StorageGRID® NAS Bridge 2.3

Management API Guide

November 2019 | 215-14200_2019-11_en-us
doccomments@netapp.com

NetApp®

Contents

Understanding the NAS Bridge management API

The NAS Bridge management API provides access to the NAS Bridge management functionality. The API is based on RESTful web services. Before using the management API, you should understand its design, architectural components, and limitations.

RESTful web services foundation

The NAS Bridge management API is based on RESTful web services. Representational State Transfer (REST) establishes guidelines for exposing server-based resources. It provides a flexible and extensible foundation for managing the NAS Bridge.

Resources and state representation

The core aspects of RESTful web services include the following:

- Identification of system or server-based resources
  Every system uses and maintains resources. A resource can be a file, business information, a process, or an administrative entity.
- Definition of resource states and associated state operations
  Resources are always in one of a finite number of states. The operations used to change states must be clearly defined.

Messages are exchanged between the client and server to access and change the state of resources according to the CRUD (Create, Read, Update, and Delete) operations.

HTTP messages

Hypertext Transfer Protocol (HTTP) is a protocol used by web services to exchange messages about resources. During HTTP message exchanges, the HTTP verbs are mapped to the resources and their corresponding state management actions.

The NAS Bridge management API relies on a subset of HTTP and uses the following HTTP verbs:

- GET
- PUT
- POST
- PATCH
- DELETE

HTTP is stateless. Therefore, to associate a set of related requests and responses under one identity, additional information must be added to the data flows, including HTTP headers or cookies. Also note that HTTP uses TCP port 80 by default.

URI endpoints

Uniform Resource Identifiers (URIs) are used to specify the endpoints where resources are located. URIs provide the general framework for creating unique resource names. The resources are exposed in a structure that is similar to a hierarchical directory.

A Uniform Resource Locator (URL) is a type of URI adapted primarily for the web and used in RESTful web services. A URL is used to identify a resource and to access a representation of the resource.

JSON formatting

While there are several possible ways that information can be transferred between a web client and server, the most popular option is JavaScript Object Notation (JSON). JSON is a standard for representing simple data structures, including objects and arrays, in plain text. JSON is used by the NAS Bridge RESTful web services to represent and transfer state information describing each resource.

Multiple access paths

You can access the NAS Bridge management API in a few different ways:

- **NAS Bridge native user interface**: An indirect way to access the API is through the NAS Bridge native web user interface. When you use a browser to access the NAS Bridge through its management IP address, the initial page is displayed with administrative functions organized by category. The browser accesses the management API and reformats the data according to the user interface design. In other words, you interact with the NAS Bridge user interface, and the user interface makes the corresponding API calls.
- **API Docs (Swagger) web page**: After you have used a browser to access the NAS Bridge through its management IP address, you can access the API Docs web page, powered by the Swagger open source platform. The API Docs page allows you to interact with the API through a user interface that illustrates how the API responds to parameters and options. The instructions for using the NAS Bridge management API contain examples that show the API Docs (Swagger) interface.

  Attention: Any API operations you perform using the API Docs webpage are live operations. Be careful not to create, update, or delete configuration data or other data by mistake.

- **Custom program** – You can access the management API using one of several different programming languages and tools, such as Python, Java, and cURL. A program, script, or tool that uses the API acts as a RESTful web services client. Using a programming language enables you to better understand the API as well as to automate the management and control of a NAS Bridge.

Uploading data

If you want to upload a file using the API, such as a NAS Bridge recovery package, you must perform a POST with the multi-part content type. Otherwise, the file will not upload.

General workflow for using the API

Use the following recommended workflow when accessing the NAS Bridge management API.

When you start an API session, an authentication token is generated. When the token is generated, it is automatically inserted into each API call during that session. Authentication is required whenever you issue an API call.

1. Create an API session by providing a user name and password. Once you log in to the management API, an authentication token is automatically generated.
2. Perform one or more additional API calls as needed to complete the desired task, supplying the necessary information for each call. The authentication token is automatically inserted into each API call.

3. Delete the session. Once the session is deleted, the authentication token is reset, and a different token will be generated when you start a new session.

   Attention: For security reasons, it is important to delete the session after completing your API calls.

Related tasks

Obtaining an authentication token on page 11

Understanding NAS Bridge resources and objects

The NAS Bridge management API allows multiple instances of each resource type to exist concurrently. Each instance can be viewed as an object. Therefore, for a given resource type, you can consider the resource instances to be in an array of one or more objects. This design gives you better flexibility and control when accessing the resource instances through the API.

Object identifiers

Each resource instance or object is assigned a unique identifier (ID). These object IDs are integer values. The IDs are unique within a specific resource type, but not within the system as a whole. For example, if you create a DNS server, it might be assigned ID=1. Subsequently, you might create an NTP server that might also be assigned ID=1. In this case, it is acceptable to have identical IDs because the resource types are different.

The IDs are generally returned in the HTTP response after a successful add request. An ID must be provided in the following situations:

- When getting the current status of a resource instance
- When linking resource instances where one object refers to another
- When deleting a resource instance

Summary of resource statuses

Each resource instance has an associated status. In general, a resource's status can be accessed and displayed through the management API.

The following status values are used for NAS Bridge resources:

- NOTIFYING
  The underlying services are being notified of a change.
- COMMITTING
  The underlying services are coordinating the commitment of a change.
- ABORTING
  A change has been rejected. Check the alarms page and error log messages for more information.
- FAILED
  A change failed and all services have completed a rollback. Check the alarms page for more information.
- READY
  A change has been successfully completed, and the corresponding resource is ready for use.

In most cases, the original request or action type is added to the beginning of the status to create the complete state description. For example, after issuing an add request, the new

resource's status might temporarily be ADDING / NOTIFYING. A similar construction applies when removing resources.

How asynchronous operation works

Many of the API calls, particularly those that create or remove a resource, can take a longer time to complete than most other API calls. NAS Bridge processes these types of requests asynchronously. When issuing a call that operates asynchronously, you must check the status of the resource instance to confirm that the request is complete.

With asynchronous processing, the initial successful HTTP response indicates that the request has been accepted but not necessarily finished. Therefore, after making an asynchronous request to add or remove a resource, you must test the resource instance for completion of the request.

Note: Refer to the API Docs (Swagger) web page for documentation to help determine whether a specific API call operates asynchronously. The implementation notes section on the page (if present) contains the details.

Checking a resource status after an add request

After issuing an API call that adds a resource, you should poll the status of the resource to verify completion of the request. A request is complete when the new resource's status is READY.

After creating the required authentication token, you should use the following high level process when asynchronously adding a resource:

1. Issue the API call to add a resource.
2. Receive an HTTP response indicating successful acceptance of the request.
3. Extract the resource ID from the HTTP response.
4. Within a timed loop, perform the following steps in each loop cycle:
   a. Get the current status of the resource based on the ID.
   b. If the resource's status is not READY, perform the loop again.
   c. If the resource's status is FAILED UPDATE or FAILED ADD, abort the operation, fix the problem (for example, remove the failed resource), and perform the loop again.
5. When the resource's status is READY, you can stop.
6. If the polling loop times out (according to your arbitrary timeout value) before the resource's status is READY, report an error.

Checking for resource removal after a delete request

After issuing an API call that deletes a resource, you should poll the resource to verify that it has been removed. A request is complete when the resource no longer exists.

After creating the required authentication token, you should use the following high-level process when asynchronously removing a resource:

1. Issue the API call to delete a resource.
2. Receive an HTTP response indicating successful acceptance of the request.
3. Within a timed loop, perform the following in each cycle:
   a. Get the current status of the resource based on the ID.

   b. If resource is located (HTTP code 200), perform the loop again.

4. When the GET request responds with Not Found (HTTP code 404), you can stop.
5. If the polling loop times out (according to your arbitrary timeout value) and the resource still exists, report an error.

Summary of resource types supported by the API

As part of using the NAS Bridge management API, you should be aware of the RESTful resource types that are supported. The API calls are organized under the various resource types.

Refer to the API Docs (Swagger) web page for a complete list of the API calls, as well as the details of each call. Also refer to the release notes publication for information regarding updates or changes to the management API.

The management API calls are organized according to the following resource types:

- Active Directory controllers
- Alert configuration
- AutoSupport (ASUP) service
- Cache devices
- Configuration exports (recovery packages)
- Debug
- Decommissions
- Disks
- DNS servers
- File systems
- Metrics
- Network interfaces
- Network logical interfaces (Network LIFs)
- Network routes
- NTP servers
- Object stores
- Passwords
- Proxy server
- Reboot
- Sessions
- SMTP servers (email servers)
- Storage API certificate
- System events
- System information

- Upgraded disk activation
- Upgrades
- Users

Accessing and using the API

The following instructions illustrate how to access the NAS Bridge management API using the API Docs (Swagger) web page. Alternatively, you can use a programming language or other command-line tool.

Accessing the API Docs web page

You access the API Docs (Swagger) web page from the NAS Bridge user interface.

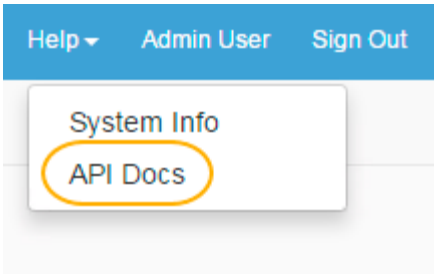
Before you begin

You must have the management IP address or domain name of the NAS Bridge.

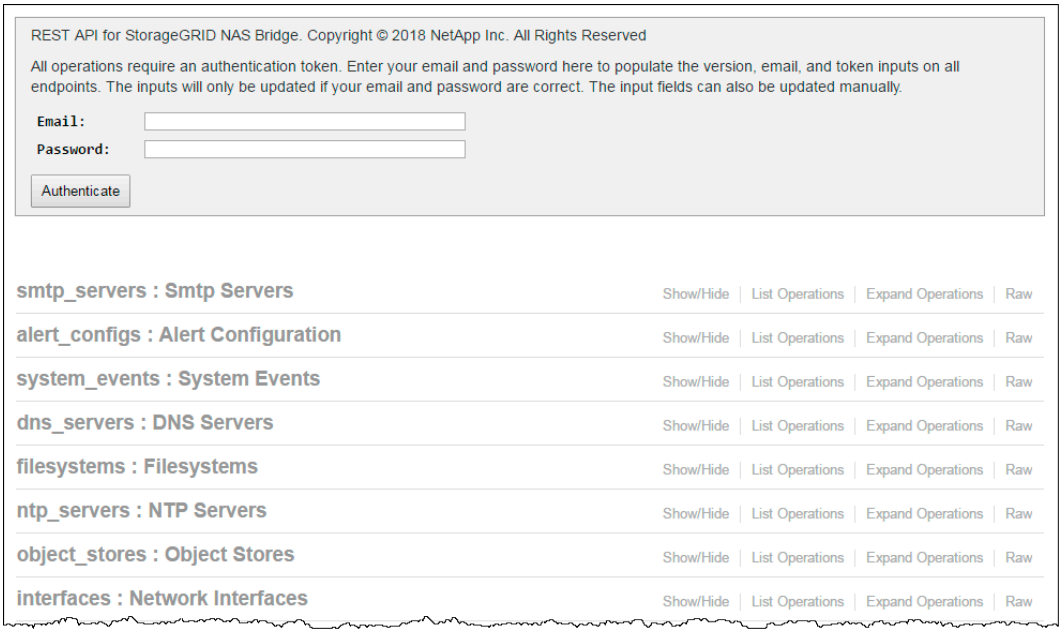
Steps

1. Log in to the NAS Bridge.
2. Click **Help** in the upper right corner of the web page.

   The Help menu is displayed.
3. Click **API Docs**.



The API Docs page is displayed in a new tab. The top of the page contains the login dialog for obtaining an authentication token, which is required before you can use the management API. The API calls are listed below the login dialog and organized by resource category.



Related tasks

Obtaining an authentication token on page 11

Related references

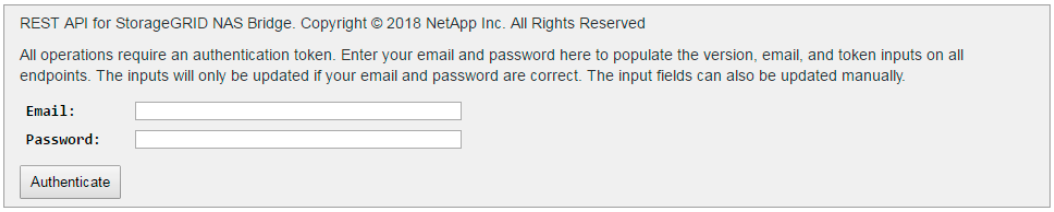
Summary of resource types supported by the API on page 8

Obtaining an authentication token

To use the management API, you must first obtain an authentication token.

Steps

1. At the top of the API Docs page, enter your user email and password.

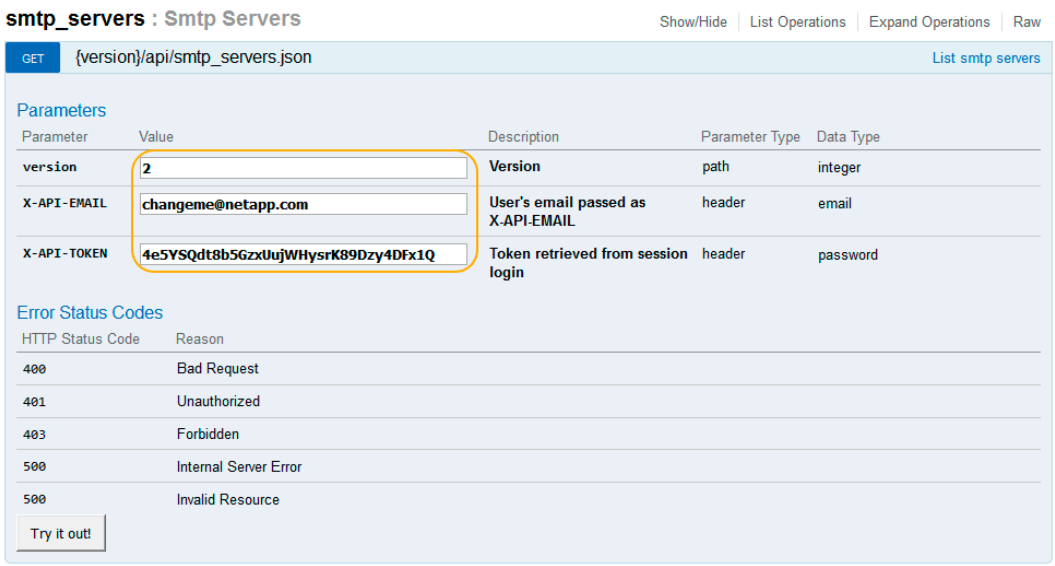


2. Click **Authenticate**.

   If your login information is correct, an authentication token is generated, and the following occurs for every API endpoint that you invoke during this session:

   - The authentication token is automatically inserted into the `X-API-TOKEN` field.
   - The email address you used to log in is automatically inserted into the `X-API-EMAIL` field.
   - The API version is inserted into the `version` field.

Example



Related tasks

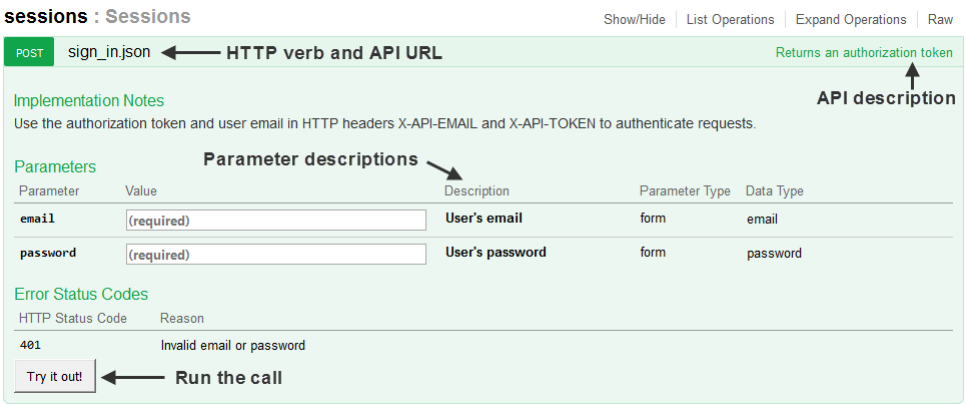
Accessing the API Docs web page on page 10

Understanding the details of an API call

The details of all the API calls are included on the API Docs (Swagger) web page. All of the API calls are documented and displayed using a common format. By understanding a single API call, you can interpret the details of all the API calls.

Steps

1. On the main API Docs page, click **sessions**.
2. Click **POST** to display the details of the API call used to request an authentication token.



3. Examine the entire page to understand the API calls and what you must enter.

   You should note the HTTP verb and URL, required input parameters, the HTTP status codes used, and any implementation notes.

Performing a simple task using the API

To better understand the API, you should use the API Docs (Swagger) web page to perform a simple task, such as creating a system event.

Before you begin

You must know how to access the API Docs web page using a browser. In addition, you must have the credentials for an administrator account, including the user name and password.

Attention: Any API operations you perform using the API Docs webpage are live operations. Be careful not to create, update, or delete configuration data or other data by mistake.

Steps

1. Creating a system event message to test the API on page 13
2. Resetting the authentication token on page 17

Related tasks

Accessing the API Docs web page on page 10

Creating a system event message to test the API

A simple way to test the management API is to create a message in the system event (alarms) log.

Before you begin

You must have logged in to the API Docs (Swagger) page so that the authentication token, email address, and API version will be automatically inserted into the corresponding fields for each API endpoint.

About this task

You use the API Docs page to create a system event. After the event has been created, you can display the message in two ways:

- Using the GET function on the API Docs (Swagger) page
- Using the NAS Bridge user interface

Steps

1. On the API Docs page, click **system_events**.
2. Click **POST** to display the API call used to create a new system event.

   The POST API call is displayed. If you logged in successfully at the top of the web page, the version, email address (user account), and authentication token are automatically populated on the page.
3. Type a test message, severity (from the list of allowed values), and test facility:

POST {version}/api/system_events.json — Create a new system event

Parameters

Parameter	Value	Description	Parameter Type	Data Type
version	2	**Version**	path	integer
X-API-EMAIL	changeme@netapp.com	**User's email passed as X-API-EMAIL**	header	email
X-API-TOKEN	xZuQBWRe2EE_N3jy6UYSwRBH_VRN1XsVTA	**Token retrieved from session login**	header	password
system_event[message]	This is a test message	**Message describing the event**	form	string
system_event[severity]	debug	**One of the following: debug info notice warning error critical alert emergency**	form	string
system_event[facility]	mymodule	**Facility, component or module reporting the event**	form	string

4. Click **Try it out**.

Request URL

```
https://10.96.104.166:443/2/api/system_events.json
```

Response Body


```
{
  "response": {
    "id": 40,
    "severity": "debug",
    "facility": "mymodule",
    "message": "This is a test message",
    "created_at": "2017-11-10T22:39:23.110Z",
    "updated_at": "2017-11-10T22:39:23.110Z",
    "config": null,
    "config_id": null
  }
}
```

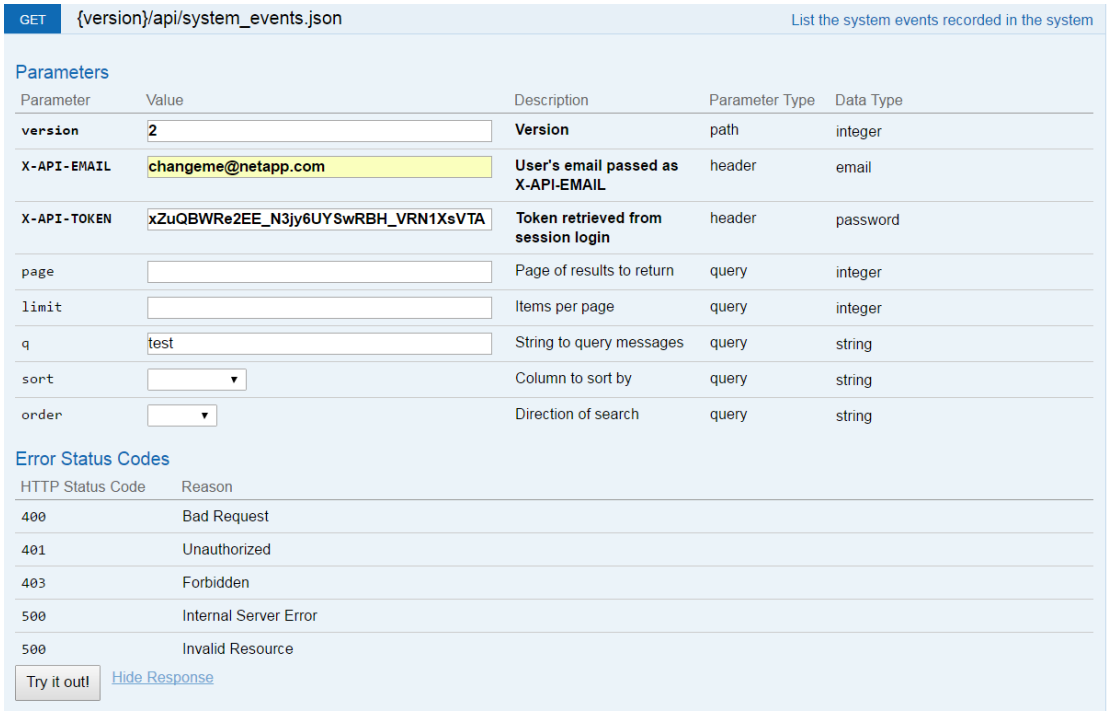

Response Code

```
200
```

The response code 200 indicates success.

5. To verify the new message was created:

   - On the API Docs page, click **system_events > GET**, then make sure that the email address and authentication token fields are set. Use the q parameter to limit the results to those messages that include all or part of your message text. Then, click **Try it out** to list the message you created.



GET {version}/api/system_events.json — List the system events recorded in the system

Parameters

Parameter	Value	Description	Parameter Type	Data Type
version	2	**Version**	path	integer
X-API-EMAIL	changeme@netapp.com	**User's email passed as X-API-EMAIL**	header	email
X-API-TOKEN	xZuQBWRe2EE_N3jy6UYSwRBH_VRN1XsVTA	**Token retrieved from session login**	header	password
page		Page of results to return	query	integer
limit		Items per page	query	integer
q	test	String to query messages	query	string
sort		Column to sort by	query	string
order		Direction of search	query	string

Error Status Codes

HTTP Status Code	Reason
400	Bad Request
401	Unauthorized
403	Forbidden
500	Internal Server Error
500	Invalid Resource

Try it out! Hide Response

Request URL

```
https://10.96.104.166:443/2/api/system_events.json?q=test
```

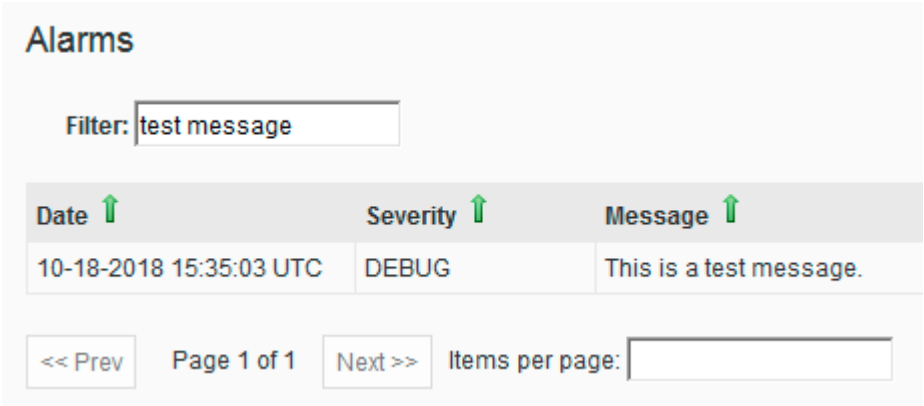
Response Body


```
    {
      "id": 40,
      "severity": "debug",
      "facility": "mymodule",
      "message": "This is a test message",
      "created_at": "2017-11-10T22:39:23.000Z",
      "updated_at": "2017-11-10T22:39:23.000Z",
      "config": null,
      "config_id": null
    }
  ],
  "count": 8,
  "pagination": {
    "current": 1,
    "previous": null,
    "next": null,
    "per_page": 25,
    "pages": 1,
    "count": 8
  }
```


Response Code

```
200
```

- From the NAS Bridge user interface, click **Alarms**, and filter or sort the events to find the message you posted.



Related tasks

Obtaining an authentication token on page 11

Resetting the authentication token

You should reset the authentication token after completing your API calls. This practice improves the security of the system by preventing the token from being reused.

Before you begin

You must have the following parameters:

- A valid authentication token. Once you obtain the token, it is automatically inserted into the required fields on API endpoints.
- The administrator email address (user account) that was used to create the authentication token.

Steps

1. On the API Docs page, click **sessions**.
2. Click **DELETE** to display the API call used to reset an authentication token.

   The DELETE API call is displayed. If you logged in successfully at the top of the web page, the version, email address (user account), and authentication token are automatically populated on the page.
3. Click **Try it out!**

   The response code 204 indicates the token was deleted.

Copyright

Trademark information

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email.

doccomments@netapp.com

To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 1395 Crossman Ave., Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277