



Konzepte

ONTAP Select

NetApp
January 29, 2026

Inhalt

Konzepte	1
REST-Webservices-Grundlage für die Bereitstellung und Verwaltung von ONTAP Select Clustern	1
Architektur und klassische Einschränkungen	1
Ressourcen und staatliche Vertretung	1
URI-Endpunkte	1
HTTP-Nachrichten	1
JSON-Formatierung	2
So greifen Sie auf die ONTAP Select Deploy API zu	2
Native Benutzeroberfläche des Bereitstellungsprogramms	2
ONTAP Select Deploy Online-Dokumentationsseite	2
Benutzerdefiniertes Programm	2
ONTAP Select Deploy API-Versionierung	2
Grundlegende Betriebsmerkmale der ONTAP Select Deploy API	3
Hypervisor-Host im Vergleich zu ONTAP Select Knoten	3
Objektkennungen	3
Anforderungskennungen	3
Synchrone und asynchrone Aufrufe	3
Bestätigen Sie den Abschluss eines lang laufenden Auftrags	4
Sicherheit	4
Anforderungs- und Antwort-API-Transaktion für ONTAP Select	4
Eingabeveriablen, die eine API-Anfrage steuern	4
Interpretieren einer API-Antwort	6
Asynchrone Verarbeitung mit dem Job-Objekt für ONTAP Select	8
Asynchrone Anforderungen, beschrieben mithilfe des Job-Objekts	8
Abfragen des mit einer API-Anforderung verknüpften Job-Objekts	8
Allgemeine Vorgehensweise zum Ausgeben einer asynchronen Anfrage	8

Konzepte

REST-Webservices-Grundlage für die Bereitstellung und Verwaltung von ONTAP Select Clustern

Representational State Transfer (REST) ist ein Stil zur Erstellung verteilter Webanwendungen. Bei der Entwicklung einer Webservice-API etabliert es eine Reihe von Technologien und Best Practices für die Bereitstellung serverbasierter Ressourcen und die Verwaltung ihrer Zustände. REST nutzt gängige Protokolle und Standards und bietet so eine flexible Grundlage für die Bereitstellung und Verwaltung von ONTAP Select Clustern.

Architektur und klassische Einschränkungen

REST wurde von Roy Fielding in seiner Doktorarbeit formal formuliert "[Dissertation](#)" an der UC Irvine im Jahr 2000. Es definiert einen Architekturstil durch eine Reihe von Einschränkungen, die zusammengenommen webbasierte Anwendungen und die zugrunde liegenden Protokolle verbessern. Die Einschränkungen etablieren eine RESTful-Webdienstanwendung basierend auf einer Client/Server-Architektur unter Verwendung eines zustandslosen Kommunikationsprotokolls.

Ressourcen und staatliche Vertretung

Ressourcen sind die grundlegenden Komponenten eines webbasierten Systems. Zu den ersten Entwurfsaufgaben beim Erstellen einer REST-Webdienstanwendung gehören:

- Identifizierung system- oder serverbasierter Ressourcen. Jedes System nutzt und verwaltet Ressourcen. Eine Ressource kann eine Datei, eine Geschäftstransaktion, ein Prozess oder eine Verwaltungseinheit 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 immer in einem von einer begrenzten Anzahl von Zuständen. Die Zustände sowie die zugehörigen Operationen, die zur Zustandsänderung verwendet werden, müssen klar definiert sein.

Zwischen Client und Server werden Nachrichten ausgetauscht, um auf die Ressourcen zuzugreifen und ihren Status gemäß dem allgemeinen CRUD-Modell (Create, Read, Update und Delete) zu ändern.

URI-Endpunkte

Jede REST-Ressource muss mithilfe eines klar definierten Adressierungsschemas definiert und bereitgestellt werden. Die Endpunkte, an denen die Ressourcen lokalisiert und identifiziert werden, verwenden einen Uniform Resource Identifier (URI). Der URI bietet einen allgemeinen Rahmen für die Erstellung eines eindeutigen Namens für jede Ressource im Netzwerk. Der Uniform Resource Locator (URL) ist ein URI-Typ, der in Webdiensten zur Identifizierung und zum Zugriff auf Ressourcen verwendet wird. Ressourcen werden typischerweise in einer hierarchischen Struktur ähnlich einem Dateiverzeichnis bereitgestellt.

HTTP-Nachrichten

HTTP (Hypertext Transfer Protocol) ist das Protokoll, das vom Webdienst-Client und -Server zum Austausch von Anforderungs- und Antwortnachrichten zu Ressourcen verwendet wird. Beim Entwurf einer Webdienstanwendung werden HTTP-Verben (wie GET und POST) den Ressourcen und den entsprechenden

Statusverwaltungsaktionen zugeordnet.

HTTP ist Zustandslos. Um eine Reihe zusammengehöriger Anfragen und Antworten einer Transaktion zuzuordnen, müssen daher zusätzliche Informationen in die HTTP-Header der Anfrage-/Antwort-Datenflüsse aufgenommen werden.

JSON-Formatierung

Informationen können auf verschiedene Weise strukturiert und zwischen Client und Server übertragen werden. Die beliebteste Option (und die mit der Deploy REST API verwendete) ist JavaScript Object Notation (JSON). JSON ist ein Industriestandard für die Darstellung einfacher Datenstrukturen in Klartext und wird zur Übertragung von Statusinformationen verwendet, die die Ressourcen beschreiben.

So greifen Sie auf die ONTAP Select Deploy API zu

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

Native Benutzeroberfläche des Bereitstellungsprogramms

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

ONTAP Select Deploy Online-Dokumentationsseite

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

Benutzerdefiniertes Programm

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

ONTAP Select Deploy API-Versionierung

Die in ONTAP Select Deploy enthaltene REST-API erhält eine Versionsnummer. Die API-Versionsnummer ist unabhängig von der Deploy-Versionsnummer. Sie sollten sich über die in Ihrer Deploy-Version enthaltene API-Version und die möglichen Auswirkungen auf Ihre Nutzung der API im Klaren sein.

Die aktuelle Version des Deploy-Verwaltungsprogramms enthält Version 3 der REST-API. Frühere Versionen des Deploy-Dienstprogramms enthalten die folgenden API-Versionen:

Bereitstellen von 2.8 und höher

ONTAP Select Deploy 2.8 und alle späteren Versionen enthalten Version 3 der REST-API.

Bereitstellen von 2.7.2 und früher

ONTAP Select Deploy 2.7.2 und alle früheren Versionen enthalten Version 2 der REST-API.



Die Versionen 2 und 3 der REST-API sind nicht kompatibel. Wenn Sie von einer früheren Version, die Version 2 der API enthält, auf Deploy 2.8 oder höher aktualisieren, müssen Sie den gesamten vorhandenen Code aktualisieren, der direkt auf die API zugreift, sowie alle Skripts, die die Befehlszeilenschnittstelle verwenden.

Grundlegende Betriebsmerkmale der ONTAP Select Deploy API

REST bietet zwar einen gemeinsamen Satz an Technologien und Best Practices, die Details der einzelnen APIs können jedoch je nach Designentscheidung variieren. Machen Sie sich vor der Verwendung der ONTAP Select Deploy API mit den Details und Betriebsmerkmalen vertraut.

Hypervisor-Host im Vergleich zu ONTAP Select Knoten

Ein Hypervisor-Host ist die zentrale Hardwareplattform, auf der eine virtuelle ONTAP Select Maschine gehostet wird. Wenn eine virtuelle ONTAP Select Maschine auf einem Hypervisor-Host bereitgestellt und aktiv ist, gilt sie als ONTAP Select-Knoten. Mit Version 3 der Deploy REST API sind Host- und Knotenobjekte 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.

Objektkennungen

Jede Ressourceninstanz oder jedes Objekt erhält bei der Erstellung eine eindeutige Kennung. Diese Kennungen sind innerhalb einer bestimmten Instanz von ONTAP Select Deploy global eindeutig. Nach dem Ausführen eines API-Aufrufs, der eine neue Objektinstanz erstellt, wird der zugehörige ID-Wert im folgenden Format an den Aufrufer zurückgegeben: location Header der HTTP-Antwort. Sie können die Kennung extrahieren und sie bei nachfolgenden Aufrufen verwenden, wenn Sie auf die Ressourceninstanz verweisen.



Inhalt und interne Struktur der Objektkennungen können sich jederzeit ändern. Sie sollten die Kennungen bei den entsprechenden API-Aufrufen nur nach Bedarf verwenden, wenn Sie auf die zugehörigen Objekte verweisen.

Anforderungskennungen

Jeder erfolgreichen API-Anfrage wird eine eindeutige Kennung zugewiesen. Die Kennung wird in der request-id Header der zugehörigen HTTP-Antwort. Mit einer Anforderungskennung können Sie die Aktivitäten einer einzelnen API-Anforderungs-Antwort-Transaktion kollektiv referenzieren. Beispielsweise können Sie alle Ereignismeldungen für eine Transaktion anhand der Anforderungs-ID abrufen.

Synchrone und asynchrone Aufrufe

Es gibt zwei Hauptmethoden, mit denen ein Server eine von einem Client empfangene HTTP-Anforderung 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 zeigt an, dass der Server die Clientanfrage akzeptiert und eine Hintergrundaufgabe zum Abschließen der Anfrage gestartet hat. Der endgültige Erfolg oder Misserfolg ist nicht sofort verfügbar und muss durch zusätzliche API-Aufrufe ermittelt werden.

Bestätigen Sie den Abschluss eines lang laufenden Auftrags

Im Allgemeinen werden Vorgänge, deren Ausführung längere Zeit in Anspruch nehmen kann, asynchron mithilfe einer Hintergrundaufgabe auf dem Server verarbeitet. Mit der Deploy-REST-API wird jede Hintergrundaufgabe durch ein Job-Objekt verankert, das die Aufgabe verfolgt und Informationen wie den aktuellen Status bereitstellt. Ein Job-Objekt mit 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 noch weitere Möglichkeiten, den Erfolg oder Misserfolg einer Anfrage zu bestimmen, darunter:

- Ereignismeldungen: Sie können alle Ereignismeldungen, die einem bestimmten API-Aufruf zugeordnet sind, mithilfe der Anforderungs-ID abrufen, die mit der ursprünglichen Antwort zurückgegeben wurde. Die Ereignismeldungen enthalten in der Regel einen Hinweis auf Erfolg oder Misserfolg und können auch beim Debuggen eines Fehlerzustands hilfreich sein.
- Ressourcenzustand oder -status Mehrere der Ressourcen verfügen über einen Zustands- oder Statuswert, den Sie abfragen können, um indirekt den Erfolg oder Misserfolg einer Anfrage zu bestimmen.

Sicherheit

Die Deploy-API verwendet die folgenden Sicherheitstechnologien:

- Transport Layer Security Der gesamte zwischen Deploy-Server und Client gesendete Netzwerkverkehr wird über TLS verschlüsselt. Die Verwendung des HTTP-Protokolls über einen unverschlüsselten Kanal wird nicht unterstützt. TLS Version 1.2 wird unterstützt.
- HTTP-Authentifizierung Für jede API-Transaktion wird die Basisauthentifizierung verwendet. Jeder Anfrage wird ein HTTP-Header hinzugefügt, der den Benutzernamen und das Kennwort in einer Base64-Zeichenfolge enthält.

Anforderungs- und Antwort-API-Transaktion für ONTAP Select

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

Eingabevervariablen, die eine API-Anfrage steuern

Sie können die Verarbeitung eines API-Aufrufs über in der HTTP-Anforderung festgelegte Parameter steuern.

Anforderungsheader

Sie müssen mehrere Header in die HTTP-Anforderung aufnehmen, darunter:

- Inhaltstyp: Wenn der Anforderungstext JSON enthält, muss dieser Header auf application/json gesetzt werden.
- Akzeptieren: Wenn der Antworttext JSON enthält, muss dieser Header auf application/json gesetzt werden.
- Autorisierung: Die Basisauthentifizierung muss mit dem Benutzernamen und dem Kennwort in einer Base64-Zeichenfolge codiert sein.

Anforderungstext

Der Inhalt des Anforderungstexts variiert je nach Aufruf. Der HTTP-Anforderungstext besteht aus einem der folgenden Elementen:

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

Filterobjekte

Wenn Sie einen API-Aufruf mit GET ausführen, können Sie die zurückgegebenen Objekte basierend auf einem beliebigen Attribut einschränken oder filtern. Sie können beispielsweise einen genauen Wert angeben, der übereinstimmen soll:

<field>=<query value>

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

Operator	Beschreibung
=	Gleich
<	Weniger als
>	Größer als
≤	Kleiner oder gleich
≥	Größer als oder gleich
	Oder
!	Ungleich
*	Gieriger Platzhalter

Sie können auch eine Reihe von Objekten zurückgeben, basierend darauf, ob ein bestimmtes Feld festgelegt ist oder nicht, indem Sie das Schlüsselwort „Null“ oder seine Negation (!null) als Teil der Abfrage verwenden.

Auswählen von Objektfeldern

Standardmäßig gibt ein API-Aufruf mit GET nur die Attribute zurück, die das Objekt bzw. die Objekte eindeutig identifizieren. Dieser Mindestsatz an Feldern dient als Schlüssel für jedes Objekt und variiert je nach Objekttyp. Sie können zusätzliche Objekteigenschaften mit dem Abfrageparameter „Felder“ wie folgt auswählen:

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

 Als Best Practice sollten Sie immer die gewünschten Felder angeben. Rufen Sie die kostengünstigen und teuren Felder nur bei Bedarf ab. Die Klassifizierung in kostengünstige und teure Felder wird von NetApp anhand interner Performanceanalysen festgelegt. Die Klassifizierung für ein bestimmtes Feld kann sich jederzeit ändern.

Sortieren von Objekten 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 Typfeld in absteigender Reihenfolge und anschließend die ID in aufsteigender Reihenfolge sortieren:

`order_by=type desc, id asc`

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

Pagination

Wenn Sie einen API-Aufruf mit GET ausführen, um auf eine Sammlung von Objekten desselben Typs zuzugreifen, werden standardmäßig alle übereinstimmenden Objekte zurückgegeben. Bei Bedarf können Sie die Anzahl der zurückgegebenen Datensätze mit dem Abfrageparameter `max_records` in der Anfrage begrenzen. Beispiel:

`max_records=20`

Bei Bedarf können Sie diesen Parameter mit anderen Abfrageparametern kombinieren, um das Ergebnis einzuschränken. Beispielsweise gibt die folgende Abfrage bis zu 10 Systemereignisse zurück, die nach der angegebenen Zeit generiert wurden:

`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

Sie können mehrere Anfragen stellen, um die Ereignisse (oder einen beliebigen Objekttyp) durchzublättern. Jeder nachfolgende API-Aufruf sollte einen neuen Zeitwert basierend auf dem letzten Ereignis im letzten Ergebnissatz verwenden.

Interpretieren einer API-Antwort

Jede API-Anfrage generiert eine Antwort an den Client. Sie können die Antwort überprüfen, um festzustellen, ob sie erfolgreich war, und bei Bedarf weitere Daten abrufen.

HTTP-Statuscode

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

Code	Bedeutung	Beschreibung
200	OK	Zeigt den Erfolg von Aufrufen an, die kein neues Objekt erstellen.
201	Erstellt	Ein Objekt wurde erfolgreich erstellt. Der Location-Antwortheader enthält die eindeutige Kennung für das Objekt.
202	Akzeptiert	Zur Ausführung der Anforderung wurde ein zeitintensiver Hintergrundjob gestartet, der Vorgang ist jedoch noch nicht abgeschlossen.
400	Ungültige Anforderung	Die Anfrageeingabe wird nicht erkannt oder ist unpassend.
403	Verboten	Der Zugriff wird aufgrund eines Autorisierungsfehlers verweigert.
404	Nicht gefunden	Die in der Anfrage genannte Ressource existiert nicht.
405	Methode nicht zulässig	Das HTTP-Verb in der Anforderung wird für die Ressource nicht unterstützt.
409	Konflikt	Der Versuch, ein Objekt zu erstellen, ist fehlgeschlagen, da das Objekt bereits vorhanden ist.
500	Interner Fehler	Auf dem Server ist ein allgemeiner interner Fehler aufgetreten.
501	Nicht implementiert	Die URI ist bekannt, kann die Anfrage jedoch nicht ausführen.

Antwortheader

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

- Anfrage-ID: Jeder erfolgreichen API-Anfrage wird eine eindeutige Anfragekennung zugewiesen.
- Standort: Wenn ein Objekt erstellt wird, enthält der Standort-Header die vollständige URL zum neuen Objekt einschließlich der eindeutigen Objektkennung.

Antworttext

Der Inhalt der mit einer API-Anfrage verknüpften Antwort unterscheidet sich je nach Objekt, Verarbeitungstyp und Erfolg oder Misserfolg der Anfrage. Der Antworttext wird in JSON gerendert.

- Einzelnes Objekt Ein einzelnes Objekt kann basierend auf der Anforderung mit einer Reihe von Feldern zurückgegeben werden. Beispielsweise können Sie GET verwenden, um ausgewählte Eigenschaften eines Clusters mithilfe der eindeutigen Kennung abzurufen.
- Mehrere Objekte Mehrere Objekte aus einer Ressourcensammlung können zurückgegeben werden. In allen Fällen wird ein einheitliches Format verwendet, mit `num_records` Gibt die Anzahl der Datensätze und Datensätze an, die ein Array der Objektinstanzen enthalten. Sie können beispielsweise alle in einem bestimmten Cluster definierten Knoten abrufen.
- Job-Objekt: Wenn ein API-Aufruf asynchron verarbeitet wird, wird ein Job-Objekt zurückgegeben, das die Hintergrundaufgabe verankert. Beispielsweise wird die POST-Anforderung zum Bereitstellen eines Clusters asynchron verarbeitet und gibt ein Job-Objekt zurück.
- Fehlerobjekt: Wenn ein Fehler auftritt, wird immer ein Fehlerobjekt zurückgegeben. Beispielsweise erhalten Sie eine Fehlermeldung, wenn Sie versuchen, einen Cluster mit einem bereits vorhandenen Namen zu erstellen.
- Leer: In bestimmten Fällen werden keine Daten zurückgegeben und der Antworttext ist leer. Beispielsweise ist der Antworttext leer, nachdem mit DELETE ein vorhandener Host gelöscht wurde.

Asynchrone Verarbeitung mit dem Job-Objekt für ONTAP Select

Einige Deploy-API-Aufrufe, insbesondere solche zum Erstellen oder Ändern einer Ressource, können länger dauern als andere Aufrufe. ONTAP Select Deploy verarbeitet diese lang laufenden Anfragen asynchron.

Asynchrone Anforderungen, beschrieben mithilfe des Job-Objekts

Nach einem asynchron ausgeführten API-Aufruf zeigt der HTTP-Antwortcode 202 an, 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 weiterläuft. Die Antwort enthält das Job-Objekt, das die Anfrage verankert, einschließlich seiner eindeutigen Kennung.



Um zu ermitteln, welche API-Aufrufe asynchron ausgeführt werden, sollten Sie auf der Online-Dokumentationsseite von ONTAP Select Deploy nachsehen.

Abfragen des mit einer API-Anforderung verknüpften Job-Objekts

Das in der HTTP-Antwort zurückgegebene Job-Objekt enthält mehrere Eigenschaften. Sie können die Status-eigenschaft abfragen, um festzustellen, ob die Anforderung erfolgreich abgeschlossen wurde. Ein Job-Objekt kann einen der folgenden Zustände aufweisen:

- 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-Polling-Anfrage Aktueller Jobstatus wird sofort zurückgegeben
- Der Auftragsstatus für lange Polling-Anfragen wird nur zurückgegeben, wenn eines der folgenden Ereignisse eintritt:
 - Der Status hat sich vor dem in der Abfrageanforderung angegebenen Datums-/Uhrzeitwert geändert.
 - Timeout-Wert ist abgelaufen (1 bis 120 Sekunden)

Standard-Polling und Long-Polling verwenden denselben API-Aufruf zum Abfragen eines Job-Objekts. Eine Long-Polling-Anforderung enthält jedoch zwei Abfrageparameter: `poll_timeout` Und `last_modified`.



Sie sollten immer Long Polling verwenden, um die Arbeitslast auf der virtuellen Bereitstellungsmaschine zu reduzieren.

Allgemeine Vorgehensweise zum Ausgeben einer asynchronen Anfrage

Sie können das folgende allgemeine Verfahren verwenden, um einen asynchronen API-Aufruf abzuschließen:

1. Führen Sie den asynchronen API-Aufruf aus.

2. Erhalten Sie eine HTTP-Antwort 202, die die erfolgreiche Annahme der Anfrage anzeigen.
3. Extrahieren Sie die Kennung für das Job-Objekt aus dem Antworttext.
4. Führen Sie innerhalb einer Schleife in jedem Zyklus Folgendes aus:
 - a. Holen Sie sich den aktuellen Status des Jobs mit einer Long-Poll-Anfrage
 - b. Wenn sich der Job in einem nicht-terminalen Zustand befindet (in der Warteschlange, läuft), führen Sie die Schleife erneut aus.
5. Stoppen Sie, wenn der Job einen Endzustand (Erfolg, Fehler) erreicht.

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 BESCHRÄNKKT 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 BESCHRÄNKKT 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 IRGENDERINER 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.