



# **Detalhes da implementação REST**

## ONTAP automation

NetApp  
January 12, 2026

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

# Índice

Detalhes da implementação REST .....	1
Características operacionais da API REST do ONTAP .....	1
Transação de API de solicitação e resposta .....	1
Suporte para operações CRUD .....	1
Identificadores de objeto .....	1
Instâncias e coleções de objetos .....	1
Operações síncronas e assíncronas .....	1
Segurança .....	2
Variáveis de entrada para uma solicitação de API REST do ONTAP .....	3
Métodos HTTP .....	3
Variáveis de caminho .....	3
Cabeçalhos de solicitação .....	3
Corpo do pedido .....	4
Filtrando objetos .....	4
Solicitando campos de objeto específicos .....	5
Ordenar objetos no conjunto de saída .....	6
Paginação ao recuperar objetos em uma coleção .....	6
Propriedades de tamanho .....	7
Interpretar uma resposta da API REST do ONTAP .....	7
Código de status HTTP .....	7
Cabeçalhos de resposta .....	8
Corpo de resposta .....	8
Ligação HAL .....	9
Erros .....	9
Processamento assíncrono com a API REST do ONTAP .....	9
Controlar como uma solicitação é processada .....	10
Consultando o objeto Job associado a uma solicitação de API .....	10
Procedimento geral para emitir uma solicitação assíncrona .....	10
Referências e acesso a objetos da API REST do ONTAP .....	11
Caminhos de acesso a objetos .....	11
Acessando um objeto usando o UUID .....	11
Acessando um objeto usando uma propriedade de objeto .....	11
Contexto cluster versus SVM .....	12
USANDO PATCH e DELETE em uma coleção de objetos .....	12
Acesse métricas de performance com a API REST do ONTAP .....	12

# Detalhes da implementação REST

## Características operacionais da API REST do ONTAP

Embora O REST estabeleça um conjunto comum de tecnologias e práticas recomendadas, os detalhes de cada API podem variar com base nas escolhas de design.

### Transação de API de solicitação e resposta

Cada chamada de API REST é executada como uma solicitação HTTP para o sistema ONTAP, que gera uma resposta associada ao cliente. Esse par de solicitação/resposta é considerado uma transação de API. Antes de usar a API, você deve estar familiarizado com as variáveis de entrada disponíveis para controlar uma solicitação e o conteúdo da saída de resposta.

### Suporte para operações CRUD

Cada um dos recursos disponíveis por meio da API REST do ONTAP é acessado com base no modelo CRUD:

- Criar
- Leia
- Atualização
- Eliminar

Para alguns dos recursos, apenas um subconjunto das operações é suportado. Você deve revisar a página de documentação da API do ONTAP no cluster do ONTAP para obter mais informações sobre cada recurso.

### Identificadores de objeto

Cada instância ou objeto de recurso recebe um identificador exclusivo quando é criado. Na maioria dos casos, o identificador é um UUID de 128 bits. Esses identificadores são globalmente exclusivos dentro de um cluster ONTAP específico. Depois de emitir uma chamada de API que cria uma nova instância de objeto, um URL com o valor de id associado é retornado ao chamador no cabeçalho do local da resposta HTTP. Você pode extrair o identificador e usá-lo em chamadas subsequentes quando se refere à instância de recurso.



O conteúdo e a estrutura interna dos identificadores de objeto podem mudar a qualquer momento. Você só deve usar os identificadores nas chamadas de API aplicáveis conforme necessário ao se referir aos objetos associados.

### Instâncias e coleções de objetos

Dependendo do caminho do recurso e do método HTTP, uma chamada de API pode ser aplicada a uma instância de objeto específica ou a uma coleção de objetos.

### Operações síncronas e assíncronas

Existem duas maneiras pelas quais o ONTAP executa uma solicitação HTTP recebida de um cliente.

## Processamento síncrono

O ONTAP executa a solicitação imediatamente e responde com um código de status HTTP de 200 ou 201 se for bem-sucedido.

Cada solicitação usando os métodos GET, HEAD e OPTIONS é sempre realizada de forma síncrona. Além disso, as solicitações que usam POST, PATCH e DELETE são projetadas para serem executadas de forma síncrona, se elas forem concluídas em menos de dois segundos.

## Processamento assíncrono

Se uma solicitação assíncrona for válida, o ONTAP criará uma tarefa em segundo plano para processar a solicitação e um objeto de tarefa para ancorar a tarefa. O status HTTP 202 é retornado ao chamador juntamente com o objeto da tarefa. Para determinar o sucesso ou falha final, você deve recuperar o estado do trabalho.

As solicitações que usam os métodos POST, PATCH e DELETE são projetadas para serem executadas assincronamente se forem esperadas levar mais de dois segundos para serem concluídas.



O `return_timeout` parâmetro de consulta está disponível com chamadas assíncronas de API e pode converter uma chamada assíncrona para concluir de forma síncrona.

["Processamento assíncrono usando o objeto Job"](#) Consulte para obter mais informações.

## Segurança

A segurança fornecida com a API REST é baseada principalmente nos recursos de segurança existentes disponíveis com o ONTAP. A seguinte segurança é usada pela API:

### Segurança da camada de transporte

Todo o tráfego enviado pela rede entre o cliente e o ONTAP LIF normalmente é criptografado usando TLS, com base nas configurações do ONTAP.

### Autenticação de cliente

As mesmas opções de autenticação disponíveis com o Gerenciador de sistema do ONTAP e o SDK de gerenciamento de rede também podem ser usadas com a API REST do ONTAP.

### Autenticação HTTP

Em um nível HTTP, por exemplo, ao acessar diretamente a API REST do ONTAP, há duas opções de autenticação conforme descrito abaixo. Em cada caso, você precisa criar um cabeçalho de autorização HTTP e incluí-lo em cada solicitação.

Opção	Descrição
Autenticação básica HTTP	O nome de usuário e a senha do ONTAP são concatenados com dois pontos. A cadeia de caracteres é convertida para base64 e incluída no cabeçalho da solicitação.
OAuth 2,0	A partir do ONTAP 9.14, você pode solicitar um token de acesso a um servidor de autorização externo e incluí-lo como um token de portador no cabeçalho da solicitação.

Para obter mais detalhes sobre o OAuth 2,0 e como ele é implementado no ONTAP, ["Visão geral da implementação do ONTAP OAuth 2,0"](#) consulte . Veja também ["Prepare-se para usar os fluxos de trabalho"](#) abaixo neste site.

## Autorização ONTAP

O ONTAP implementa um modelo de autorização baseado em função. A conta que você usa ao acessar a API REST do ONTAP ou a página de documentação da API deve ter a autoridade adequada.

# Variáveis de entrada para uma solicitação de API REST do ONTAP

Você pode controlar como uma chamada de API é processada através de parâmetros e variáveis definidos na solicitação HTTP.

## Métodos HTTP

Os métodos HTTP suportados pela API REST do ONTAP são mostrados na tabela a seguir.



Nem todos os métodos HTTP estão disponíveis em cada um dos pontos finais REST. Além disso, tanto O PATCH quanto O DELETE podem ser usados em uma coleção. Consulte *referências de objetos e Access* para obter mais informações.

Método HTTP	Descrição
OBTER	Recupera propriedades de objeto em uma instância ou coleção de recursos.
POST	Cria uma nova instância de recurso com base na entrada fornecida.
PATCH	Atualiza uma instância de recurso existente com base na entrada fornecida.
ELIMINAR	Exclui uma instância de recurso existente.
CABEÇA	Efetivamente emite uma solicitação GET, mas retorna apenas os cabeçalhos HTTP.
OPÇÕES	Determine quais métodos HTTP são suportados em um endpoint específico.

## Variáveis de caminho

O caminho do endpoint usado com cada chamada da API REST pode incluir vários identificadores. Cada ID corresponde a uma instância de recurso específica. Os exemplos incluem ID do cluster e ID SVM.

## Cabeçalhos de solicitação

Você deve incluir vários cabeçalhos na solicitação HTTP.

### Tipo de conteúdo

Se o corpo da solicitação incluir JSON, esse cabeçalho deve ser definido como `application/json`.

### Aceitar

Esse cabeçalho deve ser definido como `application/hal+json`. Se, em vez disso, for definido como `application/json` nenhum dos links HAL será retornado, exceto um link necessário para recuperar o próximo lote de Registros. Se o cabeçalho for qualquer outra coisa além desses dois valores, o valor padrão do `content-type` cabeçalho na resposta será `application/hal+json`.

### Autorização

A autenticação básica deve ser definida com o nome de usuário e senha codificados como uma cadeia de

caracteres base64. Por exemplo:

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

## Corpo do pedido

O conteúdo do corpo da solicitação varia de acordo com a chamada específica. O corpo da solicitação HTTP consiste em um dos seguintes:

- Objeto JSON com variáveis de entrada
- Objeto JSON vazio

## Filtrando objetos

Ao emitir uma chamada de API com o método GET, você pode limitar ou filtrar os objetos retornados com base em qualquer atributo usando um parâmetro de consulta.

### Analizar e interpretar parâmetros de consulta

Um conjunto de um ou mais parâmetros pode ser anexado à cadeia de caracteres URL começando após o ? caractere. Se mais de um parâmetro for fornecido, os parâmetros de consulta serão divididos com base no & caractere. Cada chave e valor no parâmetro são divididos no = caractere.

Por exemplo, você pode especificar um valor exato para corresponder usando o sinal igual:

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

Para uma consulta mais complexa, o operador adicional é colocado após o sinal de igual. Por exemplo, para selecionar o conjunto de objetos com base em um campo específico que seja maior ou igual a algum valor, a consulta seria:

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

### Operadores de filtragem

Além dos exemplos fornecidos acima, operadores adicionais estão disponíveis para retornar objetos em uma faixa de valores. Um resumo dos operadores de filtragem suportados pela API REST do ONTAP é mostrado na tabela abaixo.



Todos os campos que não estão definidos são geralmente excluídos de consultas correspondentes.

Operador	Descrição
.	Igual a.
*	Menos de
>	Superior a.
O que é que é	Inferior ou igual a
>	Maior ou igual a
!	Não é igual a
*	Wildcard ganancioso

Você também pode retornar uma coleção de objetos com base se um campo específico está definido ou não usando a `null` palavra-chave ou sua negação `!null` como parte da consulta.

### Exemplos de fluxo de trabalho

Alguns exemplos estão incluídos abaixo dos fluxos de trabalho da API REST neste site.

- ["Listar discos"](#)

Filtro com base na `state` variável para selecionar os discos sobressalentes.

## Solicitando campos de objeto específicos

Por padrão, emitir uma chamada de API usando O GET retorna apenas os atributos que identificam exclusivamente o objeto ou objetos, juntamente com um auto-link HAL. Este conjunto mínimo de campos atua como uma chave para cada objeto e varia de acordo com o tipo de objeto. Você pode selecionar propriedades de objeto adicionais usando o `fields` parâmetro de consulta das seguintes maneiras:

- Campos comuns ou padrão

```
`fields=*` Especifique para recuperar os campos de objeto mais comumente usados. Esses campos são normalmente mantidos na memória do servidor local ou requerem pouco processamento para acessar. Estas são as mesmas propriedades retornadas para um objeto depois de usar GET com uma chave de caminho de URL (UUID).
```

- Todos os campos

```
`fields=**` Especifique para recuperar todos os campos de objeto, incluindo aqueles que exigem processamento adicional de servidor para acessar.
```

- Seleção de campo personalizada

```
`fields=<field_name>` Use para especificar o campo exato desejado. Ao solicitar vários campos, os valores devem ser separados usando vírgulas sem espaços.
```



Como prática recomendada, você deve sempre identificar os campos específicos que deseja. Você só deve recuperar o conjunto de campos comuns ou todos os campos quando necessário. Quais campos são classificados como comuns e retornados usando `fields=*` , são determinados pelo NetApp com base na análise interna de desempenho. A classificação de um campo pode mudar em versões futuras.

## Ordenar objetos no conjunto de saída

Os Registros em uma coleção de recursos são retornados na ordem padrão definida pelo objeto. Você pode alterar a ordem usando o `order_by` parâmetro de consulta com o nome do campo e a direção de classificação da seguinte forma:

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

Por exemplo, você pode classificar o campo `tipo` em ordem decrescente seguido de `id` em ordem crescente:

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

Observe o seguinte:

- Se você especificar um campo de classificação, mas não fornecer uma direção, os valores serão classificados em ordem crescente.
- Ao incluir vários parâmetros, você deve separar os campos com uma vírgula.

## Paginação ao recuperar objetos em uma coleção

Ao emitir uma chamada de API usando GET para acessar uma coleção de objetos do mesmo tipo, o ONTAP tenta retornar o maior número possível de objetos com base em duas restrições. Você pode controlar cada uma dessas restrições usando parâmetros de consulta adicionais na solicitação. A primeira restrição alcançada para uma SOLICITAÇÃO GET específica termina a solicitação e, portanto, limita o número de Registros retornados.



Se uma solicitação terminar antes de iterar todos os objetos, a resposta conterá o link necessário para recuperar o próximo lote de Registros.

### Limitando o número de objetos

Por padrão, o ONTAP retorna um máximo de 10.000 objetos para uma SOLICITAÇÃO GET. Você pode alterar esse limite usando o `max_records` parâmetro de consulta. Por exemplo:

```
max_records=20
```

O número de objetos realmente retornados pode ser menor do que o máximo em efeito, com base na restrição de tempo relacionada, bem como o número total de objetos no sistema.

### Limitar o tempo usado para recuperar os objetos

Por padrão, o ONTAP retorna o maior número possível de objetos dentro do tempo permitido para a solicitação GET. O tempo limite padrão é de 15 segundos. Você pode alterar esse limite usando o `return_timeout` parâmetro de consulta. Por exemplo:

```
return_timeout=5
```

O número de objetos realmente retornados pode ser menor que o máximo em efeito, com base na restrição relacionada ao número de objetos, bem como o número total de objetos no sistema.

### Estreitar o conjunto de resultados

Se necessário, você pode combinar esses dois parâmetros com parâmetros de consulta adicionais para restringir o conjunto de resultados. Por exemplo, o seguinte retorna até 10 eventos ems gerados após o tempo especificado:

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

Você pode emitir várias solicitações para percorrer os objetos. Cada chamada de API subsequente deve usar um novo valor de tempo com base no evento mais recente no último conjunto de resultados.

## Propriedades de tamanho

Os valores de entrada usados com algumas chamadas de API, bem como certos parâmetros de consulta são numéricos. Em vez de fornecer um inteiro em bytes, você pode opcionalmente usar um sufixo como mostrado na tabela a seguir.

Sufixo	Descrição
KB	KB kilobytes (1024 bytes) ou kibibytes
MB	MB megabytes (KB x 1024 bytes) ou megabytes
GB	GB Gigabytes (MB x 1024 bytes) ou gibbytes
TB	TB Terabytes (GB x 1024 bytes) ou tebibytes
PB	PB petabytes (TB x 1024 bytes) ou petabytes

### Informações relacionadas

- ["Referências de objetos e acesso"](#)

## Interpretar uma resposta da API REST do ONTAP

Cada solicitação de API gera uma resposta de volta ao cliente. Você deve examinar a resposta para determinar se ela foi bem-sucedida e recuperar dados adicionais, conforme necessário.

## Código de status HTTP

Os códigos de status HTTP usados pela API REST do ONTAP são descritos abaixo.

Código	Frase do motivo	Descrição
200	OK	Indica sucesso para chamadas que não criam um novo objeto.
201	Criado	Um objeto foi criado com sucesso. O cabeçalho de localização na resposta inclui o identificador exclusivo para o objeto.
202	Aceito	Um trabalho em segundo plano foi iniciado para executar a solicitação, mas ainda não foi concluído.
400	Pedido incorreto	A entrada de solicitação não é reconhecida ou é inadequada.
401	Não autorizado	A autenticação do utilizador falhou.
403	Proibido	O acesso é negado devido a um erro de autorização.
404	Não encontrado	O recurso referido na solicitação não existe.
405	Método não permitido	O método HTTP na solicitação não é suportado para o recurso.

Código	Frase do motivo	Descrição
409	Conflito	Uma tentativa de criar um objeto falhou porque um objeto diferente deve ser criado primeiro ou o objeto solicitado já existe.
500	Erro interno	Ocorreu um erro interno geral no servidor.

## Cabeçalhos de resposta

Vários cabeçalhos estão incluídos na resposta HTTP gerada pelo ONTAP.

### Localização

Quando um objeto é criado, o cabeçalho do local inclui o URL completo para o novo objeto, incluindo o identificador exclusivo atribuído ao objeto.

### Tipo de conteúdo

Isso normalmente será `application/hal+json`.

## Corpo de resposta

O conteúdo do corpo de resposta resultante de uma solicitação de API difere com base no objeto, no tipo de processamento e no sucesso ou falha da solicitação. A resposta é sempre renderizada em JSON.

- Um único objeto

Um único objeto pode ser retornado com um conjunto de campos com base na solicitação. Por exemplo, você pode usar GET para recuperar propriedades selecionadas de um cluster usando o identificador exclusivo.

- Vários objetos

Vários objetos de uma coleção de recursos podem ser retornados. Em todos os casos, há um formato consistente usado, com `num_records` a indicação do número de Registros e Registros contendo um array das instâncias do objeto. Por exemplo, você pode recuperar os nós definidos em um cluster específico.

- Objeto trabalho

Se uma chamada de API for processada de forma assíncrona, um objeto Job será retornado que ancora a tarefa em segundo plano. Por exemplo, a SOLICITAÇÃO DE PATCH usada para atualizar a configuração do cluster é processada de forma assíncrona e retorna um objeto Job.

- Objeto de erro

Se ocorrer um erro, um objeto de erro é sempre retornado. Por exemplo, você receberá um erro ao tentar alterar um campo não definido para um cluster.

- Objeto JSON vazio

Em certos casos, nenhum dado é retornado e o corpo de resposta inclui um objeto JSON vazio.

## Ligação HAL

A API REST do ONTAP usa o HAL como o mecanismo para suportar a Hypermedia como o motor do Estado de aplicação (HATEOAS). Quando um objeto ou atributo é retornado que identifica um recurso específico, um link codificado em HAL também é incluído, permitindo que você localize e determine facilmente detalhes adicionais sobre o recurso.

## Erros

Se ocorrer um erro, um objeto de erro é retornado no corpo de resposta.

### Formato

Um objeto de erro tem o seguinte formato:

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

Você pode usar o valor do código para determinar o tipo ou categoria de erro geral e a mensagem para determinar o erro específico. Quando disponível, o campo de destino inclui a entrada específica do usuário associada ao erro.

### Códigos de erro comuns

Os códigos de erro comuns são descritos na tabela a seguir. As chamadas de API específicas podem incluir códigos de erro adicionais.

Código		Descrição
1	409	Já existe um objeto com o mesmo identificador.
2	400	O valor de um campo tem um valor inválido ou está em falta ou um campo extra foi fornecido.
3	400	A operação não é suportada.
4	405	Não é possível encontrar um objeto com o identificador especificado.
6	403	A permissão para executar a solicitação é negada.
8	409	O recurso está em uso.

## Processamento assíncrono com a API REST do ONTAP

Depois de emitir uma solicitação de API projetada para ser executada assincronamente, um objeto de tarefa é sempre criado e retornado ao chamador. O trabalho descreve e

ancora uma tarefa em segundo plano que processa a solicitação. Dependendo do código de status HTTP, você deve recuperar o estado da tarefa para determinar se a solicitação foi bem-sucedida.

["Referência da API"](#) Consulte para determinar quais chamadas de API são projetadas para serem executadas de forma assíncrona.

## Controlar como uma solicitação é processada

Você pode usar o `return_timeout` parâmetro de consulta para controlar como uma chamada de API assíncrona é processada. Há dois resultados possíveis ao usar este parâmetro.

### O temporizador expira antes de a solicitação ser concluída

Para solicitações válidas, o ONTAP retorna um código de status HTTP 202 junto com o objeto job. Você deve recuperar o estado do trabalho para determinar se a solicitação foi concluída com êxito.

### A solicitação é concluída antes que o temporizador expire

Se a solicitação for válida e for concluída com êxito antes que o tempo expire, o ONTAP retornará um código de status HTTP 200 juntamente com o objeto da tarefa. Como a solicitação é concluída de forma síncrona, como indicado pelo 200, você não precisa recuperar o estado da tarefa.



O valor padrão para o `return_timeout` parâmetro é zero segundos. Portanto, se você não incluir o parâmetro, o código de status HTTP 202 sempre será retornado para uma solicitação válida.

## Consultando o objeto Job associado a uma solicitação de API

O objeto Job retornado na resposta HTTP contém várias propriedades. Você pode consultar a propriedade `State` em uma chamada API subsequente para determinar se a solicitação foi concluída com êxito. Um objeto Job está sempre num dos seguintes estados:

### estados não-terminais

- Em fila de espera
- Em execução
- Em pausa

### estados do terminal

- Sucesso
- Falha

## Procedimento geral para emitir uma solicitação assíncrona

Você pode usar o seguinte procedimento de alto nível para concluir uma chamada assíncrona de API. Este exemplo assume que o `return_timeout` parâmetro não é utilizado ou que o tempo expira antes da conclusão do trabalho em segundo plano.

1. Emite uma chamada de API que foi projetada para ser executada assincronamente.
2. Receber uma resposta HTTP 202 indicando aceitação de uma solicitação válida.
3. Extraia o identificador do objeto Job do corpo de resposta.

4. Dentro de um loop temporizado, execute o seguinte em cada ciclo:
  - a. Obtenha o estado atual do trabalho.
  - b. Se o trabalho estiver em um estado não terminal, execute o loop novamente.
5. Pare quando o trabalho atingir um estado terminal (sucesso, falha).

#### Informações relacionadas

- ["Atualizar contacto do cluster"](#)
- ["Obter instância de trabalho"](#)

## Referências e acesso a objetos da API REST do ONTAP

As instâncias de recursos ou objetos expostos por meio da API REST do ONTAP podem ser referenciados e acessados de várias maneiras diferentes.

### Caminhos de acesso a objetos

Em um nível alto, há dois tipos de caminho ao acessar um objeto:

- Primário

O objeto é o alvo principal ou direto da chamada API.

- Estrangeiro

O objeto não é a referência primária da chamada API, mas está ligado a partir do objeto primário. É, portanto, um objeto estranho ou downstream e referenciado através de um campo no objeto primário.

### Acessando um objeto usando o UUID

Cada objeto recebe um identificador exclusivo quando é criado, que na maioria dos casos é um UUID de 128 bits. Os valores UUID atribuídos são imutáveis e são usados internamente no ONTAP para acessar e gerenciar os recursos. Por causa disso, o UUID geralmente fornece a maneira mais rápida e estável de acessar objetos.

Para muitos dos tipos de recurso, um valor UUID pode ser fornecido como parte da chave de caminho no URL para acessar um objeto específico. Por exemplo, você pode usar o seguinte para acessar uma instância de nó: `/cluster/nodes/{uuid}

### Acessando um objeto usando uma propriedade de objeto

Além de um UUID, você também pode acessar um objeto usando uma propriedade de objeto. Na maioria dos casos, é conveniente usar a propriedade name. Por exemplo, você pode usar o seguinte parâmetro de consulta na cadeia de caracteres de URL para acessar uma instância de nó pelo seu nome: `/cluster/nodes?name=node\_one`. Além de um parâmetro de consulta, um objeto estranho pode ser acessado através de uma propriedade no objeto primário.

Embora você possa usar o nome ou outra propriedade para acessar um objeto em vez do UUID, existem várias desvantagens possíveis:

- O campo de nome não é imutável e pode ser alterado. Se o nome de um objeto for alterado antes de

acessar um objeto, o objeto errado será retornado ou um erro de acesso ao objeto falhará.



Esse problema pode ocorrer com um método POST ou PATCH em um objeto estranho ou com um método GET em um objeto primário.

- O ONTAP deve traduzir o campo de nome para o UUID correspondente. Este é um tipo de acesso indireto que pode se tornar um problema de desempenho.

Em particular, uma degradação do desempenho é possível quando uma ou mais das seguintes situações é verdadeira:

- Método GET é usado
- Uma grande coleção de objetos é acessada
- Uma consulta complexa ou elaborada é usada

## Contexto cluster versus SVM

Há vários pontos de extremidade REST que são compatíveis com cluster e SVM. Ao usar um desses endpoints, você pode indicar o contexto da chamada da API por meio do `scope=[svm|cluster]` valor. Exemplos de endpoints que suportam um contexto duplo incluem interfaces IP e funções de segurança.



O valor do escopo tem uma base de valor padrão nas propriedades fornecidas para cada chamada de API.

## USANDO PATCH e DELETE em uma coleção de objetos

Todos os endpoints REST que suportam PATCH ou EXCLUSÃO em uma instância de recurso também suportam o mesmo método em uma coleção de objetos. O único requisito é que pelo menos um campo deve ser fornecido através de um parâmetro de consulta na cadeia de caracteres de URL. Ao emitir UM PATCH ou EXCLUIR sobre uma coleção, isso é equivalente a fazer o seguinte internamente:

- GET baseado em consulta para recuperar a coleção
- Sequência serial de PATCH ou EXCLUIR chamadas em cada objeto da coleção

O tempo limite para a operação pode ser definido por `return_timeout` com um padrão de 15 segundos. Se não for concluída antes do tempo limite, a resposta inclui um link para o próximo objeto. Você deve reemitir o mesmo método HTTP usando o próximo link para continuar a operação.

## Acesse métricas de performance com a API REST do ONTAP

O ONTAP coleta métricas de performance sobre protocolos e objetos de storage selecionados da SVM e relata essas informações por meio da API REST. Você pode usar esses dados para monitorar o desempenho de um sistema ONTAP.

Para um determinado objeto de storage ou protocolo, os dados de performance se dividem em três categorias:

- IOPS

- Latência
- Taxa de transferência

Dentro de cada categoria, um ou mais dos seguintes tipos de dados estão disponíveis:

- Leitura (R)
- Escrever (W)
- Outro (o)
- Total (T)

A tabela a seguir resume os dados de desempenho disponíveis por meio da API REST do ONTAP, incluindo a versão quando eles foram adicionados. Consulte a página de documentação on-line da API REST em seu sistema ONTAP para obter mais informações.

Objeto de storage ou protocolo	IOPS	Latência	Taxa de transferência	Lançamento do ONTAP
Porta Ethernet	Não aplicável	Não aplicável	RWT	9,8
Porta de FC	RWOT	RWOT	RWT	9,8
Interface IP	Não aplicável	Não aplicável	RWT	9,8
Interface FC	RWOT	RWOT	RWT	9,8
Namespace NVMe	RWOT	RWOT	RWOT	9,8
Estatísticas Qtree	RWOT bruto	Não aplicável	RWOT bruto	9,8
Volume FlexCache	RWOT	RWOT	RWT	9,8
Nó – utilização do processo	Utilização do processo como valor numérico	Utilização do processo como valor numérico	Utilização do processo como valor numérico	9,8
Volume de nuvem	RWOT	RWOT	Não aplicável	9,7
LUN	RWOT	RWOT	RWOT	9,7
Agregado	RWOT	RWOT	RWOT	9,7
Protocolo SVM NFS	RWOT	RWOT	RWT	9,7
Protocolo CIFS da SVM	RWOT	RWOT	RWT	9,7
Protocolo SVM FCP	RWOT	RWOT	RWT	9,7
Protocolo SVM iSCSI	RWOT	RWOT	RWT	9,7
Protocolo SVM NVMe	RWOT	RWOT	RWT	9,7
Cluster	RWOT	RWOT	RWOT	9,6
Volumes	RWOT	RWOT	RWOT	9,6

## Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTE DOCUMENTO. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSAENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTE SOFTWARE, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

**LEGENDA DE DIREITOS LIMITADOS:** o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

## Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.