Konzepte

ONTAP Select

NetApp
November 08, 2026

This PDF was generated from https://docs.netapp.com/de-de/ontap-select-9191/concept_api_rest.html on November 08, 2026. Always check docs.netapp.com for the latest.

Inhalt

Konzepte . . . 1
Informationen zu REST-Webdiensten für die Bereitstellung und Verwaltung von ONTAP Select Clustern . . . 1
Architektur und klassische Beschränkungen . . . 1
Ressourcen und Statusdarstellung . . . 1
URI-Endpunkte . . . 1
HTTP-Nachrichten . . . 1
JSON-Formatierung . . . 2
Informationen zum Zugriff auf die ONTAP Select Deploy API . . . 2
Nativen Benutzeroberfläche für das Dienstprogramm bereitstellen . . . 2
ONTAP Select Deploy Online-Dokumentationsseite . . . 2
Benutzerdefiniertes Programm . . . 2
Grundlegende Betriebsmerkmale der ONTAP Select Deploy API . . . 2
Hypervisor-Host versus ONTAP Select Node . . . 3
Objektbezeichner . . . 3
Anforderungskennungen . . . 3
Synchrone und asynchrone Aufrufe . . . 3
Bestätigen Sie den Abschluss eines langwierigen Auftrags . . . 3
Sicherheit . . . 4
Anfrage- und Antwort-API-Transaktion für ONTAP Select . . . 4
Eingabevariablen, die eine API-Anfrage steuern . . . 4
Eine API-Antwort interpretieren . . . 6
Asynchrone Verarbeitung mit dem Job-Objekt für ONTAP Select . . . 7
Asynchrone Anfragen werden mithilfe des Job-Objekts beschrieben. . . . 7
Abfrage des Job-Objekts, das einer API-Anfrage zugeordnet ist . . . 8
Allgemeines Verfahren zum Ausstellen einer asynchronen Anfrage . . . 8

Konzepte

Informationen zu REST-Webdiensten für die Bereitstellung und Verwaltung von ONTAP Select Clustern

Representational State Transfer (REST) ist ein Stil zur Entwicklung verteilter Webanwendungen. Angewendet auf das Design von Web-Service-APIs, etabliert REST eine Reihe von Technologien und Best Practices für die Bereitstellung serverbasierter Ressourcen und die Verwaltung ihrer Zustände. Es nutzt gängige Protokolle und Standards, um eine flexible Grundlage für die Bereitstellung und Verwaltung von ONTAP Select Clustern zu bieten.

Architektur und klassische Beschränkungen

REST wurde von Roy Fielding in seiner PhD "Dissertation" an der UC Irvine im Jahr 2000 formal artikuliert. Es definiert einen Architekturstil durch eine Reihe von Einschränkungen, die gemeinsam webbasierte Anwendungen und die zugrunde liegenden Protokolle verbessert haben. Die Einschränkungen etablieren eine RESTful-Webservices-Anwendung auf Basis einer Client/Server-Architektur unter Verwendung eines zustandslosen Kommunikationsprotokolls.

Ressourcen und Statusdarstellung

Ressourcen sind die grundlegenden Komponenten eines webbasierten Systems. Bei der Entwicklung einer REST-Webservices-Anwendung gehören folgende frühe Entwurfsaufgaben dazu:

- Identifizierung von System- oder serverbasierten Ressourcen. Jedes System nutzt und verwaltet Ressourcen. Eine Ressource kann eine Datei, eine Geschäftstransaktion, ein Prozess oder eine administrative Einheit sein. Eine der ersten Aufgaben beim Entwurf einer Anwendung auf Basis von REST-Webdiensten ist die Identifizierung der Ressourcen.
- Definition von Ressourcenzuständen und zugehörigen Zustandsoperationen: Ressourcen befinden sich stets in einem von endlich vielen Zuständen. Die Zustände sowie die zugehörigen Operationen, die Zustandsänderungen bewirken, müssen klar definiert sein.

Nachrichten werden zwischen Client und Server ausgetauscht, um auf die Ressourcen zuzugreifen und ihren Zustand gemäß dem generischen CRUD-Modell (Create, Read, Update, and Delete) zu ändern.

URI-Endpunkte

Jede REST-Ressource muss definiert und mithilfe eines klar definierten Adressierungsschemas bereitgestellt werden. Die Endpunkte, an denen die Ressourcen lokalisiert und identifiziert werden, verwenden einen Uniform Resource Identifier (URI). Der URI bietet ein allgemeines Framework zur Erstellung eines eindeutigen Namens für jede Ressource im Netzwerk. Der Uniform Resource Locator (URL) ist eine Art URI, der in Webdiensten verwendet wird, um Ressourcen zu identifizieren und darauf zuzugreifen. Ressourcen werden typischerweise in einer hierarchischen Struktur ähnlich einem Dateiverzeichnis bereitgestellt.

HTTP-Nachrichten

HTTP (Hypertext Transfer Protocol) ist das Protokoll, das von Webdienst-Clients und -Servern zum Austausch von Anfrage- und Antwortnachrichten über die Ressourcen verwendet wird. Als Teil der Entwicklung einer webbasierten Dienstanwendung werden HTTP-Verben (wie GET und POST) den Ressourcen und den

entsprechenden Statusverwaltungsaktionen zugeordnet.

HTTP ist zustandslos. Um also eine Reihe zusammengehöriger Anfragen und Antworten einer Transaktion zuzuordnen, müssen zusätzliche Informationen in den HTTP-Headern enthalten sein, die mit den Anfrage-/Antwortdaten übertragen werden.

JSON-Formatierung

Informationen lassen sich auf verschiedene Weisen strukturieren und zwischen Client und Server übertragen. Die gängigste Option (und diejenige, die auch von der Deploy REST API verwendet wird) ist JavaScript Object Notation (JSON). JSON ist ein Industriestandard zur Darstellung einfacher Datenstrukturen in Klartext und dient der Übertragung von Zustandsinformationen, die Ressourcen beschreiben.

Informationen zum Zugriff auf die ONTAP Select Deploy API

Aufgrund der inhärenten Flexibilität von REST-Webdiensten kann auf die ONTAP Select Deploy API auf verschiedene Arten zugegriffen werden.

Die in ONTAP Select Deploy enthaltene REST-API besitzt eine Versionsnummer. Diese Versionsnummer ist unabhängig von der Deploy-Versionsnummer. Das ONTAP Select 9.19.1 Deploy Verwaltungsprogramm beinhaltet Version 3 der REST-API.

Nativen Benutzeroberfläche für das Dienstprogramm bereitstellen

Der primäre Zugriff auf die API erfolgt über die ONTAP Select Deploy Web-Benutzeroberfläche. Der Browser ruft die API auf und formatiert die Daten entsprechend dem Design der Benutzeroberfläche. Sie können auch über die Befehlszeilenschnittstelle des Deploy-Dienstprogramms auf die API zugreifen.

ONTAP Select Deploy Online-Dokumentationsseite

Die ONTAP Select Deploy Online-Dokumentationsseite bietet einen alternativen Zugriffspunkt bei der Verwendung eines Browsers. Neben der Möglichkeit, einzelne API-Aufrufe direkt auszuführen, enthält die Seite auch eine detaillierte Beschreibung der API, einschließlich Eingabeparametern und weiterer Optionen für jeden Aufruf. Die API-Aufrufe sind in mehrere verschiedene Funktionsbereiche oder Kategorien unterteilt.

Benutzerdefiniertes Programm

Sie können die Deploy API mit verschiedenen Programmiersprachen und Tools nutzen. Beliebte Optionen sind Python, Java und cURL. Ein Programm, Skript oder Tool, das die API verwendet, fungiert als REST-Webservices-Client. Die Verwendung einer Programmiersprache ermöglicht ein besseres Verständnis der API und bietet die Möglichkeit, die ONTAP Select Deployments zu automatisieren.

Grundlegende Betriebsmerkmale der ONTAP Select Deploy API

REST etabliert zwar einen gemeinsamen Satz an Technologien und Best Practices, die Details der einzelnen APIs können jedoch je nach Design variieren. Sie sollten sich mit den Details und den betrieblichen Eigenschaften der ONTAP Select Deploy API vertraut machen, bevor Sie die API verwenden.

Hypervisor-Host versus ONTAP Select Node

Ein Hypervisor-Host ist die zentrale Hardwareplattform, auf der eine ONTAP Select virtuelle Maschine gehostet wird. Wenn eine virtuelle ONTAP Select-Maschine auf einem Hypervisor-Host bereitgestellt und aktiv ist, wird die virtuelle Maschine als *ONTAP Select-Knoten* betrachtet. Mit Version 3 der Deploy REST API sind die Host- und Knoten-Objekte getrennt und unterschiedlich. Dies ermöglicht eine Eins-zu-Viele-Beziehung, bei der ein oder mehrere ONTAP Select-Knoten auf demselben Hypervisor-Host ausgeführt werden können.

Objektbezeichner

Jeder Ressourceninstanz bzw. jedem Objekt wird bei der Erstellung eine eindeutige Kennung zugewiesen. Diese Kennungen sind innerhalb einer bestimmten ONTAP Select Deploy Instanz global eindeutig. Nach einem API-Aufruf, der eine neue Objektinstanz erstellt, wird der zugehörige id-Wert im `location` Header der HTTP-Antwort an den Aufrufer zurückgegeben. Sie können die Kennung extrahieren und bei nachfolgenden Aufrufen verwenden, um auf die Ressourceninstanz zu verweisen.

Der Inhalt und die interne Struktur der Objektbezeichner können sich jederzeit ändern. Sie sollten die Bezeichner nur bei den entsprechenden API-Aufrufen verwenden, wenn Sie auf die zugehörigen Objekte verweisen müssen.

Anforderungskennungen

Jeder erfolgreichen API-Anfrage wird eine eindeutige Kennung zugewiesen. Die Kennung wird im `request-id` Header der zugehörigen HTTP-Antwort zurückgegeben. Sie können eine Anfrage-Kennung verwenden, um gemeinsam auf die Aktivitäten einer einzelnen spezifischen API-Anfrage-Antwort-Transaktion zu verweisen. Beispielsweise können Sie alle Ereignismeldungen für eine Transaktion basierend auf der Anfrage-ID abrufen.

Synchrone und asynchrone Aufrufe

Es gibt zwei Hauptmethoden, mit denen ein Server eine vom Client empfangene HTTP (Hypertext Transfer Protocol)-Anfrage ausführt:

- Synchron: Der Server führt die Anfrage sofort aus und antwortet mit einem Statuscode von 200, 201 oder 204.
- Asynchron Der Server akzeptiert die Anfrage und antwortet mit dem Statuscode 202. Dies bedeutet, dass der Server die Clientanfrage angenommen und einen Hintergrundprozess zur Bearbeitung gestartet hat. Ob die Anfrage erfolgreich war oder nicht, ist nicht sofort ersichtlich und muss durch weitere API-Aufrufe ermittelt werden.

Bestätigen Sie den Abschluss eines langwierigen Auftrags

Generell wird jede Operation, die lange dauern kann, asynchron mithilfe einer Hintergrundaufgabe auf dem Server verarbeitet. Mit der Deploy REST API ist jede Hintergrundaufgabe durch ein Job-Objekt verankert, das die Aufgabe verfolgt und Informationen wie den aktuellen Status bereitstellt. Ein Job-Objekt, einschließlich seiner eindeutigen Kennung, wird in der HTTP-Antwort zurückgegeben, nachdem eine Hintergrundaufgabe erstellt wurde.

Sie können das Job-Objekt direkt abfragen, um den Erfolg oder Misserfolg des zugehörigen API-Aufrufs zu ermitteln. Weitere Informationen finden Sie unter *asynchrone Verarbeitung mit dem Job-Objekt*.

Neben der Verwendung des Job-Objekts gibt es weitere Möglichkeiten, den Erfolg oder Misserfolg einer Anfrage zu ermitteln, darunter:

- Ereignismeldungen Sie können alle Ereignismeldungen abrufen, die mit einem bestimmten API-Aufruf verknüpft sind, indem Sie die mit der ursprünglichen Antwort zurückgegebenen Anfrage-ID verwenden. Die Ereignismeldungen enthalten typischerweise einen Hinweis auf Erfolg oder Misserfolg und können auch beim Debuggen eines Fehlerzustands nützlich sein.
- Ressourcenzustand oder -status Mehrere der Ressourcen verwalten einen Zustand oder Statuswert, den Sie abfragen können, um indirekt den Erfolg oder Misserfolg einer Anfrage zu bestimmen.

Sicherheit

Die Deploy-API verwendet folgende Sicherheitstechnologien:

- Transport Layer Security: Der gesamte Netzwerkverkehr zwischen Deploy-Server und Client wird per TLS verschlüsselt. Die Verwendung des HTTP (Hypertext Transfer Protocol) über einen unverschlüsselten Kanal wird nicht unterstützt. TLS Version 1.2 wird unterstützt.
- Die HTTP-Authentifizierung verwendet die Basic-Authentifizierung für jede API-Transaktion. Jedem Request wird ein HTTP-Header hinzugefügt, der den Benutzernamen und das Passwort als Base64-String enthält.

Anfrage- und Antwort-API-Transaktion für ONTAP Select

Jeder Deploy-API-Aufruf wird als HTTP-Anfrage an die Deploy-virtuelle Maschine gesendet, die eine entsprechende Antwort an den Client generiert. Dieses Anfrage-/Antwort-Paar wird als API-Transaktion betrachtet. Bevor Sie die Deploy-API verwenden, sollten Sie mit den verfügbaren Eingabevariablen zur Steuerung einer Anfrage und dem Inhalt der Antwortausgabe vertraut sein.

Eingabevariablen, die eine API-Anfrage steuern

Sie können steuern, wie ein API-Aufruf durch Parameter verarbeitet wird, die in der HTTP-Anfrage festgelegt sind.

Anfrageheader

Sie müssen mehrere Header in die HTTP-Anfrage einfügen, darunter:

- content-type Wenn der Anfragetext JSON enthält, muss dieser Header auf application/json gesetzt werden.
- accept Falls der Antworttext JSON enthalten soll, muss dieser Header auf application/json gesetzt werden.
- Die Basic-Authentifizierung muss mit dem Benutzernamen und Passwort erfolgen, die in einer Base64-Zeichenkette codiert sind.

Anfragekörper

Der Inhalt des Anfragetextes variiert je nach spezifischem Aufruf. Der HTTP-Anfragetext besteht aus einem der folgenden Elemente:

- JSON-Objekt mit Eingabevariablen (z. B. dem Namen eines neuen Clusters)
- Leer

Filterobjekte

Bei einem API-Aufruf mit GET können Sie die zurückgegebenen Objekte anhand beliebiger Attribute einschränken oder filtern. Beispielsweise können Sie einen exakten Wert angeben, der übereinstimmen soll:

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

Neben der exakten Übereinstimmung stehen weitere Operatoren zur Verfügung, um eine Menge von Objekten über einen Wertebereich zurückzugeben. ONTAP Select unterstützt die unten aufgeführten Filteroperatoren.

Operator	Beschreibung
=	Gleich
<	Weniger als
>	Größer als
⇐	Kleiner oder gleich
>=	Größer oder gleich
	Oder
!	Nicht gleich
*	Gieriger Wildcard

Sie können auch eine Menge von Objekten zurückgeben, je nachdem, ob ein bestimmtes Feld gesetzt ist oder nicht, indem Sie das Schlüsselwort null oder seine Negation (!null) als Teil der Abfrage verwenden.

Objektfelder auswählen

Standardmäßig liefert ein API-Aufruf mit der GET-Methode nur die Attribute zurück, die das Objekt oder die Objekte eindeutig identifizieren. Dieser minimale Satz an Feldern dient als Schlüssel für jedes Objekt und variiert je nach Objekttyp. Sie können zusätzliche Objekteigenschaften mithilfe des Abfrageparameters fields auf folgende Weise auswählen:

- Kostengünstige Felder: Geben Sie `fields=*` an, um die Objektfelder abzurufen, die im lokalen Serverspeicher verwaltet werden oder für deren Zugriff nur geringe Verarbeitung erforderlich ist.
- Teure Felder Geben Sie `fields=**` an, um alle Objektfelder abzurufen, einschließlich derjenigen, für deren Zugriff eine zusätzliche Serververarbeitung erforderlich ist.
- Benutzerdefinierte Feldauswahl Verwenden Sie `fields=FIELDNAME` , um das genaue Feld anzugeben, das Sie möchten. Wenn Sie mehrere Felder anfordern, müssen die Werte durch Kommas ohne Leerzeichen getrennt werden.



Als Best Practice sollten Sie immer die spezifischen Felder identifizieren, die Sie benötigen. Sie sollten das Set an kostengünstigen oder teuren Feldern nur bei Bedarf abrufen. Die Einteilung in kostengünstige und teure Felder wird von NetApp basierend auf interner Leistungsanalyse bestimmt. Die Klassifizierung für ein bestimmtes Feld kann sich jederzeit ändern.

Sortieren Sie die Objekte im Ausgabesatz

Die Datensätze einer Ressourcensammlung werden in der vom Objekt definierten Standardreihenfolge zurückgegeben. Sie können die Reihenfolge mithilfe des Abfrageparameters order_by mit dem Feldnamen und der Sortierrichtung wie folgt ändern:

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

Sie können beispielsweise das Feld „Typ“ in absteigender Reihenfolge und anschließend das Feld „ID“ in aufsteigender Reihenfolge sortieren:
```
order_by=type desc, id asc
```

Wenn Sie mehrere Parameter angeben, müssen Sie die Felder durch ein Komma trennen.

Paginierung

Beim Ausführen eines API-Aufrufs mit GET zum Zugriff auf eine Sammlung von Objekten desselben Typs werden standardmäßig alle übereinstimmenden Objekte zurückgegeben. Bei Bedarf können Sie die Anzahl der zurückgegebenen Datensätze mithilfe des Abfrageparameters max_records in der Anfrage begrenzen. Beispiel:
```
max_records=20
```

Bei Bedarf können Sie diesen Parameter mit anderen Abfrageparametern kombinieren, um die Ergebnismenge einzugrenzen. Beispielsweise liefert die folgende Abfrage bis zu 10 Systemereignisse zurück, die nach dem angegebenen Zeitpunkt generiert wurden:
```
time⇒ 2019-04-04T15:41:29.140265Z&max_records=10
```

Sie können mehrere Anfragen senden, um die Ereignisse (oder beliebige Objekttypen) seitenweise zu durchlaufen. Jeder nachfolgende API-Aufruf sollte einen neuen Zeitwert basierend auf dem neuesten Ereignis im letzten Ergebnissatz verwenden.

Eine API-Antwort interpretieren

Jede API-Anfrage erzeugt eine Antwort, die an den Client zurückgesendet wird. Sie können die Antwort untersuchen, um festzustellen, ob sie erfolgreich war, und bei Bedarf zusätzliche Daten abrufen.

HTTP-Statuscode

Die von der Deploy REST API verwendeten HTTP-Statuscodes werden im Folgenden beschrieben.

Code	Bedeutung	Beschreibung
200	OK	Zeigt den Erfolg von Aufrufen an, die kein neues Objekt erzeugen.
201	Erstellt	Ein Objekt wurde erfolgreich erstellt; der Location-Response-Header enthält die eindeutige Kennung für das Objekt.
202	Akzeptiert	Zur Ausführung der Anfrage wurde ein Hintergrundprozess mit langer Laufzeit gestartet, der jedoch noch nicht abgeschlossen ist.
400	Ungültige Anforderung	Die Eingabe der Anfrage wird nicht erkannt oder ist ungeeignet.
403	Verboten	Der Zugriff wurde aufgrund eines Autorisierungsfehlers verweigert.
404	Nicht gefunden	Die in der Anfrage genannte Ressource existiert nicht.
405	Methode nicht zulässig	Das HTTP-Verb in der Anfrage wird für die Ressource nicht unterstützt.
409	Konflikt	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, da das Objekt bereits existiert.

Code	Bedeutung	Beschreibung
500	Interner Fehler	Auf dem Server ist ein allgemeiner interner Fehler aufgetreten.
501	Nicht implementiert	Die URI ist bekannt, aber sie kann die Anfrage nicht ausführen.

Antwortheader

Die vom Deploy-Server generierte HTTP-Antwort enthält mehrere Header, darunter:

- request-id Jeder erfolgreichen API-Anfrage wird eine eindeutige request-id zugewiesen.
- location Wenn ein Objekt erstellt wird, enthält der Location-Header die vollständige URL zum neuen Objekt einschließlich der eindeutigen Objektkennung.

Antworttext

Der Inhalt der Antwort auf eine API-Anfrage variiert je nach Objekt, Verarbeitungstyp und Erfolg oder Misserfolg der Anfrage. Der Antworttext wird im JSON-Format dargestellt.

- Ein einzelnes Objekt kann mit einer Reihe von Feldern basierend auf der Anfrage zurückgegeben werden. Beispielsweise können Sie mit GET ausgewählte Eigenschaften eines Clusters anhand seiner eindeutigen Kennung abrufen.
- Es können mehrere Objekte aus einer Ressourcensammlung zurückgegeben werden. Dabei wird stets ein einheitliches Format verwendet, wobei `num_records` die Anzahl der Datensätze angegeben wird und die Datensätze ein Array der Objektinstanzen enthalten. Beispielsweise können Sie alle in einem bestimmten Cluster definierten Knoten abrufen.
- Wird ein API-Aufruf asynchron verarbeitet, wird ein Job-Objekt zurückgegeben, das die Hintergrundaufgabe steuert. Beispielsweise wird die POST-Anfrage zum Bereitstellen eines Clusters asynchron verarbeitet und gibt ein Job-Objekt zurück.
- Fehlerobjekt: Tritt ein Fehler auf, wird immer ein Fehlerobjekt zurückgegeben. Beispielsweise erhalten Sie einen Fehler, wenn Sie versuchen, einen Cluster mit einem Namen zu erstellen, der bereits existiert.
- In bestimmten Fällen werden keine Daten zurückgegeben und der Antworttext ist leer. Der Antworttext ist beispielsweise nach der Verwendung von DELETE zum Löschen eines vorhandenen Hosts leer.

Asynchrone Verarbeitung mit dem Job-Objekt für ONTAP Select

Einige Deploy-API-Aufrufe, insbesondere solche, die eine Ressource erstellen oder ändern, können länger dauern als andere Aufrufe. ONTAP Select Deploy verarbeitet diese lang andauernden Anfragen asynchron.

Asynchrone Anfragen werden mithilfe des Job-Objekts beschrieben.

Nach einem asynchronen API-Aufruf signalisiert der HTTP-Antwortcode 202, dass die Anfrage erfolgreich validiert und akzeptiert, aber noch nicht abgeschlossen wurde. Die Anfrage wird als Hintergrundaufgabe verarbeitet, die nach der ersten HTTP-Antwort an den Client weiterhin ausgeführt wird. Die Antwort enthält das Job-Objekt, das die Anfrage verankert, einschließlich seiner eindeutigen Kennung.

Sie sollten die Online-Dokumentationsseite von ONTAP Select Deploy konsultieren, um festzustellen, welche API-Aufrufe asynchron ausgeführt werden.

Abfrage des Job-Objekts, das einer API-Anfrage zugeordnet ist

Das in der HTTP-Antwort zurückgegebene Job-Objekt enthält mehrere Eigenschaften. Sie können die Eigenschaft state abfragen, um festzustellen, ob die Anfrage erfolgreich abgeschlossen wurde. Ein Job-Objekt kann sich in einem der folgenden Zustände befinden:

- In der Warteschlange
- Wird ausgeführt
- Erfolg
- Versagen

Es gibt zwei Techniken, die Sie beim Abfragen eines Job-Objekts verwenden können, um einen Endzustand für die Aufgabe zu erkennen, entweder Erfolg oder Misserfolg:

- Standard-Abfrage: Aktueller Jobstatus wird sofort zurückgegeben
- Der Auftragsstatus einer Long-Polling-Anfrage wird nur dann zurückgegeben, wenn eine der folgenden Bedingungen erfüllt ist:
  - Der Status hat sich nach dem im Abfragezeitpunkt angegebenen Datum/Uhrzeit geändert.
  - Zeitüberschreitungswert ist abgelaufen (1 bis 120 Sekunden)

Standard-Polling und Long-Polling verwenden denselben API-Aufruf, um ein Job-Objekt abzufragen. Eine Long-Polling-Anfrage enthält jedoch zwei Abfrageparameter: `poll_timeout` und `last_modified`.



Um die Arbeitslast auf der virtuellen Maschine Deploy zu reduzieren, sollten Sie stets Long Polling verwenden.

Allgemeines Verfahren zum Ausstellen einer asynchronen Anfrage

Sie können die folgende allgemeine Vorgehensweise verwenden, um einen asynchronen API-Aufruf durchzuführen:

1. Führe den asynchronen API-Aufruf aus.
2. Sie erhalten eine HTTP-Antwort 202, die die erfolgreiche Annahme der Anfrage signalisiert.
3. Extrahieren Sie die Kennung für das Job-Objekt aus dem Antworttext.
4. Führe innerhalb einer Schleife in jedem Durchlauf Folgendes aus:
   a. Ermitteln Sie den aktuellen Status des Jobs mit einer Long-Poll-Anfrage
   b. Befindet sich der Job in einem nicht-terminalen Zustand (in der Warteschlange, wird ausgeführt), führen Sie die Schleife erneut aus.
5. Beenden, wenn der Job einen Endzustand erreicht (Erfolg, Fehler).

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGENDEINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter http://www.netapp.com/TM aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.