



## Lyve Cloud Account API version 2 Guide

---



**Hier klicken, um eine aktuelle Online-Version** dieses Dokuments aufzurufen. Auch finden Sie hier die aktuellsten Inhalte sowie erweiterbare Illustrationen, eine übersichtlichere Navigation sowie Suchfunktionen.

# Contents

<b>1</b>	<b>Overview</b>	<b>7</b>
<b>2</b>	<b>Protocol Basics</b>	<b>8</b>
	Data structure	8
	• API endpoint	8
	CRUD operations	8
	Response to request	8
	• Successful request	8
	• Failed request	9
<b>3</b>	<b>API Error Codes</b>	<b>10</b>
<b>4</b>	<b>Authentication</b>	<b>12</b>
	Request	12
	• Body parameter	12
	• Parameters	12
	• Code samples	13
	• Go	13
	• Java	14
	• JavaScript	14
	• Python	14
	• Ruby	15
	Responses	15
	Testing a session token for validity	17
	Request	18
	• Code samples	18
	• Go	18
	• Java	18
	• JavaScript	19
	• Python	19
	• Ruby	19
	Responses	20
<b>5</b>	<b>Creating Permissions</b>	<b>22</b>
	Creating new permissions	22
	• Using the Policy permission file	23
	Request	24
	• Body parameter	24
	• Parameters	24

- Code samples ..... 26
  - Go ..... 26
  - Java ..... 27
  - JavaScript ..... 28
  - Python ..... 28
  - Ruby ..... 28
- Responses ..... 29

## 6 Listing Permissions ..... 33

- Request ..... 33
  - Parameters ..... 33
  - Code samples ..... 33
    - Go ..... 33
    - Java ..... 34
    - JavaScript ..... 34
    - Python ..... 35
    - Ruby ..... 35
- Responses ..... 35

## 7 Getting Permission by ID ..... 40

- Request ..... 40
  - Parameters ..... 40
  - Code samples ..... 40
    - Go ..... 40
    - Java ..... 41
    - JavaScript ..... 41
    - Python ..... 42
    - Ruby ..... 42
- Responses ..... 42

## 8 Deleting Permission by ID ..... 48

- Request ..... 48
  - Parameters ..... 48
  - Code samples ..... 48
    - Go ..... 48
    - Java ..... 49
    - JavaScript ..... 49
    - Python ..... 50
    - Ruby ..... 50
- Responses ..... 50

## 9 Updating a Permission ..... 53

- Request ..... 53
  - Body parameter ..... 53
  - Parameters ..... 53

- Code samples ..... 55
  - Go ..... 55
  - Java ..... 56
  - JavaScript ..... 57
  - Python ..... 57
  - Ruby ..... 58
- Responses ..... 58

## 10 .Creating a Service Account ..... 62

- Request ..... 62
  - Body parameter ..... 62
  - Parameters ..... 62
  - Code samples ..... 63
    - Go ..... 63
    - Java ..... 64
    - JavaScript ..... 64
    - Python ..... 65
    - Ruby ..... 65
- Responses ..... 65

## 11 .Listing Service Accounts ..... 70

- Request ..... 70
  - Parameters ..... 70
  - Code samples ..... 70
    - Go ..... 70
    - Java ..... 71
    - JavaScript ..... 72
    - Python ..... 72
    - Ruby ..... 72
- Responses ..... 72

## 12 .Getting Service Account Data by ID ..... 76

- Request ..... 76
  - Parameters ..... 76
  - Code samples ..... 76
    - Go ..... 76
    - Java ..... 77
    - JavaScript ..... 77
    - Python ..... 77
    - Ruby ..... 77
- Responses ..... 78

## 13 .Updating a Service Account ..... 82

- Request ..... 82
  - Body parameter ..... 82

• Parameters .....	82
• Code samples .....	83
• Go .....	83
• Java .....	83
• JavaScript .....	84
• Python .....	84
• Ruby .....	84
Responses .....	85
<b>14 .Enabling a Service Account .....</b>	<b>88</b>
Request .....	88
• Parameters .....	88
• Code samples .....	88
• Go .....	88
• Java .....	89
• JavaScript .....	89
• Python .....	89
• Ruby .....	90
Responses .....	90
<b>15 .Disabling a Service Account .....</b>	<b>94</b>
Request .....	94
• Parameters .....	94
• Code samples .....	94
• Go .....	94
• Java .....	95
• JavaScript .....	95
• Python .....	95
• Ruby .....	96
Responses .....	96
<b>16 .Deleting a Service Account by ID .....</b>	<b>99</b>
Request .....	99
• Parameters .....	99
• Code samples .....	99
• Go .....	99
• Java .....	100
• JavaScript .....	100
• Python .....	100
• Ruby .....	100
Responses .....	101
<b>17 .Usage .....</b>	<b>104</b>
Getting historical storage usage by month .....	104
Request .....	104

• Parameters .....	104
• Code samples .....	105
• Go .....	105
• Java .....	105
• JavaScript .....	106
• Python .....	106
• Ruby .....	106
Responses .....	107
Getting current month storage usage .....	109
Request .....	109
• Code samples .....	110
• Go .....	110
• Java .....	110
• JavaScript .....	111
• Python .....	111
• Ruby .....	111
Responses .....	112

# Overview

The Lyve Cloud Account API provides programmatic access to Lyve Cloud to perform administrative tasks. You can use the APIs to manage permissions, service accounts, and usage. These APIs are accessed after successful authentication by calling an endpoint that uses secret credentials. Every active Lyve Cloud account administrator can generate one set of account API credentials.

The Account API expects each admin user to generate unique credentials using the Lyve Cloud console to authenticate. For more information on generating the API credentials, see [Using Account API](#).

# Protocol Basics

## Data structure

The API uses JSON structures, which may include the following data types:

- **Boolean** – True or false
- **Number** – Numeric value, For example, 1234
- **String** – Characters enclosed in quotes, For example, "Lyve Cloud"

## API endpoint

An endpoint is the URL of the entry point for a Lyve Cloud web service.

```
https://api.lyvecloud.seagate.com/v2
```

## CRUD operations

The following methods can be used to Create, Read, Update and Delete Lyve Cloud entities:

Operation	Syntax	Body Structure	Response Structure
Create an entity	POST /uri	Required	The created entity
Read an entity	GET /uri/<id>	-	The requested entity
Read all entities	GET /uri	-	List of entities
Update an entity	PUT /uri/<id>	Required	The updated entity
Delete an entity	DELETE /uri/<id>	-	The deleted entity

## Response to request

### Successful request

The successful status message is 200 (OK); for some API request calls, a successful response contains the JSON format. The details of the JSON format are explained in the respective API request.



## Failed request

When an error occurs, the header information contains the following:

- Content-Type: application/JSON
- An appropriate 4xx or 5xx HTTP status code

The following sample error response shows the structure of response elements common to all REST error responses.

```
{
  "code": "string",
  "message": "string"
}
```

The following table explains the REST error response elements.

Name	Description
Code	A string that identifies the error. It is meant to be read and understood by programs that detect and handle errors by type.
Message	A description of the error condition. Programs may display the message directly to the end user if they meet an error condition. An error message in the body aids in resolving the issue.

# API Error Codes

During an error, you will receive a JSON object in the response body containing the information about the error. Error responses also include an error message in the body to assist in resolving the problem.

**Table 1. API error codes**

Code	HTTP Status Code	Message
<code>ExpiredToken</code>	Bad Request(400)	Token expired.
<code>InvalidToken</code>	Bad Request(400)	Token is not valid.
<code>InvalidTimeRange</code>	Bad Request (400)	Invalid time range.
<code>InvalidArgument</code>	Bad Request (400)	This error might occur for the following reasons: <ul style="list-style-type: none"><li>• One or more of the specified arguments was not valid.</li><li>• The request was missing an argument.</li></ul>
<code>InvalidBucketName</code>	Bad Request (400)	A bucket that keeps audit logs must be set with Object Immutability.
<code>AuthenticationFailed</code>	Forbidden (403)	Authentication failed.
<code>NoServiceAvailable</code>	Forbidden (403)	The account has no services enabled for it.
<code>PermissionNotFound</code>	Not Found (404)	A permission was not found.
<code>ServiceAccountNotFound</code>	Not Found (404)	A service account was not found.

Code	HTTP Status Code	Message
<code>ServiceAccountExpired</code>	Not Found (404)	The service account has expired. Couldn't enable the expired service.
<code>ServiceAccountExpired</code>	Not Found (404)	The service account has expired. Couldn't disable the expired service.
<code>BucketNotFound</code>	Not Found (404)	A bucket was not found.
<code>PermissionNotReady</code>	Conflict (409)	The permission is still being processed by some regions.
<code>ServiceAccountNotReady</code>	Conflict (409)	The service account is still being processed by some regions.
<code>PermissionNameAlreadyExists</code>	Conflict (409)	The permission name is already in use. Please use a different name.
<code>ServiceAccountNameAlreadyExists</code>	Conflict (409)	The service account name is already in use. Please use a different name.
<code>ServiceNotReady</code>	Service Unavailable (503)	The server is not ready to handle the request. Please retry the request later.
<code>InternalError</code>	Internal Server Error (500)	The server encountered an internal error. Please retry the request.

# Authentication

Lyve Cloud uses a token to authenticate access to the Lyve Cloud Account API. The Token is obtained by calling an endpoint that uses the Account ID, Access Key, and Secret Key. It returns a valid time-bound token, where the default expiration duration is 24 hours. When a token expires, the account API returns `HTTP 400` code, after which the client application needs to obtain a new token.

Each request to an API endpoint must include the bearer header value. This value requires specifying a token in the authorization header of a request.

A token is a unique string. Tokens eliminate the need for passing user credentials with requests. Such a token is issued by the `/auth/token` endpoint.

## Request

A POST to `/auth/token` is used to exchange user credentials for an access token.

**POST /auth/token**

## Body parameter

```
{
  "accountId": "string",
  "accessKey": "string",
  "secret": "string"
}
```

## Parameters

Name	In	Type	Required	Description
accountId	body	string	True	The Account ID is a unique identifier of the Lyve Cloud Account. The Account ID is created during the customer onboarding and is unique across all Lyve Cloud accounts.

Name	In	Type	Required	Description
accessKey	body	string	True	The access key is generated when you generate Account API credentials. For more information, see <a href="#">Using Account API</a> .
secret	body	string	True	The secret key is generated when you generate Account API credentials. For more information, see <a href="#">Using Account API</a> .

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Content-Type": []string{
            "application/json",
        },
        "Accept": []string{
            "application/json",
        },
    }

    jsonReq := `{"key":"value"}` // replace with your JSON request
    data := bytes.NewBuffer([]byte(jsonReq))
    req, err := http.NewRequest("POST", "https://api.lyvecloud.seagate.com/v2/auth/token/", data)
    if err != nil {
        // handle error
    }
    req.Header = headers

    client := &http.Client{}
    resp, err := client.Do(req)
    if err != nil {
        // handle error
    }
    // handle response _ = resp }
```

## Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/auth/token/");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("POST");
            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } } }
```

## JavaScript

```
const inputBody = `{
  "accountId": "string",
  "accessKey": "string",
  "secret": "string"
}`;

const headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json'
};

fetch("https://api.lyvecloud.seagate.com/v2/auth/token/", {
  method: 'POST',
  body: inputBody,
  headers: headers
})
.then(function(res) {
  return res.json();
})
.then(function(body) { console.log(body); });
```

## Python

```

import requests
headers = {
    'Content-Type': 'application/json',
    'Accept': 'application/json'
}

data = {
    "accountId": "string",
    "accessKey": "string",
    "secret": "string"
}

r = requests.post('https://api.lyvecloud.seagate.com/v2/auth/token/', headers=headers, json=data) print(r.json())

```

## Ruby

```

require 'rest-client'
require 'json'

headers = {
    'Content-Type' => 'application/json',
    'Accept' => 'application/json'
}

data = {
    "accountId" => "string",
    "accessKey" => "string",
    "secret" => "string"
}

result = RestClient.post 'https://api.lyvecloud.seagate.com/v2/auth/token/', data.to_json, headers

p JSON.parse(result)

```

## Responses

Status Code	Description	Return JSON payload
-------------	-------------	---------------------

Status Code	Description	Return JSON payload						
200	<p>OK</p> <p>The session token is returned successfully to use in subsequent API calls. The expiration time of the token is returned in response, after the expiration duration the token is expired.</p>	<pre data-bbox="854 222 1448 407"> {   "token": "string",   "expirationSec": 864000, } </pre> <table border="1" data-bbox="854 432 1448 900"> <thead> <tr> <th data-bbox="854 432 1102 512">Field</th> <th data-bbox="1102 432 1448 512">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="854 512 1102 705">token</td> <td data-bbox="1102 512 1448 705">The JWT session token to use in subsequent API calls.</td> </tr> <tr> <td data-bbox="854 705 1102 900">expirationSec</td> <td data-bbox="1102 705 1448 900">The number of seconds remaining till the token expires.</td> </tr> </tbody> </table>	Field	Description	token	The JWT session token to use in subsequent API calls.	expirationSec	The number of seconds remaining till the token expires.
Field	Description							
token	The JWT session token to use in subsequent API calls.							
expirationSec	The number of seconds remaining till the token expires.							
400	<p>Bad request</p> <p>The request is invalid and has invalid information.</p>	<pre data-bbox="854 1010 1448 1194"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="854 1220 1448 1873"> <thead> <tr> <th data-bbox="854 1220 1130 1299">code</th> <th data-bbox="1130 1220 1448 1299">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="854 1299 1130 1873">InvalidArgument</td> <td data-bbox="1130 1299 1448 1873"> <p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul> </td> </tr> </tbody> </table>	code	message	InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul>		
code	message							
InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul>							



Status Code	Description	Return JSON payload				
403	Authentication failed.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>Authentication failed</td> <td>Authentication failed.</td> </tr> </tbody> </table>	code	message	Authentication failed	Authentication failed.
code	message					
Authentication failed	Authentication failed.					
500	Internal Server Error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalServerError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceNotReady</td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

## Testing a session token for validity

To access the API, you must request an access token when authenticating a user. The **GET /auth/token** is used to validate a session and return the remaining duration for that session.

## Request

The GET API allows validating a session. It returns the remaining duration of the token.

```
GET /auth/token
```

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }

    data := bytes.NewBuffer([]byte{
        jsonReq,
    })
    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/auth/token/", data)
    req.Header = headers
    client := &http.Client{}
    resp, err := client.Do(req)
    // ... }
```

### Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;
```

```

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/auth/token/");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("GET");
            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```

const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};
fetch('https://api.lyvecloud.seagate.com/v2/auth/token/', {
  method: 'GET',
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });

```

## Python

```

import requests

headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}

r = requests.get('https://api.lyvecloud.seagate.com/v2/auth/token/', headers=headers) print(r.json())

```

## Ruby

```

require 'rest-client'
require 'json'

headers = {

```

```

'Accept' => 'application/json',
'Authorization' => 'Bearer {access-token}'
}

result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/auth/token/', headers: headers

p JSON.parse(result)

```

## Responses

Status Code	Description	Return JSON payload						
200	<p>OK</p> <p>A session token is valid and is in effect.</p>	<pre>{   "expirationSec": 0 }</pre> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>expirationSec</code></td> <td>Number of seconds remaining until the token expires.</td> </tr> </tbody> </table>	Field	Description	<code>expirationSec</code>	Number of seconds remaining until the token expires.		
Field	Description							
<code>expirationSec</code>	Number of seconds remaining until the token expires.							
400	<p>Bad request</p> <p>Either the token is expired or not valid.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ExpiredToken</code></td> <td>Token expired.</td> </tr> <tr> <td><code>InvalidToken</code></td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	message	<code>ExpiredToken</code>	Token expired.	<code>InvalidToken</code>	Token is not valid.
code	message							
<code>ExpiredToken</code>	Token expired.							
<code>InvalidToken</code>	Token is not valid.							

Status Code	Description	Return JSON payload				
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>InternalServerError</code></td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.
code	message					
<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceNotReady</code></td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.
code	message					
<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.					

# Creating Permissions

Permissions control access to buckets and define which actions the service accounts can perform on a bucket.

Bucket permission and Policy permission are two options available for granting permission to your buckets.

- **Bucket permission:** Bucket permission is used to set **Read only**, **Write only**, or **All operations** permission for selected buckets. Using Bucket permissions, you can grant access permissions to your bucket and the objects in the bucket. Only the admin and storage admin can associate permissions for the buckets. The permissions attached to the bucket apply to all of the objects in the bucket. For more information, see [Managing bucket access permissions](#)
- **Policy permission:** Policy permissions are used to manage bucket-level permission by uploading a JSON file. You can also import a file compatible with the AWS IAM policy file. Using the Policy permission, you can allow or deny requests at a granular level based on the elements in the policy, resources, and aspects or conditions of the request. For more information, see [Creating policy permissions](#).

## Creating new permissions

You can create bucket permissions without any buckets in the account only if you apply permission to all buckets in the account or all buckets with a prefix.

Apply permission types using the **type** parameter.

- **One or more existing buckets:** Choose one or more from the bucket list.
- **All buckets in this account with a prefix:** The bucket names must use the same few initial characters. For example, if four unique buckets for customer01 are created, such as customer01rawdata, customer01zipdata, customer01media and customer01, enter a prefix of the bucket names to assign and apply the permission. In this case, use the same beginning characters for each bucket for our prefix, customer01.
- **All buckets in the account:** Apply permission to all current and future buckets in the account.
- **Policy:** Upload or import a JSON file compatible with the AWS IAM policy file. Specify the details of the permission at a granular level.

Apply actions using the **action** parameter.

- **All Operations:** This option allows to perform all operations on all buckets meeting the conditions applied using the type parameter.
- **Read only:** This option allows you to perform a read only operation on all buckets meeting the condition applied using the type parameter.
- **Write only:** This option allows you to write objects without reading them back on all buckets meetings

the conditions applied using the type parameter.

## Using the Policy permission file

In the example below, the policy permission has three statements:

- **Statement 1:** Allows object listing with a prefix in the bucket. It is done using a Condition element.
- **Statement 2:** Allows read and write operations for objects with the prefix in the bucket.
- **Statement 3:** Denies delete object operation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "statement1",
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::mybucket"
      ],
      "Condition": {
        "StringLike": {
          "s3:prefix": [
            "David/*"
          ]
        }
      }
    },
    {
      "Sid": "statement2",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::mybucket/David/*"
      ]
    },
    {
      "Sid": "statement3",
      "Action": [
        "s3:DeleteObject"
      ],
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::mybucket/David/*",
        "arn:aws:s3:::mycorporatebucket/share/marketing/*"
      ]
    }
  ]
}
```

```
}  
]  
}
```

The following example illustrates creating a policy permission file using Account API.

```
{  
  "name": "permission_name",  
  "description": "Test data",  
  "type": "policy",  
  "policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":\"statement15feb1\",\"Effect\":\"Allow\",\"Action\":[\"s3:*\"],\"Resource\":[\"arn:aws:s3::*/*\"]}]}"  
}
```

## Request

The POST request creates controlled access to the buckets and defines actions the service account can perform on the specified buckets in the account.

**POST/permissions**

## Body parameter

```
{  
  "name": "string",  
  "description": "string",  
  "type": "all-buckets",  
  "actions": "all-operations",  
  "prefix": "string",  
  "buckets": [ "string" ],  
  "policy": {}  
}
```

## Parameters

Name	In	Type	Required	Description
name	body	string	true	Name of the permission. The name allows only alphanumeric, '-', '_' or spaces. Maximum length is 128 characters.



Name	In	Type	Required	Description
description	body	string	true	Description of the permission. Maximum length is 1000 characters.
type	body	string	true	<p>The values for the permission type can be:</p> <ul style="list-style-type: none"> <li>• <b>all-buckets:</b> The permission is applied to all the existing and new buckets in the account. When you select the type as <b>all-buckets</b>, the parameters prefix, buckets, and policy are not part of the request.</li> <li>• <b>bucket-prefix:</b> Specify a string of characters at the beginning of the bucket's name as a prefix to apply for permission. When you select the type as <b>bucket-prefix</b>, the parameters buckets and policy are not part of the request.</li> <li>• <b>bucket-names:</b> The permission is applied to the specified bucket names. When you select the type as <b>bucket-names</b>, the parameters prefix and policy are not part of the request.</li> <li>• <b>policy:</b> Permission is applied based on the policy permission file. For more information, see <a href="#">the section called "Using the Policy permission file"</a>. When you select the type as <b>policy</b>, the parameters action, prefix, and buckets are not part of the request.</li> </ul>

Name	In	Type	Required	Description
actions	body	string	false	<p>The values for the actions can be:</p> <ul style="list-style-type: none"> <li>• <b>all-operations</b>: Allows you to perform all operations.</li> <li>• <b>read-only</b>: Allows you to perform a read only operation on one or more selected buckets and their objects.</li> <li>• <b>write-only</b>: Allows you to write objects into the selected buckets without reading them back.</li> </ul> <p>These values are applied to the permission type.</p>
prefix	body	string	false	Prefix of your bucket names to assign and apply the permission based on the permission type <b>bucket-prefix</b> .
buckets	body	[string]	false	Specify one or more bucket names based on permission type <b>bucket-names</b> .
policy	body	object	false	The policy file is based on permission type <b>policy</b> .

## Code samples

### Go

```

package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Content-Type": []string{
            "application/json",
        },
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{

```

```

    "Bearer {access-token}",
  },
}

jsonReq := `{"key":"value"}` // replace with your JSON request
data := bytes.NewBuffer([]byte(jsonReq))
req, err := http.NewRequest("PUT", "https://api.lyvecloud.seagate.com/v2/permissions/", data)
if err != nil {
    // handle error
}
req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}
// handle response _ = resp }

```

## Java

```

import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/permissions/");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("PUT");

            // For a PUT request, we need to send data
            con.setDoOutput(true);
            String jsonInputString = "{\"key\": \"value\"}"; // replace with your actual JSON data
            try(OutputStream os = con.getOutputStream()) {
                byte[] input = jsonInputString.getBytes("utf-8");
                os.write(input, 0, input.length);
            }

            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```
const inputBody = `{
  "name": "string",
  "description": "string",
  "type": "all-buckets",
  "actions": "all-operations",
  "prefix": "string",
  "buckets": ["string"],
  "policy": {}
};

const headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};

fetch("https://api.lyvecloud.seagate.com/v2/permissions/", {
  method: 'PUT',
  body: inputBody,
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });
```

## Python

```
import requests
headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}
r = requests.put("https://api.lyvecloud.seagate.com/v2/permissions/", headers=headers) print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Content-Type' => 'application/json',
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}
```

```

data = {
  "name" => "string",
  "description" => "string",
  "type" => "all-buckets",
  "actions" => "all-operations",
  "prefix" => "string",
  "buckets" => ["string"],
  "policy" => {}
}

```

```

result = RestClient.put 'https://api.lyvecloud.seagate.com/v2/permissions/', data.to_json, headers: headers

```

```

p JSON.parse(result)

```

## Responses

Status Code	Description	Return JSON Payload
200	<p>OK</p> <p>The request to create permission was successfully submitted.</p> <p>Note—There might be a few seconds difference between the time the successful response is received and when the action is completed, as some regions may still process the create permission request.</p>	<pre> {   "id": "string" } </pre>

Status Code	Description	Return JSON Payload								
400	<p>Bad Request.</p> <p>The request is invalid and has invalid permission information.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> <tr> <td>InvalidArgument</td> <td> <p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul> </td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.	InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul>
code	message									
ExpiredToken	Token expired.									
InvalidToken	Token is not valid.									
InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>• One or more of the specified arguments was not valid.</li> <li>• The request was missing an argument.</li> </ul>									
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.				
code	message									
NoServiceAvailable	The account has no services enabled for it.									

Status Code	Description	Return JSON Payload				
409	The permission name already exists.	<pre data-bbox="786 220 1446 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 430 1446 856"> <thead> <tr> <th data-bbox="786 430 1216 514">code</th> <th data-bbox="1216 430 1446 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 514 1216 856"> <span data-bbox="829 533 1172 569">PermissionNameAlreadyExists</span> </td> <td data-bbox="1216 514 1446 856">           The permission name is already in use. Please use a different name.         </td> </tr> </tbody> </table>	code	message	<span data-bbox="829 533 1172 569">PermissionNameAlreadyExists</span>	The permission name is already in use. Please use a different name.
code	message					
<span data-bbox="829 533 1172 569">PermissionNameAlreadyExists</span>	The permission name is already in use. Please use a different name.					
500	The server encountered an internal error.	<pre data-bbox="786 1003 1446 1192"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 1213 1446 1453"> <thead> <tr> <th data-bbox="786 1213 1024 1297">code</th> <th data-bbox="1024 1213 1446 1297">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 1297 1024 1453"> <span data-bbox="829 1316 977 1352">InternalError</span> </td> <td data-bbox="1024 1297 1446 1453">           The server encountered an internal error. Please retry the request.         </td> </tr> </tbody> </table>	code	message	<span data-bbox="829 1316 977 1352">InternalError</span>	The server encountered an internal error. Please retry the request.
code	message					
<span data-bbox="829 1316 977 1352">InternalError</span>	The server encountered an internal error. Please retry the request.					

Status Code	Description	Return JSON Payload				
503	Service Unavailable	<pre data-bbox="787 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="787 430 1448 709"> <thead> <tr> <th data-bbox="787 430 1075 514">code</th> <th data-bbox="1075 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 514 1075 709">ServiceNotReady</td> <td data-bbox="1075 514 1448 709">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					



# Listing Permissions

All the permissions created for the account are listed.

## Request

The **GET /permissions/** allows listing all the permissions of the account. Optionally, you can specify the name to get the specific permission in the account.

**GET /permissions**

## Parameters

Name	In	Type	Required	Description
name	query	string	false	This is an optional parameter. Query the <b>name</b> parameter for single permission.

## Code samples

### Go

```
package main

import (
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }

    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/permissions/", nil)
```

```

if err != nil {
    // handle error
}
req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}
// handle response _ = resp }

```

## Java

```

import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/permissions/");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("GET");
            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```

const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/permissions/', {
  method: 'GET',
  headers: headers
})
.then(function(res) {
  if (!res.ok) {
    throw new Error(`HTTP error! status: ${res.status}`);
  }

```

```

    }
    return res.json();
  })
  .then(function(body) {
    console.log(body);
  })
  .catch(function(error) {
    console.log('There was a problem with the fetch operation: ' + error.message); });

```

## Python

```

import requests

headers = {
    'Accept': 'application/json',
    'Authorization': 'Bearer {access-token}'
}

try:
    r = requests.get('https://api.lyvecloud.seagate.com/v2/permissions/', headers=headers)
    r.raise_for_status() # Raises a HTTPError if the response status is 4xx, 5xx
    print(r.json())
except requests.exceptions.RequestException as err:
    print ("There was a problem with the request:", err)

```

## Ruby

```

require 'rest-client'
require 'json'

headers = {
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

begin
  result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/permissions/', headers: headers
  p JSON.parse(result)
rescue RestClient::ExceptionWithResponse => e
  e.response
end

```

## Responses

Status Code	Description	Return JSON payload								
200	<p>OK</p> <p>Successfully returned permission list.</p>	<pre data-bbox="717 247 1448 646">[   {     "name": "string",     "id": "string",     "description": "string",     "type": "all-buckets",     "readyState": true,     "createTime": "2019-08-24T14:15:22Z"   } ]</pre> <table border="1" data-bbox="717 667 1448 995"> <thead> <tr> <th data-bbox="717 667 943 751">Field</th> <th data-bbox="943 667 1448 751">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="717 751 943 835">name</td> <td data-bbox="943 751 1448 835">The permission name.</td> </tr> <tr> <td data-bbox="717 835 943 919">id</td> <td data-bbox="943 835 1448 919">The permission id.</td> </tr> <tr> <td data-bbox="717 919 943 995">description</td> <td data-bbox="943 919 1448 995">Description of the permission.</td> </tr> </tbody> </table>	Field	Description	name	The permission name.	id	The permission id.	description	Description of the permission.
Field	Description									
name	The permission name.									
id	The permission id.									
description	Description of the permission.									

Status Code	Description	Field type	Description Return JSON payload
			<p>The values for the permission type can be:</p> <ul style="list-style-type: none"> <li> <b>all-buckets:</b> The permission is applied to all the existing and new buckets in the account. <p>When you select the type as <b>all-buckets</b> , the parameters prefix, buckets and policy are not part of the request.</p> </li> <li> <b>bucket-prefix:</b> Specify a string of characters at the beginning of the name of the bucket as a prefix to apply for permission. <p>When you select the type as <b>bucket-prefix</b>, the parameters buckets and policy are not part of the request.</p> </li> <li> <b>bucket-names:</b> The permission is applied to the specified buckets names. <p>When you select the type as <b>bucket-names</b>, the parameters prefix and policy are not part of the request.</p> </li> <li> <b>policy:</b> Permission is applied based on the policy permission file. For more information, see <a href="#">Managing bucket access permissions</a>. <p>When you select the type as <b>policy</b>, the parameters actions, prefix, and buckets are not part of the request.</p> </li> </ul>

Status Code	Description	Field readyState	Description Return JSON payload					
		createTime	Time of the permission creation.					
400	Bad request  Either the token is invalid or expired.	<pre data-bbox="716 527 1448 716">{   "code": "string",   "message": "string" }</pre> <table border="1" data-bbox="716 737 1448 982"> <thead> <tr> <th data-bbox="716 737 1036 821">code</th> <th data-bbox="1036 737 1448 821">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="716 821 1036 898">ExpiredToken</td> <td data-bbox="1036 821 1448 898">Token expired.</td> </tr> <tr> <td data-bbox="716 898 1036 982">InvalidToken</td> <td data-bbox="1036 898 1448 982">Token is not valid.</td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	message							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							
404	Not Found  The permission is no longer available.	<pre data-bbox="716 1029 1448 1218">{   "code": "string",   "message": "string" }</pre> <table border="1" data-bbox="716 1239 1448 1442"> <thead> <tr> <th data-bbox="716 1239 1036 1323">code</th> <th data-bbox="1036 1239 1448 1323">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="716 1323 1036 1442">PermissionNotFound</td> <td data-bbox="1036 1323 1448 1442">Permission was not found.</td> </tr> </tbody> </table>	code	message	PermissionNotFound	Permission was not found.		
code	message							
PermissionNotFound	Permission was not found.							
403	Forbidden  The account has no services enabled.	<pre data-bbox="716 1491 1448 1680">{   "code": "string",   "message": "string" }</pre> <table border="1" data-bbox="716 1701 1448 1904"> <thead> <tr> <th data-bbox="716 1701 1036 1785">code</th> <th data-bbox="1036 1701 1448 1785">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="716 1785 1036 1904">NoServiceAvailable</td> <td data-bbox="1036 1785 1448 1904">The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							

Status Code	Description	Return JSON payload				
500	The server encountered an internal error	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalServerError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceNotReady</td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Getting Permission by ID

Permission ID is a unique identifier for a permission.

## Request

Performing a GET operation on specific permission retrieves the permission.

```
GET /permissions/{permissionId}
```

## Parameters

Name	In	Type	Required	Description
permissionId	path	string	true	Numeric ID of the permission.

## Code samples

### Go

```
package main

import (
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }

    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}", nil)
    if err != nil {
        // handle error
    }
}
```



```

}
req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}
// handle response _ = resp }

```

## Java

```

import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("GET");
            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```

const headers = {
    'Accept': 'application/json',
    'Authorization': 'Bearer {access-token}'
};

fetch("https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}", {
    method: 'GET',
    headers: headers
})
.then(function(res) {
    if (!res.ok) {
        throw new Error(`HTTP error! status: ${res.status}`);
    }
    return res.json();
})

```

```

    })
    .then(function(body) {
      console.log(body);
    })
    .catch(function(error) {
      console.log('There was a problem with the fetch operation: ' + error.message); });

```

## Python

```

import requests

headers = {
    'Accept': 'application/json',
    'Authorization': 'Bearer {access-token}'
}

try:
    r = requests.get('https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', headers=headers)
    r.raise_for_status() # Raises a HTTPError if the response status is 4xx, 5xx
    print(r.json())
except requests.exceptions.RequestException as err:
    print ("There was a problem with the request:", err)

```

## Ruby

```

require 'rest-client'
require 'json'

headers = {
    'Accept' => 'application/json',
    'Authorization' => 'Bearer {access-token}'
}

result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', params: {}, headers: headers

puts JSON.parse(result)

```

## Responses

Status Code	Description	Return JSON Payload
200	OK Specified Permission	<pre>{   "id": "string",</pre>

Status Code	Description	Return JSON Payload								
	ID is found, and the details are returned in the response.	<pre data-bbox="737 79 1448 420"> "name": "string", "description": "string", "type": "all-buckets", "readyState": true, "actions": "all-operations", "prefix": "string", "buckets": [ "string" ], "policy": {} } </pre> <p data-bbox="737 478 1398 548">Note—If the type is <b>policy</b>, then the fields actions, prefix, and buckets will not display.</p> <table border="1" data-bbox="737 594 1448 919"> <thead> <tr> <th data-bbox="737 594 956 674">Field</th> <th data-bbox="956 594 1448 674">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="737 674 956 753">id</td> <td data-bbox="956 674 1448 753">ID of the permission.</td> </tr> <tr> <td data-bbox="737 753 956 833">name</td> <td data-bbox="956 753 1448 833">Name of the permission.</td> </tr> <tr> <td data-bbox="737 833 956 919">description</td> <td data-bbox="956 833 1448 919">Description of the permission.</td> </tr> </tbody> </table>	Field	Description	id	ID of the permission.	name	Name of the permission.	description	Description of the permission.
Field	Description									
id	ID of the permission.									
name	Name of the permission.									
description	Description of the permission.									

Status Code	Description	Field	Return JSON Payload Description
		type	<p>The values for the permission type can be:</p> <ul style="list-style-type: none"> <li> <b>all-operations:</b> Allows to perform all the operations.           <p>When you select the type as <b>all-operations</b>, the parameters prefix, buckets and policy are not part of the request.</p> </li> <li> <b>bucket-prefix:</b> Specify a string of characters at the beginning of the bucket's name as a prefix to apply for permission.           <p>When you select the type as <b>bucket-prefix</b>, the parameters buckets and policy are not part of the request.</p> </li> <li> <b>bucket-names:</b> Permission is applied to the specified bucket names.           <p>When you select the type as <b>bucket-names</b>, the parameters prefix and policy are not part of the request.</p> </li> <li> <b>policy:</b> Permission is applied based on the policy permission file. For more information, see <a href="#">Using policy permission files</a>.           <p>When you select the type as <b>policy</b>, the parameters actions, prefix, and buckets are not part of the request.</p> </li> </ul>

Status Code	Description	Field	Return JSON Payload Description						
		<p><code>readyState</code></p> <p><code>actions</code></p> <p><code>prefix</code></p> <p><code>buckets</code></p> <p><code>policy</code></p>	<p>State of the permissions. True if the permission is ready across all regions.</p> <p>The values for the actions can be:</p> <ul style="list-style-type: none"> <li>• <b>all-operations</b>: Allows you to perform all operations.</li> <li>• <b>read-only</b>: Allows you to perform a read only operation on one or more selected buckets and their objects.</li> <li>• <b>write-only</b>: This allows you to write objects into the selected buckets without reading them back.</li> </ul> <p>Specify a prefix of your bucket names to assign and apply the permission based on the permission type <b>bucket-prefix</b>.</p> <p>Specify one or more bucket names based on permission type.</p> <p>The policy file based on permission type <b>policy</b>.</p>						
400	Either the token is not valid or expired.		<pre data-bbox="737 1472 1448 1656">{   "code": "string",   "message": "string" }</pre> <table border="1" data-bbox="737 1682 1448 1923"> <thead> <tr> <th data-bbox="737 1682 1045 1761">code</th> <th data-bbox="1045 1682 1448 1761">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="737 1761 1045 1841"><code>ExpiredToken</code></td> <td data-bbox="1045 1761 1448 1841">Token expired.</td> </tr> <tr> <td data-bbox="737 1841 1045 1923"><code>InvalidToken</code></td> <td data-bbox="1045 1841 1448 1923">Token is not valid.</td> </tr> </tbody> </table>	code	message	<code>ExpiredToken</code>	Token expired.	<code>InvalidToken</code>	Token is not valid.
code	message								
<code>ExpiredToken</code>	Token expired.								
<code>InvalidToken</code>	Token is not valid.								

Status Code	Description	Return JSON Payload				
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.
code	message					
NoServiceAvailable	The account has no services enabled for it.					
404	<p>Not Found</p> <p>The permission is not available.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>PermissionNotFound</td> <td>The permission was not found.</td> </tr> </tbody> </table>	code	message	PermissionNotFound	The permission was not found.
code	message					
PermissionNotFound	The permission was not found.					
500	<p>An internal error has occurred.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalServerError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					

Status Code	Description	Return JSON Payload				
503	Service Unavailable	<div data-bbox="737 220 1448 409" style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"> <pre>{   "code": "string",   "message": "string" }</pre> </div> <table border="1" data-bbox="737 430 1448 709"> <thead> <tr> <th data-bbox="737 430 1026 514">code</th> <th data-bbox="1026 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="737 514 1026 709">ServiceNotReady</td> <td data-bbox="1026 514 1448 709">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Deleting Permission by ID

If the permission is associated with a service account, you cannot delete that permission. Also, once you delete permission, you cannot restore it.

## Request

Performing a **Delete** permission operation removes existing permissions from the database.

```
DELETE /permissions/{permissionId}
```

## Parameters

Name	In	Type	Required	Description
permissionId	path	string	true	Numeric ID of the permission to delete.

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "text/plain",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }

    jsonReq := "{json request body}" // replace with your JSON request body
```



```

data := bytes.NewBuffer([]byte(jsonReq))

req, err := http.NewRequest("DELETE", "https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}", data)
if err != nil {
    // handle error
}

req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}

// handle response }

```

## Java

```

try {
    URL obj = new URL("https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}");
    HttpURLConnection con = (HttpURLConnection) obj.openConnection();
    con.setRequestMethod("DELETE");

    int responseCode = con.getResponseCode();
    System.out.println("Response Code : " + responseCode);

    BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
    String inputLine;
    StringBuffer response = new StringBuffer();

    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
    in.close();

    System.out.println(response.toString());

} catch (Exception e) { e.printStackTrace(); }

```

## JavaScript

```

const headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', {
  method: 'DELETE',
  headers: headers

```

```

    })
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(body => {
      console.log(body);
    })
    .catch(error => {
      console.error('There has been a problem with your fetch operation:', error); });

```

## Python

```

import requests

headers = {
    'Accept': 'text/plain',
    'Authorization': 'Bearer {access-token}'
}

try:
    r = requests.delete('https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', headers=headers)
    r.raise_for_status() # Raises a HTTPError if the response status is 4xx, 5xx
except requests.exceptions.RequestException as err:
    print(f"An error occurred: {err}")
else: print(r.json())

```

## Ruby

```

require 'rest-client'
require 'json'

headers = {
  'Accept' => 'text/plain',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.delete 'https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', params: {}, headers: headers

puts JSON.parse(result)

```

## Responses

Status Code	Description	Returned JSON payload						
200	OK The delete operation is successful.	Permission deleted successfully.						
400	Bad request	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>code</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	code	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	code							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							
403	Forbidden The account has no services enabled.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							
404	Not Found The permission to be deleted is not found.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>PermissionNotFound</td> <td>Permission was not found.</td> </tr> </tbody> </table>	code	message	PermissionNotFound	Permission was not found.		
code	message							
PermissionNotFound	Permission was not found.							

Status Code	Description	Returned JSON payload				
409	<p>Conflict</p> <p>The permission is not ready for deletion operation and is still being processed by some regions.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>PermissionNotReady</td> <td>Permission is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	PermissionNotReady	Permission is still being processed by some regions.
code	message					
PermissionNotReady	Permission is still being processed by some regions.					
500	<p>The server has encountered an internal error</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalServerError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					
503	<p>Service Unavailable</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceNotReady</td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Updating a Permission

You can update all the permission parameters, bucket permission, and their associated actions.

## Request

Performing a PUT operation on specific a permission updates the permission.

```
PUT /permissions/{permissionId}
```

## Body parameter

```
{
  "name": "string",
  "description": "string",
  "type": "bucket-names",
  "actions": "all-operations",
  "prefix": "string",
  "buckets": [ "string" ],
  "policy": {}
}
```

## Parameters

Name	In	Type	Required	Description
permissionId	path	string	true	Numeric ID of the permission to update.
name	body	string	true	Name of the permission.
description	body	string	true	Description of the permission.

Name	In	Type	Required	Description
type	body	string	true	<p>The values for the permission type can be:</p> <ul style="list-style-type: none"> <li> <p><b>all-buckets:</b> The permission is applied to all the existing buckets in the account and to new buckets.</p> <p>When you select the type as <b>all-buckets</b>, the parameters prefix, buckets and policy are not part of the request.</p> </li> <li> <p><b>bucket-prefix:</b> Specify a string of characters at the beginning of the bucket's name as a prefix to apply for permission.</p> <p>When you select the type as <b>bucket-prefix</b>, the parameters buckets and policy are not part of the request.</p> </li> <li> <p><b>bucket-names:</b> Permission is applied to the specified bucket names.</p> <p>When you select the type as <b>bucket-names</b>, the parameters prefix and policy are not part of the request.</p> </li> <li> <p><b>policy:</b> Using the Policy permission, you can allow or deny requests at a granular level based on the elements in the policy, resources, and conditions of the request.</p> <p>When you select the type as <b>policy</b>, the parameters actions, prefix, and buckets are not part of the request.</p> </li> </ul>

Name	In	Type	Required	Description
actions	body	string	false	<p>The values for the actions can be:</p> <ul style="list-style-type: none"> <li>• <b>all-operations</b>: Allows you to perform all the operations.</li> <li>• <b>read-only</b>: Allows you to perform a read only operation on one or more selected buckets and their objects.</li> <li>• <b>write-only</b>: Allows you to write objects into the selected buckets without reading them back.</li> </ul> <p>These values are applied to the permission type.</p>
prefix	body	string	false	Prefix of your bucket names to assign and apply the permission based on the permission type <b>bucket-prefix</b> .
buckets	body	string	false	Specify one or more bucket names based on permission type <b>bucket-names</b> .
policy	body	object	false	The policy file is based on permission type <b>policy</b> . For more information, see <a href="#">Managing bucket access permissions</a> .

## Code samples

### Go

```

package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Content-Type": []string{
            "application/json",
        },
    },

```

```

    "Accept": []string{
        "application/json",
    },
    "Authorization": []string{
        "Bearer {access-token}",
    },
}

jsonReq := "{json request body}" // replace with your JSON request body
data := bytes.NewBuffer([]byte(jsonReq))

req, err := http.NewRequest("POST", "https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}", data)
if err != nil {
    // handle error
}

req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}

// handle response }

```

## Java

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("POST");

            // Set request headers
            con.setRequestProperty("Content-Type", "application/json");
            con.setRequestProperty("Accept", "application/json");
            con.setRequestProperty("Authorization", "Bearer {access-token}");

            int responseCode = con.getResponseCode();
            System.out.println("Response Code : " + responseCode);

            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();

```



```

while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
in.close();

System.out.println(response.toString());

} catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```

const inputBody = { "name": "string", "description": "string", "type": "all-buckets", "actions": "all-operations", "prefix": "string"
, "buckets": [ "string" ], "policy": {} };
const headers = {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
    'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', {
    method: 'POST',
    body: inputBody,
    headers: headers
})
.then(response => {
    if (!response.ok) {
        throw new Error("Network response was not ok");
    }
    return response.json(); }) .then(body => { console.log(body); }) .catch(error => { console.error("There has been a problem
with your fetch operation:", error); });

```

## Python

```

import requests

headers = {
    'Content-Type': 'application/json',
    'Accept': 'application/json',
    'Authorization': 'Bearer {access-token}'
}

data = { "name": "string", "description": "string", "type": "all-buckets", "actions": "all-operations", "prefix": "string", "buckets":
[ "string" ], "policy": {} } # replace with your actual data

try:
    r = requests.post('https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}', headers=headers, data=data)
    r.raise_for_status() # Raises a HTTPError if the response status is 4xx, 5xx
except requests.exceptions.RequestException as err:
    print(f"An error occurred: {err}")
else: print(r.json())

```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Content-Type' => 'application/json',
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.post(
  'https://api.lyvecloud.seagate.com/v2/permissions/{PermissionId}',
  {}.to_json, # Convert params to JSON
  headers
)

puts JSON.parse(result) # Use 'puts' instead of 'p' for printing
```

## Responses

Status	Description	Return JSON Payload
200	OK  The update operation is successful.	Successfully updated the permission.

Status	Description	Return JSON Payload								
400	<p>Bad Request</p> <p>It is an invalid request or invalid permission information.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> <tr> <td>InvalidArgument</td> <td> <p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul> </td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.	InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>
code	message									
ExpiredToken	Token expired.									
InvalidToken	Token is not valid.									
InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>									
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.				
code	message									
NoServiceAvailable	The account has no services enabled for it.									

Status	Description	Return JSON Payload						
404	<p>Not Found</p> <p>The permission to be updated is no longer available.</p>	<pre data-bbox="787 184 1448 369"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="787 394 1448 596"> <thead> <tr> <th data-bbox="787 394 1112 474">code</th> <th data-bbox="1112 394 1448 474">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 474 1112 596">PermissionNotFound</td> <td data-bbox="1112 474 1448 596">Permission was not found.</td> </tr> </tbody> </table>	code	message	PermissionNotFound	Permission was not found.		
code	message							
PermissionNotFound	Permission was not found.							
409	<p>The permission name already exists</p> <p>The permission is not ready for update operation and is being processed by some regions.</p>	<pre data-bbox="787 646 1448 831"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="787 856 1448 1549"> <thead> <tr> <th data-bbox="787 856 1218 936">code</th> <th data-bbox="1218 856 1448 936">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 936 1218 1281">PermissionNameAlreadyExists</td> <td data-bbox="1218 936 1448 1281">The permission name is already in use. Please use a different name.</td> </tr> <tr> <td data-bbox="787 1281 1218 1549">PermissionNotReady</td> <td data-bbox="1218 1281 1448 1549">Permission is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	PermissionNameAlreadyExists	The permission name is already in use. Please use a different name.	PermissionNotReady	Permission is still being processed by some regions.
code	message							
PermissionNameAlreadyExists	The permission name is already in use. Please use a different name.							
PermissionNotReady	Permission is still being processed by some regions.							

Status	Description	Return JSON Payload				
500	<p>Locked</p> <p>The server encountered an internal error.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalError	The server encountered an internal error. Please retry the request.
code	message					
InternalError	The server encountered an internal error. Please retry the request.					
503	<p>Service Unavailable</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceNotReady</td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Creating a Service Account

Service accounts allow applications to authenticate and access Lyve Cloud buckets and objects. When creating a service account, the access key and secret key are generated. This information must be saved at account creation, as you cannot recover key details afterwards. You must create buckets and assign permission to buckets before creating a service account. For more information, see [Administrator's Guide - Bucket Management](#) and [Managing bucket access permissions](#).



**Note**—You must have at least one permission before creating a service account.

## Request

The POST request creates a service account. The service account generates the credentials holding the access and secret keys.

**POST** /service-accounts

## Body parameter

```
{
  "name": "string",
  "description": "string",
  "permissions": [ "string" ]
}
```

## Parameters

Name	In	Type	Required	Description
name	body	string	true	Name of the new service account.  The name allows only alphanumeric, '-', '_' or space. Maximum length is 128 characters.

Name	In	Type	Required	Description
description	body	string	false	Description of the new service account.
permissions	body	[string]	true	Permission ID to apply to the service account.

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Content-Type": {
            "application/json",
        },
        "Accept": {
            "application/json",
        },
        "Authorization": {
            "Bearer {access-token}",
        },
    }

    jsonReq := "{}" // Replace with your JSON request
    data := bytes.NewBuffer([]byte(jsonReq))

    req, err := http.NewRequest("PUT", "https://api.lyvecloud.seagate.com/v2/service-accounts/", data)
    if err != nil {
        // handle error
    }

    req.Header = headers

    client := &http.Client{}
    resp, err := client.Do(req)
    if err != nil {
        // handle error
    }

    // handle response _ = resp }
```

## Java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;

public class Main {
    public static void main(String[] args) throws Exception {
        URL url = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/");
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("PUT");

        // Set headers
        con.setRequestProperty("Content-Type", "application/json");
        con.setRequestProperty("Accept", "application/json");
        con.setRequestProperty("Authorization", "Bearer {access-token}");

        // Enable input and output streams
        con.setDoOutput(true);
        con.setDoInput(true);

        // Write request body
        String jsonInputString = "{}"; // Replace with your JSON request
        try(OutputStream os = con.getOutputStream()) {
            byte[] input = jsonInputString.getBytes("utf-8");
            os.write(input, 0, input.length);
        }

        // Get response code
        int responseCode = con.getResponseCode();
        System.out.println("Response Code : " + responseCode);

        // Read response
        try(BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "utf-8"))) {
            StringBuilder response = new StringBuilder();
            String responseLine = null;
            while ((responseLine = br.readLine()) != null) { response.append(responseLine.trim()); } System.out.println(response.toString()); } } }
```

## JavaScript

```
const inputBody = JSON.stringify({
    "name": "string",
    "description": "string",
    "permissions": ["string"]
});

const headers = {
```



```
'Content-Type': 'application/json',
'Accept': 'application/json',
'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/', {
  method: 'PUT', body: inputBody, headers: headers }) .then(res => res.json()) .then(body => console.log(body));
```

## Python

```
import requests

headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}

# You should have a payload for a PUT request
data = {} # Replace with your actual data

r = requests.put('https://api.lyvecloud.seagate.com/v2/service-accounts/', headers=headers, json=data) print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Content-Type' => 'application/json',
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

# You should have a payload for a PUT request
params = {} # Replace with your actual data

result = RestClient.put(
  'https://api.lyvecloud.seagate.com/v2/service-accounts/',
  params.to_json, # Convert params to JSON
  headers
)

puts JSON.parse(result) # Use 'puts' instead of 'p' for printing
```

## Responses

Status Code	Description	Return JSON Payload										
200	<p>OK</p> <p>The request to create a Service Account was successfully sent.</p> <p>Note—There might be few seconds difference between the time the successful response is received, and when the action is completed, as some regions may still process the create service account request.</p> <p>If you need to start using the service account immediately, we suggest you wait for a few seconds and perform a retry operation until <code>readyState=True</code>. For more information, see <a href="#">Getting service account data by ID</a>.</p>	<pre data-bbox="760 222 1448 478"> {   "id": "string",   "accessKey": "string",   "secret": "string"   "expirationDate": "2022-11-15" } </pre> <table border="1" data-bbox="760 506 1448 1100"> <thead> <tr> <th data-bbox="760 506 1015 583">Field</th> <th data-bbox="1015 506 1448 583">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="760 583 1015 667"><code>id</code></td> <td data-bbox="1015 583 1448 667">ID of the service account.</td> </tr> <tr> <td data-bbox="760 667 1015 825"><code>accessKey</code></td> <td data-bbox="1015 667 1448 825">Access key generated after creating a service account.</td> </tr> <tr> <td data-bbox="760 825 1015 982"><code>secret</code> :</td> <td data-bbox="1015 825 1448 982">Secret key generated after creating a service account.</td> </tr> <tr> <td data-bbox="760 982 1015 1100"><code>expirationDate</code></td> <td data-bbox="1015 982 1448 1100">Date when the service account will expire.</td> </tr> </tbody> </table>	Field	Description	<code>id</code>	ID of the service account.	<code>accessKey</code>	Access key generated after creating a service account.	<code>secret</code> :	Secret key generated after creating a service account.	<code>expirationDate</code>	Date when the service account will expire.
Field	Description											
<code>id</code>	ID of the service account.											
<code>accessKey</code>	Access key generated after creating a service account.											
<code>secret</code> :	Secret key generated after creating a service account.											
<code>expirationDate</code>	Date when the service account will expire.											

Status Code	Description	Return JSON Payload								
400	<p>Bad Request</p> <p>Invalid service account information.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> <tr> <td>InvalidArgument</td> <td> <p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul> </td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.	InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>
code	message									
ExpiredToken	Token expired.									
InvalidToken	Token is not valid.									
InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>									
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.				
code	message									
NoServiceAvailable	The account has no services enabled for it.									

Status Code	Description	Return JSON Payload				
409	The service account name already exists. The service account name must be unique.	<pre data-bbox="760 222 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="760 432 1448 951"> <thead> <tr> <th data-bbox="760 432 1235 514">code</th> <th data-bbox="1235 432 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="760 514 1235 951"> <div data-bbox="802 533 1195 569" style="border: 1px solid red; padding: 2px; display: inline-block;">ServiceAccountNameAlreadyExists</div> </td> <td data-bbox="1235 514 1448 951">The service account name is already in use. Please use a different name.</td> </tr> </tbody> </table>	code	message	<div data-bbox="802 533 1195 569" style="border: 1px solid red; padding: 2px; display: inline-block;">ServiceAccountNameAlreadyExists</div>	The service account name is already in use. Please use a different name.
code	message					
<div data-bbox="802 533 1195 569" style="border: 1px solid red; padding: 2px; display: inline-block;">ServiceAccountNameAlreadyExists</div>	The service account name is already in use. Please use a different name.					
500	Server encountered an internal error.	<pre data-bbox="760 1278 1448 1465"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="760 1488 1448 1726"> <thead> <tr> <th data-bbox="760 1488 997 1570">code</th> <th data-bbox="997 1488 1448 1570">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="760 1570 997 1726"> <div data-bbox="802 1589 954 1625" style="border: 1px solid red; padding: 2px; display: inline-block;">InternalError</div> </td> <td data-bbox="997 1570 1448 1726">The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<div data-bbox="802 1589 954 1625" style="border: 1px solid red; padding: 2px; display: inline-block;">InternalError</div>	The server encountered an internal error. Please retry the request.
code	message					
<div data-bbox="802 1589 954 1625" style="border: 1px solid red; padding: 2px; display: inline-block;">InternalError</div>	The server encountered an internal error. Please retry the request.					

Status Code	Description	Return JSON Payload				
503	Service Unavailable	<pre data-bbox="760 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="760 430 1448 709"> <thead> <tr> <th data-bbox="760 430 1047 514">code</th> <th data-bbox="1047 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="760 514 1047 709">ServiceNotReady</td> <td data-bbox="1047 514 1448 709">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Listing Service Accounts

All service accounts in the account are listed or a specific service account is listed. Each service account has a unique identifier.

## Request

The **GET** API lists all the service accounts or a specific service account.

```
GET /service-accounts
```

## Parameters

Name	In	Type	Required	Description
name	query	string	false	This is an optional parameter. Use the parameter to query for a single service account by name.

## Code samples

### Go

```
package main

import (
    "net/http"
    "io/ioutil"
    "fmt"
)

func main() {
    headers := map[string][]string{
        "Accept": {
            "application/json",
        },
    },
```

```

    "Authorization": {
        "Bearer {access-token}",
    },
}

req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/service-accounts/", nil)
if err != nil {
    // handle error
}

req.Header = headers

client := &http.Client{}
resp, err := client.Do(req)
if err != nil {
    // handle error
}

// Read response
defer resp.Body.Close()
body, _ := ioutil.ReadAll(resp.Body)

fmt.Println(string(body)) }

```

## Java

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class Main {
    public static void main(String[] args) throws Exception {
        URL url = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/");
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("GET");

        // Set headers
        con.setRequestProperty("Accept", "application/json");
        con.setRequestProperty("Authorization", "Bearer {access-token}");

        // Get response code
        int responseCode = con.getResponseCode();
        System.out.println("Response Code : " + responseCode);

        // Read response
        try(BufferedReader br = new BufferedReader(new InputStreamReader(con.getInputStream(), "utf-8"))) {
            StringBuilder response = new StringBuilder();
            String responseLine = null;
            while ((responseLine = br.readLine()) != null) { response.append(responseLine.trim()); } System.out.println(response.toString()); } } }

```

## JavaScript

```
const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};
fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/', {
  method: 'GET',
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });
```

## Python

```
import requests
headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}
r = requests.get('https://api.lyvecloud.seagate.com/v2/service-accounts/', headers = headers) print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/service-accounts/', params: {}, headers: headers

puts JSON.parse(result)
```

## Responses

Status Code	Description	Schema
-------------	-------------	--------



Status Code	Description	Schema												
200	<p>OK</p> <p>The operation is successful.</p>	<pre data-bbox="786 220 1448 585">[   {     "name": "string",     "id": "string",     "enabled": true,     "readyState": true,     "description": "string"   } ]</pre> <table border="1" data-bbox="786 611 1448 1287"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>name</code></td> <td>Name of the service account.</td> </tr> <tr> <td><code>id</code></td> <td>ID of a service account.</td> </tr> <tr> <td><code>enabled</code></td> <td>State of the Service Account. It can be enabled or disabled.</td> </tr> <tr> <td><code>readyState</code></td> <td>True if the service account is ready across all regions.</td> </tr> <tr> <td><code>description</code></td> <td>Description of the service account.</td> </tr> </tbody> </table>	Field	Description	<code>name</code>	Name of the service account.	<code>id</code>	ID of a service account.	<code>enabled</code>	State of the Service Account. It can be enabled or disabled.	<code>readyState</code>	True if the service account is ready across all regions.	<code>description</code>	Description of the service account.
Field	Description													
<code>name</code>	Name of the service account.													
<code>id</code>	ID of a service account.													
<code>enabled</code>	State of the Service Account. It can be enabled or disabled.													
<code>readyState</code>	True if the service account is ready across all regions.													
<code>description</code>	Description of the service account.													
400	<p>Bad request</p> <p>The token is either invalid or expired.</p>	<pre data-bbox="786 1386 1448 1572">{   "code": "string",   "message": "string" }</pre> <table border="1" data-bbox="786 1598 1448 1841"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ExpiredToken</code></td> <td>Token expired.</td> </tr> <tr> <td><code>InvalidToken</code></td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	message	<code>ExpiredToken</code>	Token expired.	<code>InvalidToken</code>	Token is not valid.						
code	message													
<code>ExpiredToken</code>	Token expired.													
<code>InvalidToken</code>	Token is not valid.													

Status Code	Description	Schema				
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.
code	message					
NoServiceAvailable	The account has no services enabled for it.					
404	<p>The service account was not found.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceAccountNotFound</td> <td>Service account was not found.</td> </tr> </tbody> </table>	code	message	ServiceAccountNotFound	Service account was not found.
code	message					
ServiceAccountNotFound	Service account was not found.					
500	<p>The server encountered an internal error.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalServerError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					

Status Code	Description	Schema				
503	Service Unavailable	<pre data-bbox="787 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="787 430 1448 709"> <thead> <tr> <th data-bbox="787 430 1075 514">code</th> <th data-bbox="1075 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 514 1075 709">ServiceNotReady</td> <td data-bbox="1075 514 1448 709">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Getting Service Account Data by ID

You can retrieve a specific service account using the unique identifier.

## Request

The GET operation returns the service accounts as specified in the parameters.

```
GET /service-accounts/{serviceAccountId}
```

## Parameters

Name	In	Type	Required	Description
serviceAccountId	path	string	true	Numeric ID of the service account.

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    data := bytes.NewBuffer([]byte{jsonReq})
    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}", data)
    req.Header = headers
```

```
client := &http.Client{}
resp, err := client.Do(req)
// ... }
```

## Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws Exception {
        URL obj = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}");
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("GET");
        int responseCode = con.getResponseCode();
        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine); } in.close(); System.out.println(response.toString()); } }
```

## JavaScript

```
const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', {
  method: 'GET', headers: headers }) .then(res => res.json()) .then(body => console.log(body));
```

## Python

```
import requests

headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}

r = requests.get('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', headers=headers) print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', params: {}, headers: headers

puts JSON.parse(result)
```

## Responses

Status Code	Description	Schema
-------------	-------------	--------

Status Code	Description	Schema														
200	OK	<pre data-bbox="787 220 1448 550"> {   "id": "string",   "name": "string",   "description": "string",   "enabled": true,   "readyState": true,   "permissions": [ "string" ] } </pre> <table border="1" data-bbox="787 573 1448 1467"> <thead> <tr> <th data-bbox="787 573 1018 657">Field</th> <th data-bbox="1018 573 1448 657">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 657 1018 737">id</td> <td data-bbox="1018 657 1448 737">ID of the service account.</td> </tr> <tr> <td data-bbox="787 737 1018 816">name</td> <td data-bbox="1018 737 1448 816">Service account name.</td> </tr> <tr> <td data-bbox="787 816 1018 936">description</td> <td data-bbox="1018 816 1448 936">Service account description.</td> </tr> <tr> <td data-bbox="787 936 1018 1094">enabled</td> <td data-bbox="1018 936 1448 1094">Specifies if the service account is enabled or disabled.</td> </tr> <tr> <td data-bbox="787 1094 1018 1312">readyState</td> <td data-bbox="1018 1094 1448 1312">State of the service account. True if the service account is ready across all regions.</td> </tr> <tr> <td data-bbox="787 1312 1018 1467">permissions</td> <td data-bbox="1018 1312 1448 1467">Permission name associated with the service account.</td> </tr> </tbody> </table>	Field	Description	id	ID of the service account.	name	Service account name.	description	Service account description.	enabled	Specifies if the service account is enabled or disabled.	readyState	State of the service account. True if the service account is ready across all regions.	permissions	Permission name associated with the service account.
Field	Description															
id	ID of the service account.															
name	Service account name.															
description	Service account description.															
enabled	Specifies if the service account is enabled or disabled.															
readyState	State of the service account. True if the service account is ready across all regions.															
permissions	Permission name associated with the service account.															

Status Code	Description	Schema						
400	<p>Bad request</p> <p>The token is either invalid or expired.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	message							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							
404	<p>The service account was not found.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceAccountNotFound</td> <td>Service account was not found.</td> </tr> </tbody> </table>	code	message	ServiceAccountNotFound	Service account was not found.		
code	message							
ServiceAccountNotFound	Service account was not found.							



Status Code	Description	Schema				
500	The server encountered an internal error.	<pre data-bbox="786 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 430 1448 669"> <thead> <tr> <th data-bbox="786 430 1024 514">code</th> <th data-bbox="1024 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 514 1024 669">InternalError</td> <td data-bbox="1024 514 1448 669">The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalError	The server encountered an internal error. Please retry the request.
code	message					
InternalError	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre data-bbox="786 716 1448 905"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 926 1448 1205"> <thead> <tr> <th data-bbox="786 926 1075 1010">code</th> <th data-bbox="1075 926 1448 1010">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 1010 1075 1205">ServiceNotReady</td> <td data-bbox="1075 1010 1448 1205">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Updating a Service Account

You can update the name and permissions associated with the service account. The secret credentials are not generated when you update.

## Request

The PUT operation updates the existing service account.

```
PUT /service-accounts/{serviceAccountId}
```

## Body parameter

```
{
  "name": "string",
  "description": "string",
  "permissions": [ "string" ]
}
```

## Parameters

Name	In	Type	Required	Description
serviceAccountId	path	string	true	Numeric ID of the service account.
name	body	string	true	Name of the service account. The name allows only alphanumeric, '-', '_' or space. Maximum length is 128 characters.
description	body	string	false	Description of the service account.
permissions	body	string	true	Permissions associated with the service account.

# Code samples

## Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Content-Type": []string{
            "application/json",
        },
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    data := bytes.NewBuffer([]byte{jsonReq})
    req, err := http.NewRequest("POST", "https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}", data)
    req.Header = headers
    client := &http.Client{}
    resp, err := client.Do(req)
    // ... }
```

## Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws Exception {
        URL obj = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}");
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("POST");
        int responseCode = con.getResponseCode();
        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine); } in.close(); System.out.println(response.toString()); } }
```

## JavaScript

```
const inputBody = `{
  "name": "string",
  "description": "string",
  "permissions": ["string"]
}`;

const headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', {
  method: 'POST', body: inputBody, headers: headers }) .then(res => res.json()) .then(body => console.log(body));
```

## Python

```
import requests

headers = {
  'Content-Type': 'application/json',
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}

r = requests.post('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', headers=headers) print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Content-Type' => 'application/json',
  'Accept' => 'application/json',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.post 'https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', params: {}, headers: headers

puts JSON.parse(result)
```

# Responses

Status Code	Description	Return JSON Payload								
200	<p>OK</p> <p>The update operation is successful.</p>	Successfully updated the service account.								
400	It is an invalid request or invalid service account information.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> <tr> <td>InvalidArgument</td> <td> <p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul> </td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.	InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>
code	message									
ExpiredToken	Token expired.									
InvalidToken	Token is not valid.									
InvalidArgument	<p>This error might occur for the following reasons:</p> <ul style="list-style-type: none"> <li>One or more of the specified arguments was not valid.</li> <li>The request was missing an argument.</li> </ul>									
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.				
code	message									
NoServiceAvailable	The account has no services enabled for it.									

Status Code	Description	Return JSON Payload						
404	The service account to update was not found.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceAccountNotFound</code></td> <td>A service account was not found.</td> </tr> </tbody> </table>	code	message	<code>ServiceAccountNotFound</code>	A service account was not found.		
code	message							
<code>ServiceAccountNotFound</code>	A service account was not found.							
409	<p>The service account name already exists. The service account name must be unique.</p> <p><i>or</i></p> <p>The service account is not ready for update operation as it is locked and processed by some regions.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceAccountNameAlreadyExists</code></td> <td>The service account name is already in use. Please use a different name.</td> </tr> <tr> <td><code>ServiceAccountNotReady</code></td> <td>The service account is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	<code>ServiceAccountNameAlreadyExists</code>	The service account name is already in use. Please use a different name.	<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.
code	message							
<code>ServiceAccountNameAlreadyExists</code>	The service account name is already in use. Please use a different name.							
<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.							

Status Code	Description	Return JSON Payload				
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>InternalError</td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalError	The server encountered an internal error. Please retry the request.
code	message					
InternalError	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceNotReady</td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Enabling a Service Account

You can enable a specific service account and use the secret credentials associated with the account.

## Request

The **PUT** operation enables the existing service account.

```
PUT /service-accounts/{serviceAccountId}/enabled
```

## Parameters

Name	In	Type	Required	Description
serviceAccountId	path	string	true	Numeric ID of the service account.

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "text/plain",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
}
```



```
data := bytes.NewBuffer([]byte{jsonReq})
req, err := http.NewRequest("PUT", "https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled", data)
req.Header = headers
client := &http.Client{}
resp, err := client.Do(req) // ... }
```

## Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws Exception {
        URL obj = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled");
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("PUT");
        int responseCode = con.getResponseCode();
        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine); }
        in.close();
        System.out.println(response.toString());
    }
}
```

## JavaScript

```
const headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
};
fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', {
  method: 'PUT',
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });
```

## Python

```
import requests

headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
}

r = requests.put('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', headers=headers)
print(r.text)
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Accept' => 'text/plain',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.put 'https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', params: {}, headers: headers

puts result
```

## Responses

Status Code	Description	Return JSON Payload						
200	Service Account enabled successfully.	-						
400	It is an invalid request or invalid service account information.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"><thead><tr><th>code</th><th>message</th></tr></thead><tbody><tr><td>ExpiredToken</td><td>Token expired.</td></tr><tr><td>InvalidToken</td><td>Token is not valid.</td></tr></tbody></table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	message							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							

Status Code	Description	Return JSON Payload						
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre data-bbox="850 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 430 1448 709"> <thead> <tr> <th data-bbox="850 430 1161 514">code</th> <th data-bbox="1161 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 514 1161 709">NoServiceAvailable</td> <td data-bbox="1161 514 1448 709">The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							
404	<p>The service account to enable was not found or expired.</p>	<pre data-bbox="850 766 1448 955"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 976 1448 1564"> <thead> <tr> <th data-bbox="850 976 1224 1060">code</th> <th data-bbox="1224 976 1448 1060">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 1060 1224 1249">ServiceAccountNotFound</td> <td data-bbox="1224 1060 1448 1249">Service account was not found.</td> </tr> <tr> <td data-bbox="850 1249 1224 1564">ServiceAccountExpired</td> <td data-bbox="1224 1249 1448 1564">Service account is expired. Could not enable the expired service.</td> </tr> </tbody> </table>	code	message	ServiceAccountNotFound	Service account was not found.	ServiceAccountExpired	Service account is expired. Could not enable the expired service.
code	message							
ServiceAccountNotFound	Service account was not found.							
ServiceAccountExpired	Service account is expired. Could not enable the expired service.							

Status Code	Description	Return JSON Payload				
409	The service account is not ready to be enabled and is still being processed by some regions.	<pre data-bbox="850 222 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 432 1448 821"> <thead> <tr> <th data-bbox="850 432 1227 516">code</th> <th data-bbox="1227 432 1448 516">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 516 1227 821">ServiceAccountNotReady</td> <td data-bbox="1227 516 1448 821">The service account is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	ServiceAccountNotReady	The service account is still being processed by some regions.
code	message					
ServiceAccountNotReady	The service account is still being processed by some regions.					
500	The server encountered an internal error.	<pre data-bbox="850 869 1448 1056"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 1079 1448 1354"> <thead> <tr> <th data-bbox="850 1079 1088 1163">code</th> <th data-bbox="1088 1079 1448 1163">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 1163 1088 1354">InternalServerError</td> <td data-bbox="1088 1163 1448 1354">The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	InternalServerError	The server encountered an internal error. Please retry the request.
code	message					
InternalServerError	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre data-bbox="850 1421 1448 1608"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 1631 1448 1940"> <thead> <tr> <th data-bbox="850 1631 1140 1715">code</th> <th data-bbox="1140 1631 1448 1715">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 1715 1140 1940">ServiceNotReady</td> <td data-bbox="1140 1715 1448 1940">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					



# Disabling a Service Account

Disabling a service account prevents you from using the secret key to authenticate.

## Request

Performing a **Disable** operation removes the service account from the user's account.

```
DELETE /service-accounts/{serviceAccountId}/enabled
```

## Parameters

Name	In	Type	Required	Description
serviceAccountId	path	string	true	Numeric ID of the service account to disable.

## Code samples

### Go

```
package main

import (
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "text/plain",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    req, err := http.NewRequest("DELETE", "https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled", nil)
    req.Header = headers
    client := &http.Client{}
```

```
resp, err := client.Do(req)
// ... }
```

## Java

```
import java.net.HttpURLConnection;
import java.net.URL;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Main {
    public static void main(String[] args) throws Exception {
        URL obj = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled");
        HttpURLConnection con = (HttpURLConnection) obj.openConnection();
        con.setRequestMethod("DELETE");
        int responseCode = con.getResponseCode();
        BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
        String inputLine;
        StringBuffer response = new StringBuffer();
        while ((inputLine = in.readLine()) != null) { response.append(inputLine); } in.close(); System.out.println(response.toString()); } }
```

## JavaScript

```
const headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', {
  method: 'DELETE', headers: headers }) .then(res => res.text()) .then(body => console.log(body));
```

## Python

```
import requests

headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
}

r = requests.delete('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', headers=headers)

print(r.text)
```

# Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Accept' => 'text/plain',
  'Authorization' => 'Bearer {access-token}'
}

result = RestClient.delete 'https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}/enabled', params: {},
headers: headers

puts result
```

# Responses

Status Code	Description	Return JSON Payload						
200	OK  Service Account disabled successfully.	-						
400	It is an invalid request or invalid service account information.	<div style="border: 1px solid #ccc; padding: 10px; margin-bottom: 10px;"><pre>{   "code": "string",   "message": "string" }</pre></div> <table border="1"><thead><tr><th>code</th><th>message</th></tr></thead><tbody><tr><td>ExpiredToken</td><td>Token expired.</td></tr><tr><td>InvalidToken</td><td>Token is not valid.</td></tr></tbody></table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	message							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							



Status Code	Description	Return JSON Payload						
403	<p>Forbidden</p> <p>The account has no services enabled.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							
404	<p>The service account to disable was not found or expired.</p>	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ServiceAccountNotFound</td> <td>A service account was not found.</td> </tr> <tr> <td>ServiceAccountExpired</td> <td>Service account is expired. Could not disable the expired service.</td> </tr> </tbody> </table>	code	message	ServiceAccountNotFound	A service account was not found.	ServiceAccountExpired	Service account is expired. Could not disable the expired service.
code	message							
ServiceAccountNotFound	A service account was not found.							
ServiceAccountExpired	Service account is expired. Could not disable the expired service.							

Status Code	Description	Return JSON Payload				
409	The service account is not ready to be disabled and is still being processed by some regions.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceAccountNotReady</code></td> <td>The service account is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.
code	message					
<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.					
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>InternalServerError</code></td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.
code	message					
<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceNotReady</code></td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.
code	message					
<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.					

# Deleting a Service Account by ID

Deleting a service account permanently prevents you from using the secret and access key to authenticate.

## Request

The Delete operation removes the existing service account from the database.

```
DELETE /service-accounts/{serviceAccountId}
```

## Parameters

Name	In	Type	Required	Description
serviceAccountId	path	string	true	Numeric ID of the service account to delete.

## Code samples

### Go

```
package main

import (
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "text/plain",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    req, err := http.NewRequest("DELETE", "https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}", nil)
    req.Header = headers
    client := &http.Client{}
```

```
resp, err := client.Do(req)
// ... }
```

## Java

```
URL obj = new URL("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}");
URLConnection con = (URLConnection) obj.openConnection();
con.setRequestMethod("DELETE");
int responseCode = con.getResponseCode();
BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
String inputLine;
StringBuffer response = new StringBuffer();
while ((inputLine = in.readLine()) != null) { response.append(inputLine); }
in.close();
System.out.println(response.toString());
```

## JavaScript

```
const headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
};
fetch("https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}", {
  method: 'DELETE',
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });
```

## Python

```
import requests
headers = {
  'Accept': 'text/plain',
  'Authorization': 'Bearer {access-token}'
}
r = requests.delete('https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', headers = headers)
print(r.json())
```

## Ruby

```
require 'rest-client'
require 'json'
headers = {
  'Accept' => 'text/plain', 'Authorization' => 'Bearer {access-token}'
```

```

}
result = RestClient.delete 'https://api.lyvecloud.seagate.com/v2/service-accounts/{ServiceAccountId}', params: {}, headers:
headers
p JSON.parse(result)

```

## Responses

Status Code	Description	Return JSON Payload						
200	The delete operation is successful.	Service account deleted successfully.						
400	Bad request  The token is invalid or expired.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>ExpiredToken</td> <td>Token expired.</td> </tr> <tr> <td>InvalidToken</td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	message	ExpiredToken	Token expired.	InvalidToken	Token is not valid.
code	message							
ExpiredToken	Token expired.							
InvalidToken	Token is not valid.							
403	Forbidden  The account has no services enabled.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td>NoServiceAvailable</td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.		
code	message							
NoServiceAvailable	The account has no services enabled for it.							

Status Code	Description	Return JSON Payload				
404	The service account to be deleted was not found.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceAccountNotFound</code></td> <td>A service account was not found.</td> </tr> </tbody> </table>	code	message	<code>ServiceAccountNotFound</code>	A service account was not found.
code	message					
<code>ServiceAccountNotFound</code>	A service account was not found.					
409	The service account is not ready to be deleted and is still being processed by some regions.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceAccountNotReady</code></td> <td>The service account is still being processed by some regions.</td> </tr> </tbody> </table>	code	message	<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.
code	message					
<code>ServiceAccountNotReady</code>	The service account is still being processed by some regions.					
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>InternalServerError</code></td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.
code	message					
<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.					

Status Code	Description	Return JSON Payload				
503	Service Unavailable	<pre data-bbox="850 222 1448 411"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="850 432 1448 760"> <thead> <tr> <th data-bbox="850 432 1140 516">code</th> <th data-bbox="1140 432 1448 516">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="850 516 1140 760">ServiceNotReady</td> <td data-bbox="1140 516 1448 760">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					

# Usage

This API is used to get historical storage usage of the account for the current month or specified time range.

## Getting historical storage usage by month

The monthly average usage is calculated by averaging the daily average usage of all the days in that month. The monthly average usage is used to calculate the monthly cost at the end of the month.



### Note

- Based on the time range, it returns zero (0) if the data is unavailable.
- The data is displayed for a maximum interval of six months.

## Request

The GET operation provides the account's monthly usage for a specified duration.

```
GET /usage/monthly
```

## Parameters

Name	In	Type	Required	Description
fromMonth	query	Integer	true	Specifies the start month, inclusive, from which to retrieve usage data.  Where, zero corresponds to January and 11 corresponds to December.
fromYear	query	Integer	true	Specifies the start year, inclusive, from which to retrieve usage data.



Name	In	Type	Required	Description
toMonth	query	Integer	true	Specifies the end month, inclusive, from which to retrieve usage data.  Where, zero corresponds to January and 11 corresponds to December.
toYear	query	Integer	true	Specifies the end year, inclusive, from which to retrieve usage data.

## Code samples

### Go

```
package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    data := bytes.NewBuffer([]byte{jsonReq})
    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/usage/monthly/", data) req.Header = headers
    client := &http.Client{} resp, err := client.Do(req) // ... }
```

### Java

```
try {
    URL obj = new URL("https://api.lyvecloud.seagate.com/v2/usage/monthly/?from%2Dmonth=11&from%2Dyear=2020&to%2Dmonth=11&to%2Dyear=2020");
    HttpURLConnection con = (HttpURLConnection) obj.openConnection();
    con.setRequestMethod("GET");
    int responseCode = con.getResponseCode();
    BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
    String inputLine;
    StringBuffer response = new StringBuffer();
    while ((inputLine = in.readLine()) != null) {
        response.append(inputLine);
    }
}
```

```

    }
    in.close();
    System.out.println(response.toString());
} catch (IOException e) { e.printStackTrace(); }

```

## JavaScript

```

const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};
fetch('https://api.lyvecloud.seagate.com/v2/usage/monthly/?from%2Dmonth=11&from%2Dyear=2020&to%2Dmonth=11&to%2Dyear=2020', {
  method: 'GET',
  headers: headers
})
.then(function(res) {
  return res.json();
}).then(function(body) { console.log(body); });

```

## Python

```

import requests
headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}
r = requests.get('https://api.lyvecloud.seagate.com/v2/usage/monthly/', params = {
  'from-month': '11',
  'from-year': '2020',
  'to-month': '11',
  'to-year': '2020'}, headers = headers) print(r.json())

```

## Ruby

```

require 'rest-client'
require 'json'
headers = {
  'Accept' => 'application/json', 'Authorization' => 'Bearer {access-token}'
}
result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/usage/monthly/', params: {
  'from-month' => '[monthNumber](#schemamonthnumber)',
  'from-year' => '[yearNumber](#schemayearnumber)',
  'to-month' => '[monthNumber](#schemamonthnumber)',
  'to-year' => '[yearNumber](#schemayearnumber)'
}, headers: headers p JSON.parse(result)

```

# Responses

Status Code	Description	Returned JSON Payload								
200	<p>OK</p> <p>Successfully returns the data usage. Each item in return corresponds to one month.</p> <p>If the account is a master account, the response includes <code>usageByBucket</code> returning data usage for every bucket in the master account, and it also includes <code>usageBySubAccount</code>, returning data usage for its available sub-accounts.</p> <p>If the account is a sub-account, the response includes <code>usageByBucket</code>, returning data usage for every bucket in the sub-account. However, <code>usageBySubAccount</code>, is not returned, as there are no sub-accounts associated to another sub account.</p>	<pre> {   "usageByBucket": [     {       "year": 0,       "month": 0,       "totalUsageGB": 0,       "buckets": [         {           "name": "string",           "usageGB": 0         }       ]     }   ],   "usageBySubAccount": [     {       "year": 0,       "month": 0,       "totalUsageGB": 0,       "subAccounts": [         {           "subAccountName": "subname1",           "subAccountId": "string",           "usageGB": 0,           "createTime": "2022-10-16T14:02:20.319Z"         }       ]     }   ] } </pre> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>year</code></td> <td>Usage of the account for selected year.</td> </tr> <tr> <td><code>month</code></td> <td>Usage of the account for selected month.</td> </tr> <tr> <td><code>name</code></td> <td>Name of the bucket.</td> </tr> </tbody> </table>	Field	Description	<code>year</code>	Usage of the account for selected year.	<code>month</code>	Usage of the account for selected month.	<code>name</code>	Name of the bucket.
Field	Description									
<code>year</code>	Usage of the account for selected year.									
<code>month</code>	Usage of the account for selected month.									
<code>name</code>	Name of the bucket.									

Status Code	Description	Field Returned JSON Payload														
		<table border="1"> <thead> <tr> <th data-bbox="786 195 1078 279">Field</th> <th data-bbox="1078 195 1489 279">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 279 1078 394">usageGB</td> <td data-bbox="1078 279 1489 394">Bucket consumption in GB.</td> </tr> <tr> <td data-bbox="786 394 1078 510">totalUsageGB</td> <td data-bbox="1078 394 1489 510">Total storage consumption in GB.</td> </tr> <tr> <td data-bbox="786 510 1078 594">buckets</td> <td data-bbox="1078 510 1489 594">Bucket consumption.</td> </tr> <tr> <td data-bbox="786 594 1078 720">subAccounts</td> <td data-bbox="1078 594 1489 720">Details of sub-account.</td> </tr> <tr> <td data-bbox="786 720 1078 835">subAccountName</td> <td data-bbox="1078 720 1489 835">Name of the sub-account.</td> </tr> <tr> <td data-bbox="786 835 1078 856">subAccountId</td> <td data-bbox="1078 835 1489 856">Unique identifier of a sub-account.</td> </tr> </tbody> </table>	Field	Description	usageGB	Bucket consumption in GB.	totalUsageGB	Total storage consumption in GB.	buckets	Bucket consumption.	subAccounts	Details of sub-account.	subAccountName	Name of the sub-account.	subAccountId	Unique identifier of a sub-account.
Field	Description															
usageGB	Bucket consumption in GB.															
totalUsageGB	Total storage consumption in GB.															
buckets	Bucket consumption.															
subAccounts	Details of sub-account.															
subAccountName	Name of the sub-account.															
subAccountId	Unique identifier of a sub-account.															
400	Bad Request	<pre data-bbox="786 884 1448 1066"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 1094 1448 1415"> <thead> <tr> <th data-bbox="786 1094 1101 1171">code</th> <th data-bbox="1101 1094 1448 1171">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 1171 1101 1255">TokenExpired</td> <td data-bbox="1101 1171 1448 1255">Token expired.</td> </tr> <tr> <td data-bbox="786 1255 1101 1339">InvalidToken</td> <td data-bbox="1101 1255 1448 1339">Token is not valid.</td> </tr> <tr> <td data-bbox="786 1339 1101 1415">InvalidTimeRange</td> <td data-bbox="1101 1339 1448 1415">Invalid time range.</td> </tr> </tbody> </table>	code	message	TokenExpired	Token expired.	InvalidToken	Token is not valid.	InvalidTimeRange	Invalid time range.						
code	message															
TokenExpired	Token expired.															
InvalidToken	Token is not valid.															
InvalidTimeRange	Invalid time range.															
403	Forbidden The account has no services enabled.	<pre data-bbox="786 1467 1448 1650"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="786 1677 1448 1913"> <thead> <tr> <th data-bbox="786 1677 1101 1755">code</th> <th data-bbox="1101 1677 1448 1755">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="786 1755 1101 1913">NoServiceAvailable</td> <td data-bbox="1101 1755 1448 1913">The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	NoServiceAvailable	The account has no services enabled for it.										
code	message															
NoServiceAvailable	The account has no services enabled for it.															

Status Code	Description	Returned JSON Payload				
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>InternalServerError</code></td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.
code	message					
<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.					
503	Service Unavailable	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ServiceNotReady</code></td> <td>The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.
code	message					
<code>ServiceNotReady</code>	The server is not ready to handle the request. Please retry the request later.					

## Getting current month storage usage

This API returns the current month's storage usage and lists all buckets in the account and their calculated total usage size to the current date. In addition, you get the total number of buckets currently in the account and the total calculated usage.



**Note**—Based on the time range, it returns zero (0) if the data is unavailable.

## Request

The GET operation provides usage for the current month to date.

## Code samples

### Go

```

package main

import (
    "bytes"
    "net/http"
)

func main() {
    headers := map[string][]string{
        "Accept": []string{
            "application/json",
        },
        "Authorization": []string{
            "Bearer {access-token}",
        },
    }
    data := bytes.NewBuffer([]byte{jsonReq})
    req, err := http.NewRequest("GET", "https://api.lyvecloud.seagate.com/v2/usage/current", data) req.Header = headers c
    client := &http.Client{} resp, err := client.Do(req) // ... }

```

### Java

```

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class Main {
    public static void main(String[] args) {
        try {
            URL obj = new URL("https://api.lyvecloud.seagate.com/v2/usage/current");
            HttpURLConnection con = (HttpURLConnection) obj.openConnection();
            con.setRequestMethod("GET");
            int responseCode = con.getResponseCode();
            BufferedReader in = new BufferedReader(new InputStreamReader(con.getInputStream()));
            String inputLine;
            StringBuffer response = new StringBuffer();
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
            in.close();
            System.out.println(response.toString());
        } catch (Exception e) { e.printStackTrace(); } }

```

## JavaScript

```
const headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
};

fetch('https://api.lyvecloud.seagate.com/v2/usage/current', {
  method: 'GET',
  headers: headers
})
.then(function(res) {
  if (!res.ok) {
    throw new Error(`HTTP error! status: ${res.status}`);
  }
  return res.json();
})
.then(function(body) {
  console.log(body);
})
.catch(function(error) { console.log('There was a problem with the fetch operation: ' + error.message); });
```

## Python

```
import requests

headers = {
  'Accept': 'application/json',
  'Authorization': 'Bearer {access-token}'
}

try:
  r = requests.get("https://api.lyvecloud.seagate.com/v2/usage/current", headers=headers)
  r.raise_for_status()
  print(r.json())
except requests.exceptions.HTTPError as err:
  print(f"HTTP error occurred: {err}")
except Exception as err:
  print(f"An error occurred: {err}")
```

## Ruby

```
require 'rest-client'
require 'json'

headers = {
  'Accept' => 'application/json',
```

```

'Authorization' => 'Bearer {access-token}'
}

begin
  result = RestClient.get 'https://api.lyvecloud.seagate.com/v2/usage/current', params: {}, headers: headers
  p JSON.parse(result)
rescue RestClient::ExceptionWithResponse => e
  puts "An error occurred: #{e.message}"
end

```

## Responses

Status Code	Description	Returned JSON payload				
200	<p>OK</p> <p>Successfully return usage data.</p> <p>If the account is a master account, the response includes <code>usageByBucket</code>, returning data usage for every bucket in the master account, and it also includes <code>usageBySubAccount</code>, returning data usage for its available sub-accounts.</p> <p>If the account is sub-account, the response includes <code>usageByBucket</code>, returning data usage for every bucket in the sub-account. However, <code>usageBySubAccount</code> is not returned, as there are no sub-accounts associated to another sub-account</p>	<pre> {   "usageByBucket": {     "numBuckets": 0,     "totalUsageGB": 0,     "buckets": [       {         "name": "string",         "usageGB": 0       }     ]   },   "usageBySubAccount": {     "totalUsageGB": 0,     "subAccounts": [       {         "subAccountName": "subname1",         "subAccountId": "string",         "usageGB": 0,         "users": 0,         "serviceAccounts": 0,         "buckets": 0,         "trial": 0,         "createTime": "2022-10-16T14:06:51.494Z"       }     ]   } } </pre> <table border="1"> <thead> <tr> <th>Field</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td><code>numBuckets</code></td> <td>Number of buckets.</td> </tr> </tbody> </table>	Field	Description	<code>numBuckets</code>	Number of buckets.
Field	Description					
<code>numBuckets</code>	Number of buckets.					



Status Code	Description	Field	Description Returned JSON payload
		totalUsageGB	Total storage consumption in GB.
		buckets	Bucket consumption.
		name	Bucket name.
		usageGB	Bucket consumption in GB
		totalUsageGB	Total usage of the master account, including the sub-accounts.
		subAccounts	Details of sub-account.
		subAccountName	Name of the sub-account.
		subAccountId	Unique identifier of a sub-account.
		usageGB	Usage of the account in GB.
		users	Lists the total number of users for each sub-account.
		serviceAccounts	Lists the number of service accounts for each sub-account.
		buckets	Total number of buckets created in the Sub-account.
		trial	Remaining number of days for the trial to expire. If the trial is not provisioned for the account, then the trial is -1.

Status Code	Description	Field	Description						
		Returned JSON payload							
		<code>createTime</code>	Date and time when the Sub-account is created.						
400	The token is not valid or is expired.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>ExpiredToken</code></td> <td>Token expired.</td> </tr> <tr> <td><code>InvalidToken</code></td> <td>Token is not valid.</td> </tr> </tbody> </table>	code	message	<code>ExpiredToken</code>	Token expired.	<code>InvalidToken</code>	Token is not valid.	
code	message								
<code>ExpiredToken</code>	Token expired.								
<code>InvalidToken</code>	Token is not valid.								
403	Forbidden The account has no services enabled.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>NoServiceAvailable</code></td> <td>The account has no services enabled for it.</td> </tr> </tbody> </table>	code	message	<code>NoServiceAvailable</code>	The account has no services enabled for it.			
code	message								
<code>NoServiceAvailable</code>	The account has no services enabled for it.								
500	The server encountered an internal error.	<pre>{   "code": "string",   "message": "string" }</pre> <table border="1"> <thead> <tr> <th>code</th> <th>message</th> </tr> </thead> <tbody> <tr> <td><code>InternalServerError</code></td> <td>The server encountered an internal error. Please retry the request.</td> </tr> </tbody> </table>	code	message	<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.			
code	message								
<code>InternalServerError</code>	The server encountered an internal error. Please retry the request.								

Status Code	Description	Returned JSON payload				
503	Service Unavailable	<pre data-bbox="787 220 1448 409"> {   "code": "string",   "message": "string" } </pre> <table border="1" data-bbox="787 430 1448 709"> <thead> <tr> <th data-bbox="787 430 1075 514">code</th> <th data-bbox="1075 430 1448 514">message</th> </tr> </thead> <tbody> <tr> <td data-bbox="787 514 1075 709">ServiceNotReady</td> <td data-bbox="1075 514 1448 709">The server is not ready to handle the request. Please retry the request later.</td> </tr> </tbody> </table>	code	message	ServiceNotReady	The server is not ready to handle the request. Please retry the request later.
code	message					
ServiceNotReady	The server is not ready to handle the request. Please retry the request later.					