



# **Dettagli sull'implementazione REST**

## **ONTAP automation**

NetApp  
January 12, 2026

This PDF was generated from [https://docs.netapp.com/it-it/ontap-automation/rest/operational\\_characteristics.html](https://docs.netapp.com/it-it/ontap-automation/rest/operational_characteristics.html) on January 12, 2026. Always check docs.netapp.com for the latest.

# Sommario

Dettagli sull'implementazione REST .....	1
Caratteristiche operative dell'API REST ONTAP .....	1
Transazione API di richiesta e risposta .....	1
Supporto per le operazioni CRUD .....	1
Identificatori di oggetti .....	1
Istanze e raccolte di oggetti .....	1
Operazioni sincrone e asincrone .....	1
Sicurezza .....	2
Variabili di input per una richiesta API REST ONTAP .....	3
Metodi HTTP .....	3
Variabili di percorso .....	3
Intestazioni delle richieste .....	3
Corpo della richiesta .....	4
Filtraggio degli oggetti .....	4
Richiesta di campi oggetto specifici .....	5
Ordinamento degli oggetti nel set di output .....	5
Impaginazione durante il recupero di oggetti in una raccolta .....	6
Proprietà delle dimensioni .....	6
Interpretare una risposta dell'API REST ONTAP .....	7
Codice di stato HTTP .....	7
Intestazioni delle risposte .....	7
Corpo di risposta .....	8
Collegamento HAL .....	8
Errori .....	8
Elaborazione asincrona con l'API REST ONTAP .....	9
Controllo della modalità di elaborazione di una richiesta .....	9
Esecuzione di query sull'oggetto Job associato a una richiesta API .....	10
Procedura generale per l'emissione di una richiesta asincrona .....	10
Riferimenti e accesso agli oggetti dell'API REST di ONTAP .....	10
Percorsi di accesso a oggetti .....	10
Accesso a un oggetto utilizzando l'UUID .....	11
Accesso a un oggetto mediante una proprietà Object .....	11
Confronto tra cluster e contesto SVM .....	11
Utilizzo DI PATCH ed ELIMINAZIONE su un insieme di oggetti .....	12
Accedi alle metriche delle performance con l'API REST di ONTAP .....	12

# Dettagli sull'implementazione REST

## Caratteristiche operative dell'API REST ONTAP

Mentre REST stabilisce un insieme comune di tecnologie e Best practice, i dettagli di ciascuna API possono variare in base alle scelte di progettazione.

### Transazione API di richiesta e risposta

Ogni chiamata API REST viene eseguita come richiesta HTTP al sistema ONTAP che genera una risposta associata al client. Questa coppia di richieste/risposte è considerata una transazione API. Prima di utilizzare l'API, è necessario conoscere le variabili di input disponibili per controllare una richiesta e il contenuto dell'output della risposta.

### Supporto per le operazioni CRUD

Si accede a ciascuna delle risorse disponibili tramite l'API REST ONTAP in base al modello CRUD:

- Creare
- Leggere
- Aggiornare
- Eliminare

Per alcune delle risorse, è supportato solo un sottoinsieme delle operazioni. Per ulteriori informazioni su ciascuna risorsa, consultare la pagina della documentazione relativa alle API ONTAP nel cluster ONTAP.

### Identificatori di oggetti

A ogni istanza o oggetto di risorsa viene assegnato un identificatore univoco al momento della creazione. Nella maggior parte dei casi, l'identificatore è un UUID a 128 bit. Questi identificatori sono univoci a livello globale all'interno di un cluster ONTAP specifico. Dopo aver eseguito una chiamata API che crea una nuova istanza di oggetto, viene restituito al chiamante un URL con il valore id associato nell'intestazione di posizione della risposta HTTP. È possibile estrarre l'identificatore e utilizzarlo nelle chiamate successive quando si fa riferimento all'istanza della risorsa.



Il contenuto e la struttura interna degli identificatori di oggetti possono cambiare in qualsiasi momento. È necessario utilizzare gli identificatori delle chiamate API applicabili solo se necessario quando si fa riferimento agli oggetti associati.

### Istanze e raccolte di oggetti

A seconda del percorso di risorsa e del metodo HTTP, una chiamata API può essere applicata a un'istanza di oggetto specifica o a un insieme di oggetti.

### Operazioni sincrone e asincrone

Esistono due modi in cui ONTAP esegue una richiesta HTTP ricevuta da un client.

## Elaborazione sincrona

ONTAP esegue immediatamente la richiesta e risponde con un codice di stato HTTP 200 o 201 se l'operazione ha esito positivo.

Ogni richiesta che utilizza i metodi GET, HEAD e OPTIONS viene sempre eseguita in maniera sincrona. Inoltre, le richieste che utilizzano POST, PATCH e DELETE sono progettate per essere eseguite in modo sincrono se si prevede che vengano completate in meno di due secondi.

## Elaborazione asincrona

Se una richiesta asincrona è valida, ONTAP crea un'attività in background per elaborare la richiesta e un oggetto di lavoro per ancorare l'attività. Lo stato 202 HTTP viene restituito al chiamante insieme all'oggetto lavoro. Per determinare l'esito positivo o negativo finale, è necessario recuperare lo stato del lavoro.

Le richieste che utilizzano i metodi POST, PATCH ed ELIMINAZIONE sono progettate per essere eseguite in modo asincrono se si prevede che il completamento delle richieste richiederà più di due secondi.



Il `return_timeout` Il parametro query è disponibile con le chiamate API asincrone e può convertire una chiamata asincrona in modo sincrono. Fare riferimento a ["Elaborazione asincrona utilizzando l'oggetto Job"](#) per ulteriori informazioni.

## Sicurezza

La sicurezza fornita con L'API REST si basa principalmente sulle funzionalità di sicurezza esistenti disponibili con ONTAP. L'API utilizza la seguente protezione:

### Transport Layer Security

Tutto il traffico inviato sulla rete tra il client e la LIF ONTAP viene in genere crittografato con TLS, in base alle impostazioni di configurazione della ONTAP.

### Autenticazione del client

Le stesse opzioni di autenticazione disponibili con Gestore di sistema di ONTAP e SDK per la gestione della rete possono essere utilizzate anche con l'API REST di ONTAP.

### Autenticazione HTTP

A livello HTTP, ad esempio quando si accede direttamente all'API REST ONTAP, sono disponibili due opzioni di autenticazione, come descritto di seguito. In ogni caso, è necessario creare un'intestazione di autorizzazione HTTP e includerla con ogni richiesta.

Opzione	Descrizione
Autenticazione di base HTTP	Il nome utente e la password di ONTAP sono concatenati con due punti. La stringa viene convertita in base64 e inclusa nell'intestazione della richiesta.
OAuth 2,0	A partire da ONTAP 9,14, è possibile richiedere un token di accesso da un server di autorizzazione esterno e includerlo come token portante nell'intestazione della richiesta.

Per ulteriori informazioni su OAuth 2,0 e su come viene implementato in ONTAP, vedere ["Panoramica dell'implementazione di ONTAP OAuth 2,0"](#). Vedere anche ["Preparati a utilizzare i flussi di lavoro"](#) qui sotto a questo luogo.

## Autorizzazione ONTAP

ONTAP implementa un modello di autorizzazione basato sui ruoli. L'account utilizzato per accedere alla pagina di documentazione API O API REST di ONTAP deve disporre dell'autorità appropriata.

## Variabili di input per una richiesta API REST ONTAP

È possibile controllare la modalità di elaborazione di una chiamata API attraverso parametri e variabili impostati nella richiesta HTTP.

### Metodi HTTP

I metodi HTTP supportati dall'API REST di ONTAP sono illustrati nella seguente tabella.



Non tutti i metodi HTTP sono disponibili in ogni endpoint REST. Inoltre, È possibile utilizzare PATCH e DELETE in una raccolta. Per ulteriori informazioni, consulta la sezione *referimenti a oggetti e accesso*.

Metodo HTTP	Descrizione
OTTIENI	Recupera le proprietà dell'oggetto su un'istanza o una raccolta di risorse.
POST	Crea una nuova istanza di risorsa in base all'input fornito.
PATCH	Aggiorna un'istanza di risorsa esistente in base all'input fornito.
ELIMINARE	Elimina un'istanza di risorsa esistente.
TESTA	Emette in modo efficace una richiesta GET ma restituisce solo le intestazioni HTTP.
OPZIONI	Determinare quali metodi HTTP sono supportati in un endpoint specifico.

### Variabili di percorso

Il percorso dell'endpoint utilizzato con ogni chiamata API REST può includere vari identificatori. Ciascun ID corrisponde a una specifica istanza di risorsa. Alcuni esempi comprendono ID del cluster e ID SVM.

### Intestazioni delle richieste

È necessario includere diverse intestazioni nella richiesta HTTP.

#### Tipo di contenuto

Se il corpo della richiesta include JSON, questa intestazione deve essere impostata su `application/json`.

#### Accettare

Questa intestazione deve essere impostata su `application/hal+json`. Se invece è impostato su `application/json` Nessuno dei collegamenti HAL verrà restituito, ad eccezione di un collegamento necessario per recuperare il successivo batch di record. Se l'intestazione è diversa da questi due valori, il valore predefinito di `content-type` l'intestazione nella risposta sarà `application/hal+json`.

#### Autorizzazione

L'autenticazione di base deve essere impostata con il nome utente e la password codificati come stringa base64. Ad esempio:

## Corpo della richiesta

Il contenuto del corpo della richiesta varia in base alla chiamata specifica. Il corpo della richiesta HTTP è costituito da uno dei seguenti elementi:

- Oggetto JSON con variabili di input
- Oggetto JSON vuoto

## Filtraggio degli oggetti

Quando si esegue una chiamata API con il metodo GET, è possibile limitare o filtrare gli oggetti restituiti in base a qualsiasi attributo utilizzando un parametro di query.

### Analisi e interpretazione dei parametri di query

Una serie di uno o più parametri può essere aggiunta alla stringa URL che inizia dopo ? carattere. Se viene fornito più di un parametro, i parametri di query vengono suddivisi in base a. & carattere. Ciascun parametro e valore nel parametro vengono suddivisi in = carattere.

Ad esempio, è possibile specificare un valore esatto da corrispondere utilizzando il segno di uguale:

`<field>=<value>`

Per una query più complessa, l'operatore aggiuntivo viene posizionato dopo il segno di uguale. Ad esempio, per selezionare l'insieme di oggetti in base a un campo specifico maggiore o uguale a un valore, la query sarà:

`<field>=>=<value>`

### Filtraggio degli operatori

Oltre agli esempi forniti in precedenza, sono disponibili operatori aggiuntivi per restituire oggetti su un intervallo di valori. Nella tabella seguente è riportato un riepilogo degli operatori di filtraggio supportati dall'API REST ONTAP.



Tutti i campi non impostati sono generalmente esclusi dalle query corrispondenti.

Operatore	Descrizione
=	Uguale a.
<	Inferiore a.
>	Maggiore di
≤	Minore o uguale a.
≥	Maggiore o uguale a.
!	Non uguale a.
*	Goloso carattere jolly

È inoltre possibile restituire un insieme di oggetti in base all'impostazione o meno di un campo specifico mediante l' `null` parola chiave o relativa negazione `!null` come parte della query.

## Esempi di flusso di lavoro

Di seguito sono riportati alcuni esempi dei flussi di lavoro delle API REST in questo sito.

- ["Elencare i dischi"](#)

Filtro basato su `state` variabile per selezionare i dischi di riserva.

## Richiesta di campi oggetto specifici

Per impostazione predefinita, l'emissione di una chiamata API con GET restituisce solo gli attributi che identificano in modo univoco lo o gli oggetti, insieme a un collegamento automatico HAL. Questo insieme minimo di campi funge da chiave per ciascun oggetto e varia in base al tipo di oggetto. È possibile selezionare altre proprietà dell'oggetto utilizzando `fields` eseguire la query del parametro nei seguenti modi:

- Campi comuni o standard

Specificare `fields=*`` per recuperare i campi oggetto più comunemente utilizzati. Questi campi vengono generalmente mantenuti nella memoria del server locale o richiedono un'elaborazione ridotta per l'accesso. Si tratta delle stesse proprietà restituite per un oggetto dopo l'utilizzo DI GET con una chiave UUID (URL PATH Key).

- Tutti i campi

Specificare `fields=**` per recuperare tutti i campi oggetto, inclusi quelli che richiedono un'ulteriore elaborazione del server per l'accesso.

- Selezione di campi personalizzati

Utilizzare `fields=<field_name>` per specificare il campo esatto desiderato. Quando si richiedono più campi, i valori devono essere separati utilizzando virgole senza spazi.



Come Best practice, devi sempre identificare i campi specifici che desideri. Recuperare solo il set di campi comuni o tutti i campi quando necessario. Quali campi sono classificati come comuni e restituiti utilizzando `fields=*`, È determinato da NetApp in base all'analisi interna delle performance. La classificazione di un campo potrebbe cambiare nelle release future.

## Ordinamento degli oggetti nel set di output

I record di una raccolta di risorse vengono restituiti nell'ordine predefinito definito dall'oggetto. È possibile modificare l'ordine utilizzando `order_by` parametro di query con il nome del campo e la direzione di ordinamento come segue:

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

Ad esempio, è possibile ordinare il campo tipo in ordine decrescente seguito da id in ordine crescente:

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

Tenere presente quanto segue:

- Se si specifica un campo di ordinamento ma non si fornisce una direzione, i valori vengono ordinati in ordine crescente.

- Quando si includono più parametri, è necessario separare i campi con una virgola.

## Impaginazione durante il recupero di oggetti in una raccolta

Quando si esegue una chiamata API utilizzando GET per accedere a un insieme di oggetti dello stesso tipo, ONTAP tenta di restituire il maggior numero possibile di oggetti in base a due vincoli. È possibile controllare ciascuno di questi vincoli utilizzando parametri di query aggiuntivi sulla richiesta. Il primo vincolo raggiunto per una richiesta GET specifica termina la richiesta e limita quindi il numero di record restituiti.



Se una richiesta termina prima di scorrere tutti gli oggetti, la risposta contiene il collegamento necessario per recuperare il batch successivo di record.

### Limitazione del numero di oggetti

Per impostazione predefinita, ONTAP restituisce un massimo di 10,000 oggetti per una richiesta GET. È possibile modificare questo limite utilizzando `max_records` parametro di query. Ad esempio:

```
max_records=20
```

Il numero di oggetti effettivamente restituiti può essere inferiore al massimo effettivo, in base al relativo vincolo temporale e al numero totale di oggetti nel sistema.

### Limitare il tempo impiegato per recuperare gli oggetti

Per impostazione predefinita, ONTAP restituisce il maggior numero di oggetti possibile entro il tempo consentito per la richiesta GET. Il timeout predefinito è 15 secondi. È possibile modificare questo limite utilizzando `return_timeout` parametro di query. Ad esempio:

```
return_timeout=5
```

Il numero di oggetti effettivamente restituiti può essere inferiore al massimo effettivo, in base al vincolo relativo al numero di oggetti e al numero totale di oggetti nel sistema.

### Restringimento del set di risultati

Se necessario, è possibile combinare questi due parametri con altri parametri di query per restringere il set di risultati. Ad esempio, quanto segue restituisce fino a 10 eventi ems generati dopo il tempo specificato:

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

È possibile inviare più richieste per scorrere gli oggetti. Ogni successiva chiamata API deve utilizzare un nuovo valore temporale basato sull'ultimo evento dell'ultimo set di risultati.

## Proprietà delle dimensioni

I valori di input utilizzati con alcune chiamate API e alcuni parametri di query sono numerici. Invece di fornire un numero intero in byte, è possibile utilizzare un suffisso come mostrato nella tabella seguente.

Suffisso	Descrizione
KB	KB kilobyte (1024 byte) o kibyte
MB	MB Megabyte (KB x 1024 byte) o megibyte
GB	GB Gigabyte (MB x 1024 byte) o gibibyte
TB	TB terabyte (GB x 1024 byte) o tebibyte

Suffisso	Descrizione
PB	PB petabyte (TB x 1024 byte) o pebibyte

#### Informazioni correlate

- ["Riferimenti e accesso agli oggetti"](#)

## Interpretare una risposta dell'API REST ONTAP

Ogni richiesta API genera una risposta al client. È necessario esaminare la risposta per determinare se è stata eseguita correttamente e recuperare dati aggiuntivi in base alle necessità.

### Codice di stato HTTP

I codici di stato HTTP utilizzati dall'API REST ONTAP sono descritti di seguito.

Codice	Frase di motivazione	Descrizione
200	OK	Indica che le chiamate che non creano un nuovo oggetto sono riuscite.
201	Creato	Creazione di un oggetto riuscita. L'intestazione della posizione nella risposta include l'identificatore univoco dell'oggetto.
202	Accettato	Un lavoro in background è stato avviato per eseguire la richiesta, ma non è stato ancora completato.
400	Richiesta errata	L'input della richiesta non viene riconosciuto o non è appropriato.
401	Non autorizzato	Autenticazione dell'utente non riuscita.
403	Vietato	Accesso negato a causa di un errore di autorizzazione.
404	Non trovato	La risorsa a cui si fa riferimento nella richiesta non esiste.
405	Metodo non consentito	Il metodo HTTP nella richiesta non è supportato per la risorsa.
409	Conflitto	Un tentativo di creazione di un oggetto non è riuscito perché è necessario creare prima un oggetto diverso oppure l'oggetto richiesto esiste già.
500	Errore interno	Si è verificato un errore interno generale nel server.

### Intestazioni delle risposte

Nella risposta HTTP generata da ONTAP sono incluse diverse intestazioni.

#### Posizione

Quando viene creato un oggetto, l'intestazione di posizione include l'URL completo del nuovo oggetto, incluso l'identificatore univoco assegnato all'oggetto.

#### Tipo di contenuto

Questo normalmente sarà `application/hal+json`.

## Corpo di risposta

Il contenuto del corpo di risposta risultante da una richiesta API varia in base all'oggetto, al tipo di elaborazione e al successo o all'errore della richiesta. Il rendering della risposta viene sempre eseguito in JSON.

- Oggetto singolo

È possibile restituire un singolo oggetto con un set di campi in base alla richiesta. AD esempio, È possibile utilizzare GET per recuperare le proprietà selezionate di un cluster utilizzando l'identificatore univoco.

- Oggetti multipli

È possibile restituire più oggetti di una raccolta di risorse. In tutti i casi, viene utilizzato un formato coerente, con `num_records` indica il numero di record e record che contengono una matrice delle istanze dell'oggetto. Ad esempio, è possibile recuperare i nodi definiti in un cluster specifico.

- Oggetto lavoro

Se una chiamata API viene elaborata in modo asincrono, viene restituito un oggetto Job che ancora l'attività in background. Ad esempio, la richiesta DI PATCH utilizzata per aggiornare la configurazione del cluster viene elaborata in modo asincrono e restituisce un oggetto Job.

- Oggetto di errore

Se si verifica un errore, viene sempre restituito un oggetto Error. Ad esempio, si riceve un errore quando si tenta di modificare un campo non definito per un cluster.

- Oggetto JSON vuoto

In alcuni casi, non viene restituito alcun dato e il corpo della risposta include un oggetto JSON vuoto.

## Collegamento HAL

L'API REST di ONTAP utilizza HAL come meccanismo per supportare Hypermedia come motore di stato dell'applicazione (HATEOAS). Quando viene restituito un oggetto o un attributo che identifica una risorsa specifica, viene incluso anche un collegamento con codifica HAL che consente di individuare e determinare facilmente ulteriori dettagli sulla risorsa.

## Errori

Se si verifica un errore, viene restituito un oggetto di errore nel corpo della risposta.

### Formato

Un oggetto di errore ha il seguente formato:

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

È possibile utilizzare il valore del codice per determinare il tipo o la categoria di errore generale e il

messaggio per determinare l'errore specifico. Se disponibile, il campo di destinazione include l'input utente specifico associato all'errore.

### Codici di errore comuni

I codici di errore più comuni sono descritti nella seguente tabella. Le chiamate API specifiche possono includere codici di errore aggiuntivi.

Codice		Descrizione
1	409	Esiste già un oggetto con lo stesso identificatore.
2	400	Il valore di un campo ha un valore non valido o manca oppure è stato fornito un campo aggiuntivo.
3	400	L'operazione non è supportata.
4	405	Impossibile trovare un oggetto con l'identificatore specificato.
6	403	L'autorizzazione per eseguire la richiesta viene negata.
8	409	La risorsa è in uso.

## Elaborazione asincrona con l'API REST ONTAP

Dopo aver inviato una richiesta API progettata per essere eseguita in modo asincrono, un oggetto di lavoro viene sempre creato e restituito al chiamante. Il lavoro descrive e ancora un'attività in background che elabora la richiesta. A seconda del codice di stato HTTP, è necessario recuperare lo stato del lavoro per determinare se la richiesta ha avuto esito positivo.

Fare riferimento a. ["Riferimento API"](#) Per determinare quali chiamate API sono progettate per essere eseguite in modo asincrono.

### Controllo della modalità di elaborazione di una richiesta

È possibile utilizzare `return_timeout` Parametro query per controllare il modo in cui viene elaborata una chiamata API asincrona. L'utilizzo di questo parametro può avere due risultati.

#### Il timer scade prima del completamento della richiesta

Per le richieste valide, ONTAP restituisce un codice di stato HTTP 202 insieme all'oggetto lavoro. È necessario recuperare lo stato del lavoro per determinare se la richiesta è stata completata correttamente.

#### La richiesta viene completata prima della scadenza del timer

Se la richiesta è valida e viene completata correttamente prima della scadenza del tempo, ONTAP restituisce un codice di stato HTTP 200 insieme all'oggetto lavoro. Poiché la richiesta viene completata in modo sincrono, come indicato dal file 200, non è necessario recuperare lo stato del lavoro.



Il valore predefinito per `return_timeout` il parametro è pari a zero secondi. Pertanto, se non si include il parametro, il codice di stato HTTP 202 viene sempre restituito per una richiesta valida.

## Esecuzione di query sull'oggetto Job associato a una richiesta API

L'oggetto Job restituito nella risposta HTTP contiene diverse proprietà. È possibile eseguire una query sulla proprietà `state` in una successiva chiamata API per determinare se la richiesta è stata completata correttamente. Un oggetto Job si trova sempre in uno dei seguenti stati:

### stati non terminali

- In coda
- In esecuzione
- In pausa

### stati del terminale

- Successo
- Guasto

## Procedura generale per l'emissione di una richiesta asincrona

È possibile utilizzare la seguente procedura di alto livello per completare una chiamata API asincrona. Questo esempio presuppone `return_timeout` il parametro non viene utilizzato o il tempo scade prima del completamento del processo in background.

1. Eseguire una chiamata API progettata per essere eseguita in modo asincrono.
2. Ricevere una risposta HTTP 202 che indichi l'accettazione di una richiesta valida.
3. Estrarre l'identificatore per l'oggetto Job dal corpo della risposta.
4. All'interno di un ciclo temporizzato, eseguire le seguenti operazioni in ogni ciclo:
  - a. Ottenere lo stato corrente del lavoro.
  - b. Se il lavoro è in uno stato non terminale, eseguire nuovamente il loop.
5. Interrompere quando il lavoro raggiunge uno stato terminale (successo, errore).

### Informazioni correlate

- ["Aggiornare il contatto del cluster"](#)
- ["Recupera istanza lavoro"](#)

## Riferimenti e accesso agli oggetti dell'API REST di ONTAP

È possibile fare riferimento alle istanze di risorse o agli oggetti esposti tramite l'API REST di ONTAP e accedervi in diversi modi.

### Percorsi di accesso a oggetti

Ad un livello elevato, esistono due tipi di percorso quando si accede a un oggetto:

- Primario

L'oggetto è la destinazione principale o diretta della chiamata API.

- Straniero

L'oggetto non è il riferimento principale della chiamata API, ma è collegato all'oggetto primario. Si tratta quindi di un oggetto esterno o downstream a cui viene fatto riferimento attraverso un campo nell'oggetto primario.

## Accesso a un oggetto utilizzando l'UUID

A ogni oggetto viene assegnato un identificatore univoco al momento della creazione, che nella maggior parte dei casi è un UUID a 128 bit. I valori UUID assegnati sono immutabili e vengono utilizzati internamente in ONTAP per accedere e gestire le risorse. Per questo motivo, l'UUID fornisce in genere il modo più rapido e stabile per accedere agli oggetti.

Per molti tipi di risorse, è possibile fornire un valore UUID come parte della chiave di percorso nell'URL per accedere a un oggetto specifico. Ad esempio, è possibile utilizzare quanto segue per accedere a un'istanza di nodo: `/cluster/nodes/{uuid}`

## Accesso a un oggetto mediante una proprietà Object

Oltre a un UUID, è anche possibile accedere a un oggetto utilizzando una proprietà Object. Nella maggior parte dei casi, è conveniente utilizzare la proprietà name. Ad esempio, è possibile utilizzare il seguente parametro di query nella stringa URL per accedere a un'istanza del nodo con il relativo nome:

`/cluster/nodes?name=node_one`. Oltre a un parametro di query, è possibile accedere a un oggetto esterno tramite una proprietà nell'oggetto primario.

Sebbene sia possibile utilizzare il nome o un'altra proprietà per accedere a un oggetto invece dell'UUID, esistono diversi svantaggi:

- Il campo del nome non è immutabile e può essere modificato. Se il nome di un oggetto viene modificato prima di accedere a un oggetto, viene restituito l'oggetto errato o viene visualizzato un errore di accesso all'oggetto.



Questo problema può verificarsi con un METODO POST o PATCH su un oggetto esterno o con un metodo GET su un oggetto primario.

- ONTAP deve convertire il campo del nome nell'UUID corrispondente. Si tratta di un tipo di accesso indiretto che può diventare un problema di performance.

In particolare, è possibile un peggioramento delle performance quando si verifica una o più delle seguenti condizioni:

- VIENE utilizzato IL metodo GET
- È possibile accedere a un'ampia raccolta di oggetti
- Viene utilizzata una query complessa o elaborata

## Confronto tra cluster e contesto SVM

Esistono diversi endpoint REST che supportano sia un cluster che SVM. Quando si utilizza uno di questi endpoint, è possibile indicare il contesto della chiamata API tramite `scope=[svm|cluster]` valore. Esempi di endpoint che supportano un contesto doppio includono interfacce IP e ruoli di sicurezza.



Il valore dell'ambito ha una base di valori predefinita sulle proprietà fornite per ogni chiamata API.

## Utilizzo DI PATCH ed ELIMINAZIONE su un insieme di oggetti

Ogni endpoint REST che supporta PATCH o ELIMINI su un'istanza di risorsa supporta anche lo stesso metodo su un insieme di oggetti. L'unico requisito è che almeno un campo debba essere fornito attraverso un parametro di query nella stringa URL. Quando si rilascia UNA PATCH o SI ELIMINA una raccolta, ciò equivale a eseguire le seguenti operazioni internamente:

- GET basato su query per recuperare l'insieme
- Sequenza seriale di CHIAMATE PATCH o DI ELIMINAZIONE su ciascun oggetto della raccolta

Il timeout per l'operazione può essere impostato da `return_timeout` con un valore predefinito di 15 secondi. Se non viene completato prima del timeout, la risposta include un collegamento all'oggetto successivo. Per continuare l'operazione, è necessario eseguire nuovamente l'emissione dello stesso metodo HTTP utilizzando il collegamento successivo.

## Accedi alle metriche delle performance con l'API REST di ONTAP

ONTAP raccoglie le metriche delle performance relative agli oggetti e ai protocolli di storage SVM selezionati e riporta queste informazioni attraverso l'API REST. È possibile utilizzare questi dati per monitorare le prestazioni di un sistema ONTAP.

Per un determinato oggetto o protocollo di storage, i dati relativi alle performance sono suddivisi in tre categorie:

- IOPS
- Latenza
- Throughput

All'interno di ciascuna categoria, sono disponibili uno o più dei seguenti tipi di dati:

- Lettura (R)
- Scrittura (W)
- Altro (o)
- Totale (T)

La seguente tabella riassume i dati sulle performance disponibili attraverso l'API REST di ONTAP, inclusa la release al momento dell'aggiunta. Per ulteriori informazioni, consultare la pagina della documentazione online dell'API REST nel sistema ONTAP.

Oggetto o protocollo di storage	IOPS	Latenza	Throughput	Release di ONTAP
Porta Ethernet	Non applicabile	Non applicabile	RWT	9.8
Porta FC	RWOT	RWOT	RWT	9.8

Oggetto o protocollo di storage	IOPS	Latenza	Throughput	Release di ONTAP
Interfaccia IP	Non applicabile	Non applicabile	RWT	9.8
Interfaccia FC	RWOT	RWOT	RWT	9.8
Namespace NVMe	RWOT	RWOT	RWOT	9.8
Statistiche qtree	RWOT raw	Non applicabile	RWOT raw	9.8
Volume FlexCache (Nome volume)	RWOT	RWOT	RWT	9.8
Nodo: Utilizzo del processo	Utilizzo del processo come valore numerico	Utilizzo del processo come valore numerico	Utilizzo del processo come valore numerico	9.8
Volume cloud	RWOT	RWOT	Non applicato	9.7
LUN	RWOT	RWOT	RWOT	9.7
Aggregato	RWOT	RWOT	RWOT	9.7
Protocollo NFS SVM	RWOT	RWOT	RWT	9.7
Protocollo CIFS SVM	RWOT	RWOT	RWT	9.7
Protocollo FCP SVM	RWOT	RWOT	RWT	9.7
Protocollo iSCSI SVM	RWOT	RWOT	RWT	9.7
Protocollo NVMe SVM	RWOT	RWOT	RWT	9.7
Cluster	RWOT	RWOT	RWOT	9.6
Volumi	RWOT	RWOT	RWOT	9.6

## Informazioni sul copyright

Copyright © 2026 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

## Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.