Swift REST API verwenden (veraltet)

StorageGRID 11.8

NetApp
March 19, 2024

Inhalt

Swift REST API verwenden (veraltet)

Übersicht über die Swift REST API

Client-Applikationen können die OpenStack Swift API zur Schnittstelle mit dem StorageGRID System nutzen.

Die Unterstützung für Swift-Client-Anwendungen wurde veraltet und wird in einer zukünftigen Version entfernt.

StorageGRID unterstützt die folgenden spezifischen Versionen von Swift und HTTP.

Element	Version
Swift-Spezifikation	OpenStack Swift Objekt Storage API v1 ab November 2015
HTTP	1.1 Weitere Informationen zu HTTP finden Sie unter HTTP/1.1 (RFCs 7230-35). **Hinweis**: StorageGRID unterstützt HTTP/1.1-Pipelining nicht.

Verwandte Informationen

"OpenStack: Objekt-Storage-API"

Geschichte der Unterstützung von Swift API in StorageGRID

Bei Änderungen an der Unterstützung des StorageGRID-Systems für die Swift REST-API sollten Sie auf dieser hinweisen.

Freigabe	Kommentare
11.8	
11.7	Die Unterstützung für Swift-Client-Anwendungen wurde veraltet und wird in einer zukünftigen Version entfernt.
11.6	Kleine redaktionelle Änderungen.
11.5	Schwache Konsistenz wurde entfernt. Stattdessen wird die verfügbare Konsistenz verwendet.
11.4	Unterstützung für TLS 1.3 hinzugefügt. Beschreibung des Zusammenhangs zwischen ILM und Konsistenz hinzugefügt.

Freigabe	Kommentare
11.3	Aktualisierte PUT-Objektvorgänge zur Beschreibung der Auswirkungen von ILM-Regeln, die synchrone Platzierung bei der Aufnahme verwenden (die ausgewogenen und strengen Optionen für das Aufnahmeverhalten) Eine zusätzliche Beschreibung der Client-Verbindungen, die Load Balancer-Endpunkte oder Hochverfügbarkeitsgruppen verwenden. TLS 1.1-Chiffren werden nicht mehr unterstützt.
11.2	Kleine redaktionelle Änderungen des Dokuments.
11.1	Zusätzlicher Support für die Verwendung von HTTP für Swift-Client-Verbindungen zu Grid-Nodes. Die Definitionen der Konsistenzwerte wurden aktualisiert.
11.0	Hinzugefügter Support für 1,000 Container für jedes Mandantenkonto.
10.3	Administrative Aktualisierungen und Korrekturen des Dokuments. Abschnitte zum Konfigurieren von benutzerdefinierten Serverzertifikaten entfernt.
10.2	Unterstützung der Swift API durch das StorageGRID System zu Beginn. Die derzeit unterstützte Version ist OpenStack Swift Object Storage API v1.

So implementiert StorageGRID Swift REST API

Eine Client-Applikation kann mithilfe von Swift REST-API-Aufrufen eine Verbindung zu Storage-Nodes und Gateway-Nodes herstellen, um Container zu erstellen und Objekte zu speichern und abzurufen. Dadurch können serviceorientierte Applikationen, die für OpenStack Swift entwickelt wurden, mit lokalem Objekt-Storage des StorageGRID Systems verbunden werden.

Swift Objekt-Management

Nachdem Swift Objekte in das StorageGRID System aufgenommen wurden, werden sie durch die Regeln für Information Lifecycle Management (ILM) in den aktiven ILM-Richtlinien gemanagt. "ILM-Regeln" Und "ILM-Richtlinien" Legen Sie fest, wie StorageGRID Kopien von Objektdaten erstellt und verteilt und wie diese Kopien über einen längeren Zeitraum gemanagt werden. Eine ILM-Regel kann beispielsweise für Objekte in bestimmten Swift Containern gelten und möglicherweise angeben, dass mehrere Objektkopien für eine bestimmte Anzahl von Jahren in mehreren Datacentern gespeichert werden.

Wenden Sie sich an Ihren NetApp Professional Services Berater oder StorageGRID Administrator, wenn Sie Informationen darüber benötigen, wie sich die ILM-Regeln und -Richtlinien des Grids auf die Objekte in Ihrem Swift Mandantenkonto auswirken.

In Konflikt stehende Clientanforderungen

Widersprüchliche Clientanforderungen, wie z. B. zwei Clients, die in denselben Schlüssel schreiben, werden auf der Grundlage der „neuesten Wins“ gelöst. Der Zeitpunkt für die Bewertung „neuester Erfolge“ basiert darauf, wann das StorageGRID System eine bestimmte Anfrage abschließt und nicht darauf, wann Swift-Clients einen Vorgang starten.

Konsistenzgarantien und -Kontrollen

Standardmäßig bietet StorageGRID Lese-/Nachher-Konsistenz für neu erstellte Objekte und schließlich die Konsistenz von Objekt-Updates und HEAD-Operationen. Alle "GET" Nach erfolgreichem Abschluss "PUT" Kann die neu geschriebenen Daten lesen. Überschreibungen vorhandener Objekte, Metadatenaktualisierungen und -Löschungen sind schließlich konsistent. Überschreibungen dauern in der Regel nur wenige Sekunden oder Minuten, können jedoch bis zu 15 Tage in Anspruch nehmen.

StorageGRID ermöglicht Ihnen außerdem die Kontrolle der Konsistenz einzelner Container. Konsistenzwerte bieten ein Gleichgewicht zwischen der Verfügbarkeit der Objekte und der Konsistenz dieser Objekte über verschiedene Storage Nodes und Standorte hinweg, wie es von Ihrer Anwendung gefordert wird.

Empfehlungen für die Implementierung von Swift REST API

Bei der Implementierung der Swift REST API zur Verwendung mit StorageGRID sollten Sie diese Empfehlungen beachten.

Empfehlungen für Köpfe zu nicht vorhandenen Objekten

Wenn Ihre Anwendung regelmäßig prüft, ob ein Objekt in einem Pfad vorhanden ist, in dem Sie nicht erwarten, dass das Objekt tatsächlich existiert, sollten Sie die Konsistenz „verfügbar“ verwenden. Sie sollten beispielsweise die Konsistenz „verfügbar“ verwenden, wenn Ihre Anwendung einen HAUPTVORGANG an einem Speicherort durchführt, bevor Sie einen PUT-Vorgang an diesem Speicherort durchführen.

Andernfalls werden möglicherweise 500 Fehler des internen Servers angezeigt, wenn ein oder mehrere Speicherknoten nicht verfügbar sind.

Sie können die Konsistenz für jeden Container mithilfe von festlegen "PUT Container-Konsistenzanforderung". Sie können die Konsistenz „verfügbar“ für jeden Container mithilfe von anzeigen "ABRUFEN der Container-Konsistenzanforderung".

Empfehlungen für Objektnamen

Bei Containern, die in StorageGRID 11.4 oder höher erstellt wurden, ist keine Beschränkung der Objektnamen auf die Performance-Best Practices mehr erforderlich. Sie können jetzt beispielsweise Zufallswerte für die ersten vier Zeichen von Objektnamen verwenden.

Befolgen Sie bei Containern, die in Versionen vor StorageGRID 11.4 erstellt wurden, weiterhin diese Empfehlungen für Objektnamen:

- Als die ersten vier Zeichen von Objektnamen sollten keine Zufallswerte verwendet werden. Dies steht im Gegensatz zu der früheren AWS Empfehlung für Namenspräfixe. Stattdessen sollten Sie nicht-zufällige, nicht-eindeutige Präfixe verwenden, wie z. B. `image`.
- Wenn Sie die frühere Empfehlung von AWS befolgen, zufällige und eindeutige Zeichen in Namenspräfixen zu verwenden, sollten Sie die Objektnamen mit einem Verzeichnisnamen vorschreiben. Verwenden Sie dieses Format:

  ```
  mycontainer/mydir/f8e3-image3132.jpg
  ```

  Anstelle dieses Formats:

```
mycontainer/f8e3-image3132.jpg
```

Empfehlungen für „Range Reads“

Wenn der "Globale Option zum Komprimieren gespeicherter Objekte" Ist aktiviert, sollten Swift-Client-Anwendungen die Ausführung VON GET-Objektoperationen vermeiden, die einen Bereich von Bytes angeben, die zurückgegeben werden sollen. Diese Vorgänge beim Lesen von Range sind ineffizient, da StorageGRID Objekte effektiv dekomprimieren muss, um auf die angeforderten Bytes zuzugreifen. GET-Objektvorgänge, die einen kleinen Byte-Bereich von einem sehr großen Objekt anfordern, sind besonders ineffizient, beispielsweise ist es sehr ineffizient, einen Bereich von 10 MB von einem komprimierten 50-GB-Objekt zu lesen.

Wenn Bereiche von komprimierten Objekten gelesen werden, können Client-Anforderungen eine Zeitdauer haben.

> Wenn Sie Objekte komprimieren müssen und Ihre Client-Applikation Bereichslesevorgänge verwenden muss, erhöhen Sie die Zeitüberschreitung beim Lesen der Anwendung.

Testen der REST API-Konfiguration von Swift

Sie können die Swift CLI verwenden, um die Verbindung zum StorageGRID System zu testen und zu überprüfen, ob Objekte gelesen und geschrieben werden können.

Bevor Sie beginnen

- Sie haben den Swift-Befehlszeilenclient heruntergeladen und installiert: "SwiftStack: python-wiftclient"
- Optional haben Sie "Ein Load Balancer-Endpunkt wurde erstellt". Andernfalls kennen Sie die IP-Adresse des zu verbindenden Storage-Node und die zu verwendende Port-Nummer. Siehe "IP-Adressen und Ports für Client-Verbindungen".
- Das ist schon "Swift Mandantenkonto erstellt".
- Sie haben sich beim Mandantenkonto angemeldet und mindestens eine Gruppe und einen Benutzer erstellt. Siehe "Erstellen von Gruppen für einen Swift Mandanten".

> Swift-Mandanten-Benutzer müssen über die Administratorgruppe verfügen, um sich bei der Swift-REST-API authentifizieren zu können.

Über diese Aufgabe

Wenn Sie keine Sicherheit konfiguriert haben, müssen Sie die hinzufügen `--insecure` Flag auf jeden dieser Befehle.

Schritte

1. Fragen Sie die Info-URL für Ihre StorageGRID Swift Implementierung:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/info
capabilities
```

Dies reicht aus, um zu testen, ob Ihre Swift-Implementierung funktionsfähig ist. Um die Kontenkonfiguration durch Speichern eines Objekts weiter zu testen, fahren Sie mit den zusätzlichen Schritten fort.

2. Legen Sie ein Objekt in den Container:

```
touch test_object
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
upload test_container test_object
--object-name test_object
```

3. Holen Sie sich den Container, um das Objekt zu überprüfen:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
list test_container
```

4. Löschen Sie das Objekt:

```
swift
-U <Tenant_Account_ID:Account_User_Name>
-K <User_Password>
-A https://<FQDN | IP>:<Port>/auth/v1.0
delete test_container test_object
```

5. Löschen Sie den Container:

```
swift
-U `<_Tenant_Account_ID:Account_User_Name_>`
-K `<_User_Password_>`
-A `\https://<_FQDN_ | _IP_>:<_Port_>/auth/v1.0'
delete test_container
```

Von Swift UNTERSTÜTZTE REST-API-Operationen

Das StorageGRID System unterstützt die meisten Operationen in der OpenStack Swift API. Informieren Sie sich vor der Integration von Swift REST API Clients mit StorageGRID über die Implementierungsdetails für Konto-, Container- und Objektvorgänge.

Von StorageGRID unterstützte Vorgänge

Die folgenden Swift-API-Operationen werden unterstützt:

- "Konto-Operationen"
- "Container-Operationen"
- "Objekt-Operationen"

Gemeinsame Antwortheader für alle Vorgänge

Das StorageGRID-System implementiert alle gemeinsamen Header für unterstützte Vorgänge, wie sie von der OpenStack Swift Objekt-Storage-API v1 definiert wurden.

Verwandte Informationen

"OpenStack: Objekt-Storage-API"

Unterstützte Swift-API-Endpunkte

StorageGRID unterstützt die folgenden Swift-API-Endpunkte: Die Info-URL, die auth-URL und die Storage-URL.

Info-URL

Sie können die Funktionen und Einschränkungen der StorageGRID-Swift-Implementierung bestimmen, indem Sie eine GET-Anfrage an die Swift-Basis-URL mit dem /info-Pfad senden.

```
https://FQDN | Node IP:Swift Port/info/
```

In der Anfrage:

- *FQDN* Ist der vollständig qualifizierte Domain-Name.
- *Node IP* Ist die IP-Adresse für den Storage-Node oder den Gateway-Node im StorageGRID-Netzwerk.
- *Swift Port* Ist die Portnummer, die für Swift-API-Verbindungen auf dem Storage-Node oder Gateway-Node verwendet wird.

Die folgende Info-URL würde beispielsweise Informationen von einem Storage-Node mit der IP-Adresse von 10.99.106.103 anfordern und Port 18083 verwenden.

```
https://10.99.106.103:18083/info/
```

Die Antwort umfasst die Funktionen der Swift-Implementierung als JSON-Wörterbuch. Ein Client-Tool kann die JSON-Antwort analysieren, um die Funktionen der Implementierung zu bestimmen und sie als Einschränkungen für nachfolgende Storage-Vorgänge zu verwenden.

Die StorageGRID-Implementierung von Swift ermöglicht nicht authentifizierten Zugriff auf die Info-URL.

Auth-URL

Ein Client kann die Swift auth URL verwenden, um sich als Benutzer eines Mandantenkontos zu authentifizieren.

`https://FQDN | Node IP:Swift Port/auth/v1.0/`

Sie müssen die Mandanten-Konto-ID, den Benutzernamen und das Passwort als Parameter in angeben `X-Auth-User` Und `X-Auth-Key` Anforderungs-Header wie folgt:

`X-Auth-User: Tenant_Account_ID:Username`

`X-Auth-Key: Password`

In den Kopfzeilen der Anfrage:

- *`Tenant_Account_ID`* Ist die Account-ID, die StorageGRID beim Erstellen des Swift-Mandanten zugewiesen hat. Dies ist die gleiche Mandantenkonto-ID, die auf der Anmeldeseite des Mandanten-Managers verwendet wird.
- *`Username`* Ist der Name eines im Mandanten-Manager erstellten Benutzers. Dieser Benutzer muss einer Gruppe angehören, die über die Swift Administrator-Berechtigung verfügt. Der Root-Benutzer des Mandanten kann nicht für die Verwendung der Swift REST API konfiguriert werden.

  Wenn Identity Federation für das Mandantenkonto aktiviert ist, geben Sie den Benutzernamen und das Passwort des föderierten Benutzers vom LDAP-Server an. Geben Sie alternativ den Domänennamen des LDAP-Benutzers an. Beispiel:

  `X-Auth-User: Tenant_Account_ID:Username@Domain_Name`

- *`Password`* Ist das Passwort für den Mandantenbenutzer. Benutzerpasswörter werden im Mandanten-Manager erstellt und gemanagt.

Als Antwort auf eine erfolgreiche Authentifizierungsanforderung werden eine Storage-URL und ein auth-Token zurückgegeben:

`X-Storage-Url: https://FQDN | Node_IP:Swift_Port/v1/Tenant_Account_ID`

`X-Auth-Token: token`

`X-Storage-Token: token`

Das Token ist standardmäßig für 24 Stunden ab der Erzeugung gültig.

Token werden für ein bestimmtes Mandantenkonto generiert. Ein gültiges Token für ein Konto ermächtigt einen Benutzer nicht, auf ein anderes Konto zuzugreifen.

Storage-URL

Eine Client-Applikation kann Swift-REST-API-Aufrufe ausstellen, um unterstützte Konto-, Container- und Objektvorgänge mit einem Gateway-Node oder Storage-Node durchzuführen. Storage-Anforderungen werden an die in der Authentifizierungsantwort zurückgegebene Storage-URL adressiert. Die Anforderung muss auch

die Kopfzeile von X-Auth-Token und den Wert enthalten, der von der auth-Anforderung zurückgegeben wurde.

```
https://FQDN | IP:Swift_Port/v1/Tenant_Account_ID

[/container][/object]

X-Auth-Token: token
```

Einige Kopf für Speicherantwort, die Nutzungsstatistiken enthalten, geben möglicherweise keine genauen Zahlen für kürzlich geänderte Objekte wieder. Es kann einige Minuten dauern, bis genaue Zahlen in diesen Kopfzeilen angezeigt werden.

Die folgenden Antwortkopfzeilen für Konto- und Container-Vorgänge sind Beispiele für solche, die Nutzungsstatistiken enthalten:

- `X-Account-Bytes-Used`
- `X-Account-Object-Count`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`

Verwandte Informationen

"Mandantenkonten und -Verbindungen konfigurieren"

"Konto-Operationen"

"Container-Operationen"

"Objekt-Operationen"

Konto-Operationen

Die folgenden Swift-API-Vorgänge werden bei Accounts durchgeführt.

GET Konto

Dieser Vorgang ruft die Containerliste ab, die mit den Statistiken zur Konto- und Kontonutzung verknüpft ist.

Der folgende Parameter für die Anfrage ist erforderlich:

- `Account`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Die folgenden unterstützten Abfrageparameter sind optional:

- `Delimiter`
- `End_marker`
- `Format`

- `Limit`
- `Marker`
- `Prefix`

Bei einer erfolgreichen Ausführung werden die folgenden Header mit einer Antwort „HTTP/1.1 204 No Content“ zurückgegeben, wenn das Konto gefunden wird und keine Container hat oder die Containerliste leer ist; oder eine „HTTP/1.1 200 OK“-Antwort, wenn das Konto gefunden wird und die Containerliste nicht leer ist:

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

HAUPTKONTO

Mit dieser Operation werden Kontoinformationen und Statistiken von einem Swift-Konto abgerufen.

Der folgende Parameter für die Anfrage ist erforderlich:

- `Account`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Eine erfolgreiche Ausführung gibt die folgenden Kopfzeilen mit einer HTTP/1.1 204 No Content-Antwort zurück:

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Account-Bytes-Used`
- `X-Account-Container-Count`
- `X-Account-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

Verwandte Informationen

"In den Audit-Protokollen werden Swift-Vorgänge nachverfolgt"

Container-Operationen

StorageGRID unterstützt maximal 1,000 Container pro Swift Konto. Die folgenden Swift-API-Vorgänge werden auf Containern durchgeführt.

Container LÖSCHEN

Durch diesen Vorgang wird ein leerer Container aus einem Swift-Konto in einem StorageGRID-System entfernt.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Eine erfolgreiche Ausführung gibt die folgenden Kopfzeilen mit einer HTTP/1.1 204 No Content-Antwort zurück:

- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

GET Container

Dieser Vorgang ruft die dem Container zugeordnete Objektliste sowie die Containerstatistiken und Metadaten in einem StorageGRID System ab.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Die folgenden unterstützten Abfrageparameter sind optional:

- `Delimiter`
- `End_marker`
- `Format`
- `Limit`

- `Marker`
- `Path`
- `Prefix`

Eine erfolgreiche Ausführung liefert die folgenden Header mit einer "HTTP/1.1 200 success" oder einer "HTTP/1.1 204 No Content"-Antwort:

- `Accept-Ranges`
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

KOPF Behälter

Dieser Vorgang ruft Containerstatistiken und Metadaten aus einem StorageGRID System ab.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Eine erfolgreiche Ausführung gibt die folgenden Kopfzeilen mit einer HTTP/1.1 204 No Content-Antwort zurück:

- `Accept-Ranges`
- `Content-Length`
- `Date`
- `X-Container-Bytes-Used`
- `X-Container-Object-Count`
- `X-Timestamp`
- `X-Trans-Id`

Legen Sie den Behälter

Durch diesen Vorgang wird ein Container für ein Konto in einem StorageGRID-System erstellt.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Eine erfolgreiche Ausführung gibt die folgenden Header mit einer "HTTP/1.1 201 created" oder "HTTP/1.1 202 Accepted" (falls der Container bereits unter diesem Konto existiert) Antwort zurück:

- `Content-Length`
- `Date`
- `X-Timestamp`
- `X-Trans-Id`

Container-Name muss im StorageGRID-Namespace eindeutig sein. Wenn der Container unter einem anderen Konto vorhanden ist, wird der folgende Header zurückgegeben: „HTTP/1.1 409-Konflikt“.

Verwandte Informationen

"Monitoring und Prüfung von Vorgängen"

Objekt-Operationen

Die folgenden Swift-API-Vorgänge werden an Objekten durchgeführt. Diese Vorgänge können im nachverfolgt werden "StorageGRID Prüfprotokoll".

Delete Objekt

Durch diesen Vorgang werden der Inhalt und die Metadaten eines Objekts aus dem StorageGRID System gelöscht.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`
- `Object`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Bei einer erfolgreichen Ausführung werden die folgenden Antwortheadern mit einem zurückgegeben `HTTP/1.1 204 No Content` Antwort:

- `Content-Length`
- `Content-Type`

- `Date`
- `X-Trans-Id`

Bei der Verarbeitung einer LÖSCHOBJEKTANFORDERUNG versucht StorageGRID, alle Kopien des Objekts sofort von allen gespeicherten Speicherorten zu entfernen. Wenn erfolgreich, gibt StorageGRID sofort eine Antwort an den Client zurück. Wenn nicht innerhalb von 30 Sekunden alle Kopien entfernt werden können (z. B. weil ein Speicherort vorübergehend nicht verfügbar ist), stellt StorageGRID die Kopien in eine Warteschlange zur Entfernung und zeigt dann den Erfolg des Clients an.

Weitere Informationen finden Sie unter "So werden Objekte gelöscht".

GET Objekt

Dieser Vorgang ruft den Objektinhalt ab und ruft die Objektmetadaten von einem StorageGRID System ab.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`
- `Object`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Die folgenden Anfragezeilen sind optional:

- `Accept-Encoding`
- `If-Match`
- `If-Modified-Since`
- `If-None-Match`
- `If-Unmodified-Since`
- `Range`

Bei einer erfolgreichen Ausführung werden die folgenden Kopfzeilen mit einem zurückgegeben `HTTP/1.1 200 OK` Antwort:

- `Accept-Ranges`
- `Content-Disposition`, Nur wenn zurückgegeben `Content-Disposition` Es wurden Metadaten festgelegt
- `Content-Encoding`, Nur wenn zurückgegeben `Content-Encoding` Es wurden Metadaten festgelegt
- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`

- `Last-Modified`
- `X-Timestamp`
- `X-Trans-Id`

HEAD-Objekt

Dieser Vorgang ruft Metadaten und Eigenschaften eines aufgenommene Objekts von einem StorageGRID System ab.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`
- `Object`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Eine erfolgreiche Ausführung gibt die folgenden Header mit einer HTTP/1.1 200 OK-Antwort zurück:

- `Accept-Ranges`
- `Content-Disposition`, Nur wenn zurückgegeben `Content-Disposition` Es wurden Metadaten festgelegt
- `Content-Encoding`, Nur wenn zurückgegeben `Content-Encoding` Es wurden Metadaten festgelegt
- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Timestamp`
- `X-Trans-Id`

PUT Objekt

Durch diesen Vorgang wird ein neues Objekt mit Daten und Metadaten erstellt oder ein vorhandenes Objekt durch Daten und Metadaten in einem StorageGRID System ersetzt.

StorageGRID unterstützt Objekte mit einer Größe von bis zu 5 tib (5,497,558,138,880 Byte).

> Widersprüchliche Clientanforderungen, wie z. B. zwei Clients, die in denselben Schlüssel schreiben, werden auf der Grundlage der „neuesten Wins“ gelöst. Der Zeitpunkt für die Bewertung „neuester Erfolge“ basiert darauf, wann das StorageGRID System eine bestimmte Anfrage abschließt und nicht darauf, wann Swift-Clients einen Vorgang starten.

Die folgenden Anfrageparameter sind erforderlich:

- `Account`
- `Container`
- `Object`

Die folgende Anfrageüberschrift ist erforderlich:

- `X-Auth-Token`

Die folgenden Anfragezeilen sind optional:

- `Content-Disposition`
- `Content-Encoding`

  Verwenden Sie keine Schrottbecherungen `Content-Encoding` Wenn die ILM-Regel für ein Objekt Objekte nach der Größe filtert und synchrone Platzierung bei der Aufnahme verwendet wird (die ausgewogenen oder strengen Optionen für das Aufnahmeverhalten).

- `Transfer-Encoding`

  Verwenden Sie keine komprimierten oder chunked `Transfer-Encoding` Wenn die ILM-Regel für ein Objekt Objekte nach der Größe filtert und synchrone Platzierung bei der Aufnahme verwendet wird (die ausgewogenen oder strengen Optionen für das Aufnahmeverhalten).

- `Content-Length`

  Wenn eine ILM-Regel Objekte nach Größe filtert und bei der Aufnahme synchrone Platzierung verwendet, müssen Sie angeben `Content-Length`.

  > Wenn Sie diese Richtlinien für nicht befolgen `Content-Encoding`, `Transfer-Encoding`, und `Content-Length`, StorageGRID muss das Objekt speichern, bevor es die Objektgröße bestimmen kann und die ILM-Regel anwenden kann. Das heißt, StorageGRID muss standardmäßig vorläufige Kopien eines Objekts bei der Aufnahme erstellen. Das heißt, StorageGRID muss die Dual-Commit-Option für das Ingest-Verhalten verwenden.

  Weitere Informationen zur synchronen Platzierung und zu ILM-Regeln finden Sie unter "Datensicherungsoptionen für die Aufnahme".

- `Content-Type`
- `ETag`
- `X-Object-Meta-<name\>` (Objektbezogene Metadaten)

  Wenn Sie die Option **User Defined Creation Time** als Referenzzeit für eine ILM-Regel verwenden möchten, müssen Sie den Wert in einem benutzerdefinierten Header namens speichern `X-Object-Meta-Creation-Time`. Beispiel:

  ```
  X-Object-Meta-Creation-Time: 1443399726
  ```

  Dieses Feld wird seit dem 1. Januar 1970 als Sekunden ausgewertet.

- `X-Storage-Class: reduced_redundancy`

  Diese Kopfzeile wirkt sich darauf aus, wie viele Objektkopien StorageGRID erstellt werden, wenn die ILM-Regel, die mit einem aufgenommenen Objekt übereinstimmt, ein Aufnahmeverhalten der Dual-Commit oder Balance angibt.

  - **Dual Commit**: Wenn die ILM-Regel die Dual Commit-Option für das Aufnahmeverhalten angibt, erstellt StorageGRID bei Aufnahme des Objekts eine einzelne Interimskopie (Single Commit).
  - **Ausgeglichen**: Wenn die ILM-Regel die Option ausgeglichen angibt, erstellt StorageGRID nur eine Zwischenkopie, wenn das System nicht sofort alle in der Regel angegebenen Kopien erstellen kann. Wenn StorageGRID eine synchrone Platzierung durchführen kann, hat diese Kopfzeile keine Auswirkung.

    Der `reduced_redundancy` Kopfzeile eignet sich am besten, wenn die ILM-Regel, die dem Objekt entspricht, eine einzige replizierte Kopie erstellt. In diesem Fall verwenden `reduced_redundancy` Eine zusätzliche Objektkopie kann bei jedem Aufnahmevorgang nicht mehr erstellt und gelöscht werden.

    Verwenden der `reduced_redundancy` Header wird unter anderen Umständen nicht empfohlen, da dies das Risiko für den Verlust von Objektdaten während der Aufnahme erhöht. Beispielsweise können Sie Daten verlieren, wenn die einzelne Kopie zunächst auf einem Storage Node gespeichert wird, der ausfällt, bevor eine ILM-Evaluierung erfolgen kann.

    > ⚠ Da nur eine Kopie zu einem beliebigen Zeitpunkt repliziert werden kann, sind Daten einem ständigen Verlust ausgesetzt. Wenn nur eine replizierte Kopie eines Objekts vorhanden ist, geht dieses Objekt verloren, wenn ein Speicherknoten ausfällt oder einen beträchtlichen Fehler hat. Während Wartungsarbeiten wie Upgrades verlieren Sie auch vorübergehend den Zugriff auf das Objekt.

  Beachten Sie, dass Sie angeben `reduced_redundancy` Wirkt sich nur darauf aus, wie viele Kopien erstellt werden, wenn ein Objekt zum ersten Mal aufgenommen wird. Es hat keine Auswirkungen darauf, wie viele Kopien des Objekts erstellt werden, wenn das Objekt durch die aktiven ILM-Richtlinien evaluiert wird. Außerdem werden Daten nicht mit niedrigerer Redundanz im StorageGRID System gespeichert.

Eine erfolgreiche Ausführung gibt die folgenden Header mit einer "HTTP/1.1 201 created"-Antwort zurück:

- `Content-Length`
- `Content-Type`
- `Date`
- `ETag`
- `Last-Modified`
- `X-Trans-Id`

OPTIONEN anfordern

Die OPTIONEN Request überprüft die Verfügbarkeit eines einzelnen Swift Service. Die OPTIONSANFORDERUNG wird vom in der URL angegebenen Speicherknoten oder Gateway-Node verarbeitet.

OPTIONEN

Client-Anwendungen können zum Beispiel eine OPTIONSANFORDERUNG an den Swift-Port auf einem Storage Node stellen, ohne Swift-Authentifizierungsdaten bereitzustellen, um zu ermitteln, ob der Storage-Node verfügbar ist. Sie können diese Anforderung zum Monitoring verwenden oder um externen Lastausgleich zu ermöglichen, wenn ein Storage-Node ausfällt.

Bei Verwendung mit der Info-URL oder der Speicher-URL gibt die OPTIONSMETHODE eine Liste der unterstützten Verben für die angegebene URL zurück (z. B. KOPF, GET, OPTIONEN und PUT). Die OPTIONSMETHODE kann nicht mit der AuthentifizURL verwendet werden.

Der folgende Parameter für die Anfrage ist erforderlich:

- `Account`

Die folgenden Anfrageparameter sind optional:

- `Container`
- `Object`

Bei einer erfolgreichen Ausführung werden die folgenden Header mit der Antwort „HTTP/1.1 204 No Content“ zurückgegeben. Für die ANFORDERUNG VON OPTIONEN an die Speicher-URL ist nicht erforderlich, dass das Ziel vorhanden ist.

- `Allow` (Eine Liste der unterstützten Verben für die angegebene URL, z. B. „KOPF“, „ABRUFEN“, „OPTIONEN“, Und PUT)
- `Content-Length`
- `Content-Type`
- `Date`
- `X-Trans-Id`

Verwandte Informationen

"Unterstützte Swift-API-Endpunkte"

Fehlerantworten bei Swift-API-Operationen

Das Verständnis möglicher Fehlerantworten kann Ihnen bei der Fehlerbehebung helfen.

Wenn während eines Vorgangs Fehler auftreten, werden möglicherweise die folgenden HTTP-Statuscodes zurückgegeben:

Swift-Fehlername	HTTP-Status
AccountNameTooLong, ContainerNameTooLong, HeaderTooBig, InvalidContainerName, InvalidRequest, InvalidURI, MetadataNameTooLong, MetadaValueTooBig, MissingSecurityHeader, ObjectNameTooLong, TooManyContainers, TooManyMetadataItems, TotalMetadaTooLarge	400 Fehlerhafte Anfrage

Swift-Fehlername	HTTP-Status
AccessDenied	403 Verbotene
ContainerNotEmpty, ContainerAlreadyExists	409 Konflikt
Interner Fehler	500 Fehler Des Internen Servers
InvalidRange	416 Angeforderter Bereich Nicht Zu Unterprüfbar
MethodenAlled	405 Methode Nicht Zulässig
MissingContentLänge	411 Länge Erforderlich
Nicht gefunden	404 Nicht Gefunden
NotImplemsted	501 Nicht Implementiert
Vorbedingungen nicht möglich	412 Voraussetzung Fehlgeschlagen
ResourceNotFound	404 Nicht Gefunden
Nicht Autorisiert	401 Nicht Autorisiert
Nicht verarbeitbarEntity	422 Nicht Verarbeitbare Einheit

StorageGRID Swift REST-API-Operationen

Speziell für das StorageGRID System wurden Vorgänge zur Swift REST API hinzugefügt.

ABRUFEN der Container-Konsistenzanforderung

"Konsistenzwerte" Sorgen Sie für ein Gleichgewicht zwischen der Verfügbarkeit der Objekte und der Konsistenz dieser Objekte über verschiedene Storage-Nodes und Standorte hinweg. Mit der GET-Container-Konsistenzanforderung können Sie die Konsistenz bestimmen, die auf einen bestimmten Container angewendet wird.

Anfrage

HTTP-Header anfordern	Beschreibung
X-Auth-Token	Gibt das Swift-Authentifizierungs-Token für das Konto an, das für die Anforderung verwendet werden soll.
X-ntap-sg-Konsistenz	Gibt den Anforderungstyp an, wobei `true` = GET Containerkonsistenz, und `false` = get Container.

HTTP-Header anfordern	Beschreibung
Host	Der Hostname, auf den die Anforderung gerichtet ist.

Anforderungsbeispiel

```
GET /v1/28544923908243208806/Swift container
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: true
Host: test.com
```

Antwort

HTTP-Kopfzeile für Antwort	Beschreibung
Datum	Datum und Uhrzeit der Antwort.
Verbindung	Ob die Verbindung zum Server offen oder geschlossen ist.
X-Trans-ID	Die eindeutige Transaktions-ID für die Anforderung.
Inhaltslänge	Die Länge des Reaktionskörpers.
X-ntap-sg-Konsistenz	Die Konsistenz, die auf den Container angewendet wird. Folgende Werte werden unterstützt: **All**: Alle Knoten erhalten die Daten sofort, oder die Anfrage schlägt fehl. **Strong-global**: Garantiert Lese-nach-Schreiben-Konsistenz für alle Client-Anfragen über alle Standorte hinweg. **Strong-site**: Garantiert Lese-nach-Schreiben Konsistenz für alle Client-Anfragen innerhalb einer Site. **Read-after-New-write**: (Standard) bietet Read-after-write-Konsistenz für neue Objekte und eventuelle Konsistenz für Objektaktualisierungen. Hochverfügbarkeit und garantierte Datensicherung Empfohlen für die meisten Fälle. **Verfügbar**: Bietet eventuelle Konsistenz für neue Objekte und Objekt-Updates. Verwenden Sie für S3-Buckets nur nach Bedarf (z. B. für einen Bucket mit Protokollwerten, die nur selten gelesen werden, oder für HEAD- oder GET-Vorgänge für nicht vorhandene Schlüssel). Nicht unterstützt für S3 FabricPool-Buckets.

Antwortbeispiel

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
x-ntap-sg-consistency: strong-site
```

PUT Container-Konsistenzanforderung

Mit der Konsistenzanforderung für PUT-Container können Sie die Konsistenz angeben, die auf Vorgänge angewendet werden soll, die auf einen Container ausgeführt werden. Standardmäßig werden neue Container mit der Konsistenz „Read-after-New-write“ erstellt.

Anfrage

HTTP-Header anfordern	Beschreibung
X-Auth-Token	Swift Authentifizierungs-Token für das Konto zur Verwendung für die Anforderung.
X-ntap-sg-Konsistenz	Die Konsistenz, die auf Vorgänge auf dem Container angewendet werden soll. Folgende Werte werden unterstützt: **All**: Alle Knoten erhalten die Daten sofort, oder die Anfrage schlägt fehl. **Strong-global**: Garantiert Lese-nach-Schreiben-Konsistenz für alle Client-Anfragen über alle Standorte hinweg. **Strong-site**: Garantiert Lese-nach-Schreiben Konsistenz für alle Client-Anfragen innerhalb einer Site. **Read-after-New-write**: (Standard) bietet Read-after-write-Konsistenz für neue Objekte und eventuelle Konsistenz für Objektaktualisierungen. Hochverfügbarkeit und garantierte Datensicherung Empfohlen für die meisten Fälle. **Verfügbar**: Bietet eventuelle Konsistenz für neue Objekte und Objekt-Updates. Verwenden Sie für S3-Buckets nur nach Bedarf (z. B. für einen Bucket mit Protokollwerten, die nur selten gelesen werden, oder für HEAD- oder GET-Vorgänge für nicht vorhandene Schlüssel). Nicht unterstützt für S3 FabricPool-Buckets.
`Host`	Der Hostname, auf den die Anforderung gerichtet ist.

Zusammenspiel von Konsistenz- und ILM-Regeln zur Beeinträchtigung der Datensicherung

Beide Ihre Wahl "Konsistenzwert" Ihre ILM-Regel wirkt sich darüber hinaus auf den Schutz von Objekten aus. Diese Einstellungen können interagieren.

Die beim Speichern eines Objekts verwendete Konsistenz wirkt sich beispielsweise auf die anfängliche

Platzierung von Objekt-Metadaten aus, während der "Aufnahmeverhalten" Die für die ILM-Regel ausgewählt wurde, wirkt sich auf die anfängliche Platzierung von Objektkopien aus. StorageGRID benötigt zur Erfüllung von Clientanfragen Zugriff auf die Metadaten und die Daten eines Objekts. Durch die Auswahl einer passenden Sicherungsstufe für die Konsistenz und das Aufnahmeverhalten können die Daten am Anfang besser gesichert und Systemantworten besser vorhersehbar sein.

Beispiel für die Interaktion von Konsistenz- und ILM-Regeln

Angenommen, Sie haben ein Grid mit zwei Standorten mit der folgenden ILM-Regel und folgender Konsistenz:

- **ILM-Regel**: Erstellen Sie zwei Objektkopien, eine am lokalen Standort und eine an einem entfernten Standort. Das strikte Aufnahmeverhalten wird ausgewählt.
- **: "Strong-global" (Objektmetadaten werden sofort an alle Standorte verteilt.)

Wenn ein Client ein Objekt im Grid speichert, erstellt StorageGRID sowohl Objektkopien als auch verteilt Metadaten an beiden Standorten, bevor der Kunde zum Erfolg zurückkehrt.

Das Objekt ist zum Zeitpunkt der Aufnahme der Nachricht vollständig gegen Verlust geschützt. Wenn beispielsweise der lokale Standort kurz nach der Aufnahme verloren geht, befinden sich Kopien der Objektdaten und der Objektmetadaten am Remote-Standort weiterhin. Das Objekt kann vollständig abgerufen werden.

Wenn Sie stattdessen dieselbe ILM-Regel und die Konsistenz von „starken Standorten“ verwenden, erhält der Client möglicherweise eine Erfolgsmeldung, nachdem die Objektdaten am Remote-Standort repliziert wurden, aber bevor die Objektmetadaten dort verteilt werden. In diesem Fall entspricht die Sicherung von Objektmetadaten nicht dem Schutzniveau für Objektdaten. Falls der lokale Standort kurz nach der Aufnahme verloren geht, gehen Objektmetadaten verloren. Das Objekt kann nicht abgerufen werden.

Die Beziehung zwischen Konsistenz- und ILM-Regeln kann komplex sein. Wenden Sie sich an NetApp, wenn Sie Hilfe benötigen.

Anforderungsbeispiel

```
PUT /v1/28544923908243208806/_Swift container_
X-Auth-Token: SGRD_3a877009a2d24cb1801587bfa9050f29
x-ntap-sg-consistency: strong-site
Host: test.com
```

Antwort

HTTP-Kopfzeile für Antwort	Beschreibung
`Date`	Datum und Uhrzeit der Antwort.
`Connection`	Ob die Verbindung zum Server offen oder geschlossen ist.
`X-Trans-Id`	Die eindeutige Transaktions-ID für die Anforderung.
`Content-Length`	Die Länge des Reaktionskörpers.

Antwortbeispiel

```
HTTP/1.1 204 No Content
Date: Sat, 29 Nov 2015 01:02:18 GMT
Connection: CLOSE
X-Trans-Id: 1936575373
Content-Length: 0
```

In den Audit-Protokollen werden Swift-Vorgänge nachverfolgt

Alle erfolgreichen Vorgänge zum LÖSCHEN, ABRUFEN, NACHFÜHREN, POSTEN und PUT werden im StorageGRID Audit-Protokoll verfolgt. Fehler und Info-, auth- oder OPTIONS-Anforderungen werden nicht protokolliert.

Konto-Operationen

- "GET Konto"
- "HAUPTKONTO"

Container-Operationen

- "Container LÖSCHEN"
- "GET Container"
- "KOPF Behälter"
- "Legen Sie den Behälter"

Objekt-Operationen

- "Delete Objekt"
- "GET Objekt"
- "HEAD-Objekt"
- "PUT Objekt"

Verwandte Informationen

- "Zugriff auf die Audit-Log-Datei"
- "Audit-Meldungen des Clients schreiben"
- "Client liest Audit-Meldungen"