유의하십시오.

- 정렬 필드를 지정하지만 방향을 지정하지 않으면 값이 오름차순으로 정렬됩니다.
- 여러 매개 변수를 포함할 때는 필드를 쉼표로 구분해야 합니다.

컬렉션의 개체를 검색할 때 페이지 매김

Get 을 사용하여 API 호출을 실행하여 같은 형식의 개체 컬렉션에 액세스하면 ONTAP 는 두 가지 제약 조건에 따라 가능한 한 많은 개체를 반환합니다. 요청에 대한 추가 쿼리 매개 변수를 사용하여 이러한 각 제약 조건을 제어할 수 있습니다. 특정 GET 요청에 대한 첫 번째 제약 조건에 도달하면 요청이 종료되고 반환된 레코드 수가 제한됩니다.



모든 개체를 반복하기 전에 요청이 종료되면 응답에는 다음 레코드 배치를 검색하는 데 필요한 링크가 포함됩니다.

개체 수 제한

기본적으로 ONTAP 는 GET 요청에 대해 최대 10,000개의 오브젝트를 반환합니다. 이 제한은 max_records 쿼리

매개 변수를 사용하여 변경할 수 있습니다. 예를 들면 다음과 같습니다.

max_records=20

실제로 반환되는 개체 수는 관련 시간 제약 조건 및 시스템의 총 개체 수에 따라 실제 최대값보다 작을 수 있습니다.

객체를 검색하는 데 사용되는 시간 제한

기본적으로 ONTAP는 GET 요청에 허용된 시간 내에 가능한 한 많은 오브젝트를 반환합니다. 기본 시간 초과는 15초입니다. RETURN_TIMEOUT' QUERY 파라미터를 이용하여 이 한계를 변경할 수 있다. 예를 들면 다음과 같습니다.

RETURN_TIMEOUT=5'입니다

실제로 반환되는 개체 수는 시스템의 총 개체 수와 개체 수에 대한 관련 제약 조건에 따라 최대 개체수보다 작을 수 있습니다.

결과 집합 축소

필요한 경우 이러한 두 매개 변수를 추가 쿼리 매개 변수와 결합하여 결과 집합의 범위를 좁힐 수 있습니다. 예를 들어, 지정된 시간 이후에 생성된 최대 10개의 EMS 이벤트가 반환됩니다.

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

여러 요청을 발행하여 객체를 통해 페이지를 이동할 수 있습니다. 이후의 각 API 호출은 마지막 결과 집합의 최신 이벤트를 기반으로 새 시간 값을 사용해야 합니다.

크기 속성

일부 API 호출과 특정 쿼리 매개 변수에 사용되는 입력 값은 숫자입니다. 정수(바이트)를 제공하는 대신 다음 표에 나와 있는 접미사를 선택적으로 사용할 수 있습니다.

접미사	설명
KB를 클릭합니다	KB 킬로바이트(1024바이트) 또는 키비바이트
MB	MB 메가바이트(KB x 1024바이트) 또는 메가바이트
GB	GB 기가바이트(MB x 1024바이트) 또는 기비바이트
TB	TB 테라바이트(GB x 1024바이트) 또는 테비바이트
PB	PB 페타바이트(TB x 1024바이트) 또는 페이비바이트

관련 정보

- ["객체 참조 및 액세스"](#)

ONTAP REST API 응답을 해석합니다

각 API 요청은 클라이언트에 대한 응답을 다시 생성합니다. 응답을 검토하여 성공 여부를 확인하고 필요에 따라 추가 데이터를 검색해야 합니다.

HTTP 상태 코드입니다

ONTAP REST API에서 사용하는 HTTP 상태 코드는 다음과 같다.

코드	사유 구문	설명
200	좋습니다	새 개체를 만들지 않는 호출에 대한 성공 여부를 나타냅니다.
201	작성됨	객체가 성공적으로 생성되었습니다. 응답의 위치 헤더에는 개체의 고유 식별자가 포함됩니다.
202	수락됨	요청을 수행하기 위해 백그라운드 작업이 시작되었지만 아직 완료되지 않았습니다.
400	잘못된 요청입니다	요청 입력이 인식되지 않거나 부적절합니다.
401	권한이 없습니다	사용자 인증에 실패했습니다.
403	금지됨	인증 오류로 인해 액세스가 거부되었습니다.
404	찾을 수 없습니다	요청에서 참조되는 리소스가 없습니다.
405	메서드가 허용되지 않습니다	요청의 HTTP 메서드가 리소스에 대해 지원되지 않습니다.
409	충돌	다른 개체를 먼저 만들어야 하거나 요청된 개체가 이미 있으므로 개체를 만들지 못했습니다.
500입니다	내부 오류입니다	서버에서 일반적인 내부 오류가 발생했습니다.

응답 헤더

ONTAP에서 생성된 HTTP 응답에는 여러 헤더가 포함되어 있습니다.

위치

개체를 만들 때 위치 머리글에는 개체에 할당된 고유 식별자를 포함하여 새 개체에 대한 전체 URL이 포함됩니다.

콘텐츠 유형

이것은 보통 '응용 프로그램/ HAL+json'입니다.

응답 바디

API 요청으로 인한 응답 본문의 내용은 객체, 처리 유형 및 요청의 성공 또는 실패에 따라 달라집니다. 응답은 항상 JSON으로 렌더링됩니다.

- 단일 개체

요청에 따라 필드 집합과 함께 단일 개체를 반환할 수 있습니다. 예를 들어, 가져오기를 사용하여 고유 식별자를 사용하여 클러스터의 선택된 속성을 검색할 수 있습니다.

- 여러 개의 개체

리소스 컬렉션의 여러 개체를 반환할 수 있습니다. 모든 경우에 사용되는 형식이 일관되게 있으며, 개체 인스턴스의 배열을 포함하는 레코드 및 레코드 수를 나타내는 num_records가 있습니다. 예를 들어, 특정 클러스터에 정의된 노드를 검색할 수 있습니다.

- 작업 객체

API 호출이 비동기적으로 처리되는 경우 백그라운드 작업을 고정하는 Job 개체가 반환됩니다. 예를 들어 클러스터 구성은 업데이트하는 데 사용되는 패치 요청은 비동기적으로 처리되고 작업 개체를 반환합니다.

- 오류 개체

오류가 발생하면 항상 Error 개체가 반환됩니다. 예를 들어, 클러스터에 대해 정의되지 않은 필드를 변경하려고 하면 오류가 발생합니다.

- JSON 개체가 비어 있습니다

경우에 따라 데이터가 반환되지 않고 응답 본문에 빈 JSON 개체가 포함되는 경우가 있습니다.

HAL 연결

ONTAP REST API는 HAL을 HATEOAS(Engine of Application State)로 Hypermedia를 지원하는 메커니즘으로 사용합니다. 특정 리소스를 식별하는 객체 또는 속성이 반환되면 HAL로 인코딩된 링크도 포함되어 리소스에 대한 추가 세부 정보를 쉽게 찾아 확인할 수 있습니다.

오류

오류가 발생하면 응답 본문에 오류 객체가 반환됩니다.

형식

오류 개체의 형식은 다음과 같습니다.

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

코드 값을 사용하여 일반 오류 유형 또는 범주를 확인하고 메시지를 사용하여 특정 오류를 확인할 수 있습니다. 사용 가능한 경우 대상 필드에는 오류와 관련된 특정 사용자 입력이 포함됩니다.

일반 오류 코드

일반적인 오류 코드는 다음 표에 설명되어 있습니다. 특정 API 호출에는 추가 오류 코드가 포함될 수 있습니다.

코드		설명
1	409	동일한 식별자를 가진 객체가 이미 있습니다.
2	400	필드 값이 잘못되었거나 누락되었거나 추가 필드가 제공되었습니다.
3	400	작업이 지원되지 않습니다.
4	405	지정된 식별자가 있는 개체를 찾을 수 없습니다.

코드		설명
6	403	요청 수행 권한이 거부되었습니다.
8	409	리소스가 사용 중입니다.

ONTAP REST API를 통한 비동기 처리

비동기적으로 실행되도록 설계된 API 요청을 실행하면 작업 객체가 항상 생성되어 호출자에게 반환됩니다. 이 작업에서는 요청을 처리하는 백그라운드 작업을 설명하고 앵커를 지정합니다. HTTP 상태 코드에 따라 작업의 상태를 검색하여 요청이 성공했는지 확인해야 합니다.

을 참조하십시오 ["API 참조입니다"](#) 비동기적으로 수행할 API 호출을 결정합니다.

요청 처리 방법 제어

`RETURN_TIMEOUT` query 파라미터를 사용하면 비동기 API 호출 처리 방식을 제어할 수 있다. 이 매개 변수를 사용할 경우 두 가지 결과를 얻을 수 있습니다.

요청이 완료되기 전에 타이머가 만료됩니다

유효한 요청의 경우 ONTAP는 작업 객체와 함께 202 HTTP 상태 코드를 반환합니다. 요청이 성공적으로 완료되었는지 확인하려면 작업의 상태를 검색해야 합니다.

타이머가 만료되기 전에 요청이 완료되었습니다

요청이 유효하고 시간이 만료되기 전에 성공적으로 완료되면 ONTAP는 작업 객체와 함께 200 HTTP 상태 코드를 반환합니다. 요청이 200으로 표시된 대로 동기적으로 완료되므로 작업 상태를 검색할 필요가 없습니다.



`RETURN_TIMEOUT` 파라미터의 기본값은 0초입니다. 따라서 매개 변수를 포함하지 않으면 유효한 요청에 대해 202 HTTP 상태 코드가 항상 반환됩니다.

API 요청과 관련된 작업 객체를 쿼리합니다

HTTP 응답에서 반환된 작업 개체에는 여러 속성이 포함되어 있습니다. 후속 API 호출에서 `state` 속성을 쿼리하여 요청이 성공적으로 완료되었는지 확인할 수 있습니다. 작업 오브젝트는 항상 다음 상태 중 하나에 있습니다.

비단말기 상태입니다

- 대기열에 있습니다
- 실행 중입니다
- 일시 중지되었습니다

터미널 상태

- 성공
- 실패

비동기 요청을 실행하는 일반 절차입니다

다음 고급 절차를 사용하여 비동기 API 호출을 완료할 수 있습니다. 이 예제에서는 `RETURN_TIMEOUT` 매개변수가

사용되지 않거나 백그라운드 작업이 완료되기 전에 시간이 만료되는 것으로 가정합니다.

1. 비동기적으로 수행하도록 설계된 API 호출을 실행합니다.
2. 유효한 요청 수락을 나타내는 HTTP 응답 202 을 수신합니다.
3. 응답 본문에서 작업 객체의 식별자를 추출합니다.
4. 시간 지정 루프 내에서 각 사이클에서 다음을 수행합니다.
 - a. 작업의 현재 상태를 가져옵니다.
 - b. 작업이 터미널 상태가 아닌 경우 루프를 다시 수행합니다.
5. 작업이 터미널 상태(성공, 실패)에 도달하면 중지합니다.

관련 정보

- ["클러스터 연락처를 업데이트합니다"](#)
- ["작업 인스턴스를 가져옵니다"](#)

ONTAP REST API 개체 참조 및 액세스

ONTAP REST API를 통해 노출된 리소스 인스턴스 또는 개체는 여러 가지 방법으로 참조되고 액세스할 수 있습니다.

오브젝트 액세스 경로

높은 수준에서는 개체에 액세스할 때 두 가지 경로 유형이 있습니다.

- 기본
 개체는 API 호출의 기본 또는 직접 타겟입니다.
- 외부
 개체는 API 호출의 기본 참조가 아니라 기본 개체에서 에 연결됩니다. 따라서 이 개체는 외부 또는 다운스트림 개체이며 기본 개체의 필드를 통해 참조됩니다.

UUID를 사용하여 개체에 액세스

모든 개체에는 생성 시 고유한 식별자가 할당되며, 대부분의 경우 128비트 UUID입니다. 할당된 UUID 값은 변경할 수 없으며 ONTAP 내에서 내부적으로 사용되어 리소스에 액세스하고 관리할 수 있습니다. 따라서 UUID는 일반적으로 개체에 가장 빠르고 안정적으로 액세스할 수 있습니다.

대부분의 리소스 유형에서 UUID 값을 URL의 경로 키의 일부로 제공하여 특정 개체에 액세스할 수 있습니다. 예를 들어, 다음을 사용하여 노드 인스턴스 `"/cluster/nodes/{uuid}"`에 액세스할 수 있습니다

개체 속성을 사용하여 개체에 액세스

UUID 외에도 개체 속성을 사용하여 개체에 액세스할 수도 있습니다. 대부분의 경우 name 속성을 사용하면 편리합니다. 예를 들어, URL 문자열에 있는 다음 쿼리 매개 변수를 사용하여 노드 인스턴스에 대한 이름(`"/cluster/nodes?name=node_one"`)을 기준으로 액세스할 수 있습니다. 쿼리 매개 변수 외에도 기본 개체의

속성을 통해 외부 개체에 액세스할 수 있습니다.

UUID 대신 이름이나 기타 속성을 사용하여 개체에 액세스할 수 있지만 다음과 같은 몇 가지 단점이 있습니다.

- 이름 필드는 변경할 수 없으며 변경할 수 있습니다. 개체에 액세스하기 전에 개체의 이름이 변경되면 잘못된 개체가 반환되거나 개체 액세스 오류가 발생합니다.



이 문제는 외부 개체의 POST 또는 패치 메서드나 기본 개체의 get 메서드에서 발생할 수 있습니다.

- ONTAP는 이름 필드를 해당 UUID로 변환해야 합니다. 이는 성능 문제가 될 수 있는 일종의 간접 액세스입니다.

특히 다음 중 하나 이상이 참일 경우 성능 저하가 발생할 수 있습니다.

- Get 메서드가 사용됩니다
- 큰 오브젝트 집합에 액세스합니다
- 복잡하거나 복잡한 쿼리가 사용됩니다

클러스터 및 SVM 컨텍스트 비교

클러스터와 SVM을 모두 지원하는 여러 개의 REST 엔드포인트가 있습니다. 이러한 엔드포인트 중 하나를 사용할 때 'scope=[svm|cluster]' 값을 통해 API 호출 컨텍스트를 나타낼 수 있습니다. 이중 컨텍스트를 지원하는 엔드포인트의 예로는 IP 인터페이스 및 보안 역할이 있습니다.



범위 값에는 각 API 호출에 대해 제공된 속성을 기반으로 하는 기본값이 있습니다.

패치 사용 및 개체 모음에서 삭제

리소스 인스턴스에서 패치나 삭제를 지원하는 모든 REST 끝점은 개체 컬렉션에서 동일한 메서드를 지원합니다. 유일한 요구 사항은 URL 문자열의 쿼리 매개 변수를 통해 하나 이상의 필드를 제공해야 한다는 것입니다. 컬렉션을 통해 패치 또는 삭제를 실행하는 경우 이는 내부적으로 다음을 수행하는 것과 같습니다.

- 쿼리 기반 가져오기 - 컬렉션을 검색합니다
- 컬렉션의 각 개체에 대한 패치 또는 삭제 호출의 일련 번호 시퀀스입니다

작업 제한 시간은 기본값 15초로 RETURN_TIMEOUT으로 설정할 수 있다. 시간 초과 전에 완료되지 않은 경우 응답에는 다음 객체에 대한 링크가 포함됩니다. 작업을 계속하려면 다음 링크를 사용하여 동일한 HTTP 메서드를 다시 실행해야 합니다.

ONTAP REST API를 통해 성능 메트릭에 액세스할 수 있습니다

ONTAP은 선택한 SVM 스토리지 오브젝트 및 프로토콜에 대한 성능 메트릭을 수집하고 REST API를 통해 이 정보를 보고합니다. 이 데이터를 사용하여 ONTAP 시스템의 성능을 모니터링할 수 있습니다.

특정 스토리지 오브젝트 또는 프로토콜에서 성능 데이터는 다음 세 가지 범주로 분류됩니다.

- IOPS

- 지연 시간

- 처리량

각 범주 내에서 다음 데이터 유형 중 하나 이상을 사용할 수 있습니다.

- 읽기®
- 쓰기(W)
- 기타(O)
- 합계(T)

다음 표에는 ONTAP REST API를 통해 사용할 수 있는 성능 데이터와 추가된 릴리즈가 요약되어 있습니다. 자세한 내용은 ONTAP 시스템의 REST API 온라인 설명서 페이지를 참조하십시오.

스토리지 오브젝트 또는 프로토콜입니다	IOPS	지연 시간	처리량	ONTAP 릴리즈
이더넷 포트	해당 없음	해당 없음	RWT	9.8
FC 포트	RWOT	RWOT	RWT	9.8
IP 인터페이스	해당 없음	해당 없음	RWT	9.8
FC 인터페이스	RWOT	RWOT	RWT	9.8
NVMe 네임스페이스	RWOT	RWOT	RWOT	9.8
qtree 통계	원시 RWOT	해당 없음	원시 RWOT	9.8
Volume FlexCache(볼륨 저장)	RWOT	RWOT	RWT	9.8
노드 - 프로세스 활용도	수치 값으로서 프로세스 활용	수치 값으로서 프로세스 활용	수치 값으로서 프로세스 활용	9.8
클라우드 볼륨	RWOT	RWOT	유연하지 않습니다	9.7
LUN을 클릭합니다	RWOT	RWOT	RWOT	9.7
집계	RWOT	RWOT	RWOT	9.7
SVM NFS 프로토콜	RWOT	RWOT	RWT	9.7
SVM CIFS 프로토콜	RWOT	RWOT	RWT	9.7
SVM FCP 프로토콜	RWOT	RWOT	RWT	9.7
SVM iSCSI 프로토콜	RWOT	RWOT	RWT	9.7
SVM NVMe 프로토콜	RWOT	RWOT	RWT	9.7
클러스터	RWOT	RWOT	RWOT	9.6
볼륨	RWOT	RWOT	RWOT	9.6

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.