
Gluster Management Gateway – 1.0.0

ReST API Guide

Table of Contents

1. Preface	4
1.1. Disclaimer	4
1.2. Who Should Use This Book	4
1.3. Prerequisite	4
1.4. General Typographic Conventions.....	4
1.5. Gluster Welcomes Your Comments	5
2. Introduction.....	6
2.1. Resources	6
2.2. URIs	6
2.3. Representations	7
2.4. Requests	8
2.5. Parameters	8
2.6. Methods	8
2.7. Headers	9
2.8. Response Formats	9
3. HTTP Status Codes.....	10
4. Authentication	11
5. API Operations for User	12
5.1. User - PUT	12
6. API Operations for Clusters	13
6.1. Clusters - GET.....	13
6.2. Clusters - POST	13
6.3. Clusters - PUT.....	13
6.4. Cluster - DELETE.....	14
7. API Operations for Servers	15
7.1. Discovered Servers - GET	15
7.2. Discovered Server - GET	16
7.3. Servers - GET.....	17
7.4. Servers - GET.....	18
7.5. Servers - POST	18
7.6. Server - DELETE	18
7.7. Server Disk - PUT	19
8. API Operations for Volumes	20
8.1. Volumes - GET	20
8.2. Volume - GET	21

- 8.3. Volumes - POST 22
- 8.4. Volume - PUT 22
- 8.5. Volume - DELETE 23
- 8.6. Volume Options Info - GET 23
- 8.7. Volume Options - POST..... 24
- 8.8. Volume Options - PUT 24
- 8.9. Volume Logs - GET 24
- 8.10. Volume Logs Download - GET 25
- 9. API Operations for Bricks 26
 - 9.1. Bricks - POST 26
 - 9.2. Brick - PUT 26
 - 9.3. Brick - DELETE 27
- 10. API Operations for Tasks 28
 - 10.1. Tasks - GET 28
 - 10.2. Task - GET 28
 - 10.3. Task - PUT 29
 - 10.4. Task - DELETE 29

1. Preface

Gluster Management Gateway provides simple and powerful ReST APIs for managing your Gluster Storage Cluster. It also hosts the Gluster Management Console binaries, which can be run from a browser using the Java Webstart. Gluster Management Console uses the ReST APIs for performing the administrative tasks on the Gluster Storage Cluster.

GlusterFS is an open source, distributed file system capable of scaling to several petabytes and handling thousands of clients. GlusterFS clusters together storage building blocks over Infiniband RDMA or TCP/IP interconnect, aggregating disk and memory resources and managing data in a single global namespace. GlusterFS is based on a stackable user space design and can deliver exceptional performance for diverse workloads.

This document describes the ReST API for Gluster Management Gateway which is a component of Gluster Management Console. The Gluster Management Console offers an advanced, GUI-based interface that you can use to centrally manage your storage cluster.

1.1. Disclaimer

Gluster, Inc. has designated English as the official language for all of its product documentation and other documentation, as well as all our customer communications. All documentation prepared or delivered by Gluster will be written, interpreted and applied in English, and English is the official and controlling language for all our documents, agreements, instruments, notices, disclosures and communications, in any form, electronic or otherwise (collectively, the “Gluster Documents”).

Any customer, vendor, partner or other party who requires a translation of any of the Gluster Documents is responsible for preparing or obtaining such translation, including associated costs. However, regardless of any such translation, the English language version of any of the Gluster Documents prepared or delivered by Gluster shall control for any interpretation, enforcement, application or resolution.

1.2. Who Should Use This Book

This guide is intended to assist software developers who want to develop applications using Gluster Filesystem. It fully documents the ReST application programming interface (API) that allows developers to interact with their cloud storage environment.

1.3. Prerequisite

This document assumes that you are familiar with the Linux operating system, concepts of File System, GlusterFS concepts, ReST web services, and HTTP/1.1.

1.4. General Typographic Conventions

The following table lists the formatting conventions that are used in this guide to make it easier for you to recognize and use specific types of information.

Convention	Description	Example
Courier Text	Commands formatted as courier	<code>gluster volume start volname</code>

Convention	Description	Example
	indicate shell commands.	
<i>Italicized Text</i>	Within a command, italicized text represents variables, which must be substituted with specific values.	<code>gluster volume start <i>volname</i></code>
Square Brackets	Within a command, optional parameters are shown in square brackets.	<code>gluster volume start volname [force]</code>
Curly Brackets	Within a command, alternative parameters are grouped within curly brackets and separated by the vertical OR bar.	<code>gluster volume { start stop delete } volname</code>

1.5. Gluster Welcomes Your Comments

Gluster welcomes your comments and suggestions on the quality and usefulness of its documentation. If you find any errors or have any other suggestions, write to us at docfeedback@gluster.com for clarification by providing the chapter, section, and page number, if available.

Gluster offers a range of resources related to Gluster software:

- Discuss technical problems and solutions on the Discussion Forum (<http://community.gluster.org>)
- Get hands-on step-by-step tutorials (http://www.gluster.com/community/documentation/index.php/Main_Page)
- Reach Support (<http://www.gluster.com/services/>)

2. Introduction

Gluster Management Console provides a Representational State Transfer (ReST) Application Programming Interface (API). The API provides software developers and system administrators with control over their GlusterFS environment outside of the standard web interface. The ReST API is useful for developers and administrators who aim to integrate the functionality of GlusterFS with custom scripts or client-side applications that are able to send Hypertext Transfer Protocol (HTTP) requests and interpret responses. The benefit of the ReST API is that it is cross-platform and any system that accesses HTTP fulfils the basic requirements to access the ReST API.

This guide acts as a reference to Gluster Management Gateway ReST API. Explaining Gluster Management Gateway ReST API requires an examination of ReST architectural style and ReSTful web services in the context of a virtualized cloud environments. The API supports XML and JSON. The following sections provide an overview of ReST architecture and explanations of the key ReST terms.

NOTE: The Gluster Management Gateway works on SSL certificate. The product is shipped with a signed SSL certificate. Users must procure their own certificate and install it.

- [Resources](#)
- [URIs](#)
- [Representations](#)
- [Requests](#)
- [Methods](#)
- [Headers](#)
- [Response Formats](#)

2.1. Resources

A ReST API focuses primarily on **resources** of a specific service. A resource is a key abstraction of information and in the context of virtualization represents an individual component, capability or data set in the virtualized cloud infrastructure.

2.2. URIs

A user performs a request on any part of the API model to control the virtualization environment. For example, access to all resources in a collection requires only a request to the collection itself. Access to a specific resource requires a request to the resource within the context of its collection. An API user targets these requests to a specific location called a **Uniform Resource Identifier (URI)**.

The ReST API combines our virtualization model with the use of HTTP to construct a series of **Uniform Resource Identifiers (URIs)** for collections and resources. This means an API user accesses collections and resources in a manner similar to loading a webpage on the internet.

For example, an API user accesses a collection group called `clusters` from a ReST API installed at `www.example.com` with the following URI:

```
http://www.example.com/glustermg/1.0.0/clusters
```

Access to a resource called *mycluster* within clusters requires the following URI:

```
http://www.example.com/glustermg/1.0.0/clusters/mycluster
```

2.3. Representations

Access to a resource URI in a ReSTful web service does not mean direct access to the resource itself but to an abstraction of the resource. This abstraction is called a representation. The API constructs representations using XML and JSON formats but this documentation focuses primarily on XML representations.

When working with a specific resource, the API constructs a representation using elements. When an API user accesses the following URL:

http://www.example.com/glustermg/1.0.0/clusters/mycluster/volumes/myvolume, an XML resource representation structures elements in the following way:

```
<volume>
  <name>myvolume</name>
  <volumeType>DISTRIBUTED_MIRROR</volumeType>
  <transportType>ETHERNET</transportType>
  <status>ONLINE</status>
  ...
</volume>
```

A virtual machine resource contains elements such as machine type, operational status, and memory in bytes, CPUs and creation time. These elements form a basic XML structure for a virtual machine resource representation:

```
<server>
  <name>glusterserver1</name>
  <uuid>uuid</uuid>
  <status>ONLINE</Status>
  <NumOfCPUs>4</numOfCPUs>
  <cpuUsage>72.63</cpuUsage>
  <totalMemory>8192.00</totalMemory>
  <memoryInUse>1384.92</memoryInUse>
  <networkInterfaces>
    <networkInterface>
      <name>eth0</name>
      <hwAddr>0a:00:27:00:00:00</ hwAddr>
      <speed>speed</speed>
      <model>ETHERNET</model>
      <ipAddress>192.168.1.123</ipAddress>
      <netMask>255.255.255.0</netmask>
      <defaultGateway>192.168.1.1</ defaultGateway>
    </networkInterface>
  </networkInterfaces>
  <disks>
    <disk>
      <name>sdb</name>
      <description>ATA ST9320423AS</description>
      <uuid>06da4fde-01fe-4c7a-9918-80dc86695fea</uuid>
      <status>INITIALIZED</status>
      <type>DATA</type>
      <interface>pci</interface>
      <fsType>ext3</fsType>
      <mountPoint>/export/sdb</mountPoint>
      <space>2097152.00</space>
      <spaceInUse>281617.72</spaceInUse>
    </disk>
  </disks>
</server>
```

```

        </disks>
    </server>

```

This example shows how the resource representation portrays complex elements, such as network interfaces and disk information. Complex elements contain sub-elements to depict multiple properties of a single element. This demonstrates how ReST representations use XML to depict very specific aspects of resources and their elements.

2.4. Requests

The ReST API uses HTTP as a protocol to transfer a request for a resource representation from a client to the server:

```

GET /glustermg/1.0.0/clusters/mycluster/volumes/myvolume/HTTP/1
Host: www.example.com
Accept: application/xml

```

This example follows the standard HTTP request message format including:

- A **Request-Line**, which requires:
 - a **method** e.g. GET;
 - a **URI** e.g. /glustermg/1.0.0/clusters/mycluster/volumes/myvolume ; and
 - a **HTTP version** e.g. HTTP/1.1.
- A **Header** with fields to define parameters that process a request e.g. Host: and Accept:; and
- An optional **Message Entity** depending on the request **method**.

Requests from a client to a server contain all necessary information to process the request without depending on any previous request or stored context on the server, which makes the ReST API a **stateless** communication.

2.5. Parameters

QUERY parameters are the most common type of parameter that is appended to the path of the URL when submitting a request such as GET and DELETE. POST and PUT are form parameters.

2.6. Methods

The benefit of using HTTP to communicate with the ReST API is the ability to use a **method** to access Gluster cloud resources. A method defines the type of request to a resource. The ReST API uses four HTTP methods:

Method	Description
GET	Retrieve a resource or collection representation
POST	Create a resource
PUT	Update a resource
DELETE	Remove a resource

The default method is `GET` but an API user has a choice of any of the four methods to access and control resources in their clustered cloud storage environment.

2.7. Headers

The HTTP request contains a **header** to define HTTP parameters. The header uses specific **header fields** to define these parameters. The table below lists the two header fields which is required in the context of the ReST API:

Header	Description
Host	The target host of the URI i.e. the location of the cloud environment and ReST API
Accept	The accepted format for the representation. This documentation uses the <code>application/xml</code> to define the representation structure as XML format.

2.8. Response Formats

Gluster Management Gateway API supports two different output types: XML and JSON. To instruct the API calls to return the desired format, the client can set appropriate value (`application/xml` or `application/json`) in the HTTP header “Accept”. When no format is specified, it will default to XML.

3.HTTP Status Codes

This section provides descriptions of the HTTP success and error status codes that Gluster Management Gateway uses.

Note: The HTTP codes listed below are compatible with CDMI specifications.

Table 1 below lists the HTTP success status codes.

Table 1. *HTTP Success Status Codes*

Code	HTTP Name	Description
200	OK	Resource Retrieved Successfully (All GET)
201	Created	Resource Created Successfully (All POST)
202	Accepted	Long Running Operation Accepted for Processing (Few PUT)
204	No Content	Operation Successful, No Data (Most PUT, All DELETE)

Table 2 below lists the HTTP error status codes.

Table 2. *HTTP Error Status Codes*

Code	HTTP Name	Description
400	Bad Request	Missing or invalid request contents
401	Unauthorized	Invalid authentication/authorization credentials
403	Forbidden	This user is not allowed to perform this request (will not be implemented in v1)
404	Not Found	Requested resource not found
405	Method not Allowed	Requested HTTP verb not allowed on this resource
406	Not Acceptable	No content type can be produced at this URI that matches the request
500	Internal Server Error	An unexpected error during processing request

4. Authentication

Authentication is the process of proving your identity to the system. All API calls **MUST** occur over SSL (HTTPS) and all API calls must be authenticated to a user. Once the user is authenticated, the user will have access to all accounts it belongs to (provided that the account has `api_access` enabled).

We support the standard HTTP Authentication: **Basic** method.

5. API Operations for User

This section describes the user operations of the Gluster Management Gateway API.

- [User - PUT](#)

5.1. User - PUT

Description: Use this operation to change the user password by issuing a `PUT` request to the URI.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/users/{userName}`

Parameters:

- `oldPassword` - old password
- `newPassword` - new password

HTTP Response Code: 204

6. API Operations for Clusters

This section describes the cluster operations of the Gluster Management Gateway API. These APIs enable you to create, register or unregister a cluster and retrieve cluster list.

- [Clusters - GET](#)
- [Clusters - POST](#)
- [Clusters - PUT](#)
- [Cluster - DELETE](#)

6.1. Clusters - GET

Description: Use this operation to get a list of all your clusters registered in your cloud environment.

Resource URI: *https://server:port/glustermg/1.0.0/clusters*

Parameters: None

Response XML:

```
<clusters>
  <cluster>clusterName1</cluster>
  <cluster>clusterName2</cluster>
</clusters>
```

HTTP Response Code: 200

6.2. Clusters - POST

Description: Use this operation to create a new cluster.

Resource URI: *https://server:port/glustermg/1.0.0/clusters/*

Parameters:

- *clusterName* - Name of the cluster to be created.

HTTP Response Code: 201

6.3. Clusters - PUT

Description: Use this operation to register an existing cluster.

Resource URI: *https://server:port/glustermg/1.0.0/clusters/*

Parameters:

- *clusterName* - Name of the cluster to be registered.
- *serverName* - One of the online servers of the cluster.

HTTP Response Code: 204

6.4. Cluster - DELETE

Description: Use this operation to unregister a cluster by issuing a `DELETE` request to its URI.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/{clusterName}`

HTTP Response Code: 204

7.API Operations for Servers

This section describes the discovered server and server operations of the Gluster Management Gateway API.

- [Discovered Servers - GET](#)
- [Servers - GET](#)
- [Servers - GET](#)
- [Servers - POST](#)
- [Server - DELETE](#)
- [Servers Disks - GET](#)
- [Server Disk - PUT](#)
- [Server Logs - GET](#)

7.1. Discovered Servers - GET

Description: Use this operation to list all the discovered Gluster servers available in your network. These servers run GlusterFS, but are not part of any storage cloud environment.

Resource URI: `https://server:port/glustermg/1.0.0/discoveredservers`

Parameters:

- `details` - value = true/false; optional; false by default

Response XML:

`details = true`

```
<servers>
  <server>
    <name>serverName</name>
    <numOfCPUs>numOfCPUs</numOfCPUs>
    <cpuUsage>currentCPUUsage</cpuUsage>
    <totalMemory>totalMemory</totalMemory>
    <memoryInUse>memoryInUse</memoryInUse>
  <networkInterfaces>
    <networkInterface>
      <name>name</name>
      <hwAddr>macAddress</hwAddr>
      <speed>speed</speed>
      <model>model</model>
      <ipAddress>ipAddress</ipAddress>
      <netMask>netMask</netMask>
      <defaultGateway/>defaultGateway</defaultGateway>
    </networkInterface>
    ...
  </networkInterfaces>
  <disks>
    <disk>...</disk>
```

```

        ...
    </disks>
</server>
...
</servers>

```

`details = false`

```

<servers>
  <server>serverName1</server>
  <server>serverName2</server>
  ...
</servers>

```

HTTP Response Code: 200

7.2. Discovered Server - GET

Description: Use this operation to get details of a given discovered server in your cluster.

Resource URI: `https://server:port/glustermg/1.0.0/discoveredservers/{serverName}`

Response XML:

```

<server>
  <name>serverName</name>
  <numOfCPUs>numOfCPUs</numOfCPUs>
  <cpuUsage>currentCPUUsage</cpuUsage>
  <totalMemory>totalMemory</totalMemory>
  <memoryInUse>memoryInUse</memoryInUse>
  <networkInterfaces>
    <networkInterface>
      <name>name</name>
      <hwAddr>macAddress</hwAddr>
      <speed>speed</speed>
      <model>model</model>
      <ipAddress>ipAddress</ipAddress>
      <netMask>netMask</netMask>
      <defaultGateway/>defaultGateway</defaultGateway>
    </networkInterface>
    ...
  </networkInterfaces>
  <disks>
    <disk>...</disk>
    ...
  </disks>
</server>

```

HTTP Response Code: 200

7.3. Servers - GET

Description: Use this operation to get a list of all servers in a cluster.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/{clusterName}/servers`

Parameters:

- `details` - value = true/false; optional; false by default
- `nextTo` - optional; first server in the response will be the one “next to” the given server name.
- `maxCount` - optional; maximum number of servers to be returned

Response XML:

`details = true`

```
<servers>
  <server>
    <name>serverName</name>
    <uuid>uuid</uuid>
    <status>status</status>
    <numOfCPUs>numOfCPUs</numOfCPUs>
    <cpuUsage>currentCPUUsage</cpuUsage>
    <totalMemory>totalMemory</totalMemory>
    <memoryInUse>memoryInUse</memoryInUse>
    <networkInterfaces>
      <networkInterface>
        <name>name</name>
        <hwAddr>macAddress</hwAddr>
        <speed>speed</speed>
        <model>model</model>
        <ipAddress>ipAddress</ipAddress>
        <netMask>netMask</netMask>
        <defaultGateway/>defaultGateway</defaultGateway>
      </networkInterface>
      ...
    </networkInterfaces>
    <disks>
      <disk>...</disk>
      ...
    </disks>
  </server>
  ...
</servers>
```

`details = false`

```
<servers>
  <server>serverName1</server>
  <server>serverName2</server>
  ...
</servers>
```

HTTP Response Code: 200

7.4. Servers - GET

Description: Use this operation to get details of particular server in a cluster.

Resource URI:

https://server:port/glustermg/1.0.0/clusters/{clusterName}/servers/{serverName}

Response XML:

```
<server>
  <name>serverName</name>
  <uuid>uuid</uuid>
  <status>status</status>
  <numOfCPUs>numOfCPUs</numOfCPUs>
  <cpuUsage>currentCPUUsage</cpuUsage>
  <totalMemory>totalMemory</totalMemory>
  <memoryInUse>memoryInUse</memoryInUse>
  <networkInterfaces>
    <networkInterface>
      <name>name</name>
      <hwAddr>macAddress</hwAddr>
      <speed>speed</speed>
      <model>model</model>
      <ipAddress>ipAddress</ipAddress>
      <netMask>netMask</netMask>
      <defaultGateway/>defaultGateway</defaultGateway>
    </networkInterface>
    ...
  </networkInterfaces>
  <disks>
    <disk>...</disk>
    ...
  </disks>
</server>
```

HTTP Response Code: 200

7.5. Servers - POST

Description: Use this operation to add a server to your cluster.

Resource URI: *https://server:port/glustermg/1.0.0/clusters/{clusterName}/servers*

Parameters:

- *serverName* - the hostname of server to be added to the cluster

HTTP Response Code: 201

7.6. Server - DELETE

Description: Use this operation to delete a server by issuing a DELETE request to its URI.

Resource URI:

https://server:port/glustermg/1.0.0/clusters/{clusterName}/servers/{serverName}

HTTP Response Code: 204

7.7. Server Disk - PUT

Description: Use this operation to initialize a server disk by issuing a `PUT` request to its URI.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/servers/{serverName}/disks/{diskName}`

Parameters:

- `fsType` - Filesystem Type: `default`, `ext3`, `ext4`, `xfs`

Response XML: NA

- The URI of newly created task will be returned in the Resource URI “location”.

HTTP Response Code: 202

8.API Operations for Volumes

This section describes the volume operations of the Gluster Management Gateway API.

- [Volumes - GET](#)
- [Volume - GET](#)
- [Volumes - POST](#)
- [Volume - PUT](#)
- [Volume - DELETE](#)
- [Volume Options Info - GET](#)
- [Volume Options - POST](#)
- [Volume Options - PUT](#)
- [Volume Logs - GET](#)
- [Volume Logs Download - GET](#)

8.1. Volumes - GET

Description: Use this operation to list all volumes of your cluster.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes`

Parameters:

- `nextTo` - optional; first volume in the response will be the one “next to” the given volume name.
- `maxCount` - optional; maximum number of volumes to be returned.

Response XML:

```
<volumes>
  <volume>
    <name>volumeName</name>
    <volumeType>volumeType</volumeType>
    <transportType>transportType</transportType>
    <status>volumeStatus</status>
    <replicaCount>replicaCount</replicaCount>
    <stripeCount>stripeCount</stripeCount>
    <bricks>
      <brick> <serverName>serverName</serverName>
              <brickDirectory>brickDir</brickDirectory>
              <diskName>diskName</diskName>
              <status>brickStatus</status>
            </brick>
      ...
    </bricks>
    <accessProtocols>
      <accessProtocol>GLUSTERFS</accessProtocol>
    </accessProtocols>
  </volume>
</volumes>
```

```

        <accessProtocol>CIFS</accessProtocol>
        ...
    </accessProtocols>
    <cifsUsers>user1</cifsUsers>
    <cifsUsers>user2</cifsUsers>
    <options>
        <option>
            <key>optionKey</key>
            <value>optionValue</value>
        </option>
        ...
    </options>
</volume>
...
</volumes>

```

HTTP Response Code: 200

8.2. Volume - GET

Description: Use this operation to get details of a specified volume in your cluster.

Resource URI:

https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}

Response XML:

```

<volume>
  <name>volumeName</name>
  <volumeType>volumeType</volumeType>
  <transportType>transportType</transportType>
  <status>volumeStatus</status>
  <replicaCount>replicaCount</replicaCount>
  <stripeCount>stripeCount</stripeCount>
  <bricks>
    <brick>
      <serverName>serverName</serverName>
      <brickDirectory>brickDir</brickDirectory>
      <diskName>diskName</diskName>
      <status>brickStatus</status>
    </brick>
    ...
  </bricks>
  <accessProtocols>
    <accessProtocol>GLUSTERFS</accessProtocol>
    ...
  </accessProtocols>
  <options>
    <option>
      <key>optionKey</key>
      <value>optionValue</value>
    </option>
    ...
  </options>
</volume>

```

HTTP Response Code: 200

8.3. Volumes - POST

Description: Use this operation to create a new volume in your cluster.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}`

Parameters:

- *volumeName* - Name of the volume to be created.
- *volumeType* - type of the volume; valid values = `PLAIN_DISTRIBUTE/DISTRIBUTED_MIRROR/DISTRIBUTED_STRIPE`
- *transportType* - transport type of the volume; optional

Note: The only value supported in this version is `ETHERNET`.

- *replicaCount* - mandatory only if *volumeType* = `DISTRIBUTED_MIRROR`
- *stripeCount* - mandatory only if *volumeType* = `DISTRIBUTED_STRIPE`
- *bricks* - comma separated list of bricks viz. `server1:dir1,server2:dir2,...servern:dirn`
- *accessProtocols* - comma separated list of access protocols; such as GLUSTER, NFS, CIFS

Note: In case of CIFS:

- *cifsUsers* - value = `user1,user2...`
- *options* - comma separated list of key-value pairs; optional; value must not contain '='

HTTP Response Code: 201

8.4. Volume - PUT

Description: Use this operation to start, stop, or rebalance your volume in your cluster.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}`

Parameters:

- **Operation** - `start/stop/rebalanceStart/rebalanceStop/cifsConfig`
- *force* - values = `true/false`; optional - `false` by default; indicates if force option is to be used with the operation.
- In case of CIFS:
 - *enableCifs* - value = `true/false`
 - *cifsUsers* - value = `user1,user2...`
- In case of *rebalanceStart*:

- *fixLayout* - true/false; optional - false by default
- *migrateData* - true/false; optional - true by default
- *forcedDataMigrate* - true/false; optional - false by default.

HTTP Response Code:

- 202 - long running operation; such as rebalance-start

In case of *rebalanceStart*, the URI of newly created rebalance *Task* will be returned in the HTTP header “location”

- 204 - for other operations

8.5. Volume - DELETE

Description: Use this operation to delete a volume by issuing a *DELETE* request to its URI.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}`

Parameters:

- *deleteData* - value = true/false; optional; false by default.

Note: Setting this parameter to *true*, all data will be deleted from the volume. This can impact the performance of the system while the deletion is in progress, hence must be used with caution.

HTTP Response Code: 204

8.6. Volume Options Info - GET

Description: Use this operation to list information about all volume options.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/options`

Response XML:

```
<options>
  <option>
    <name>optionName</name>
    <description>description</description>
    <defaultValue>defaultValue</defaultValue>
  </option>
  ...
</options>
```

HTTP Response Code: 200

8.7. Volume Options - POST

Description: Use this operation to set options on a specified volume.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/options`

Parameters:

- `key` - option key
- `value` - option value

HTTP Response Code: 201

8.8. Volume Options - PUT

Description: Use this operation to reset options on a specified volume.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/options`

HTTP Response Code: 204

8.9. Volume Logs - GET

Description: Use this operation to fetch the volume logs for given criteria.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/logs`

Parameters:

- `brickName` - Fully qualified name of the brick from where logs are to be retrieved in the format `serverName:brickDirectory`. Choose `ALL` to retrieve logs from all bricks of the volume
- `severity` - Severity of the logs to be fetched. Valid values are `EMERGENCY`, `ALERT`, `CRITICAL`, `ERROR`, `WARNING`, `NOTICE`, `INFO`, `DEBUG`, `TRACE`
- `fromTimeStamp` - From timestamp in format `MM/dd/yyyy HH:mm:ss`
- `toTimeStamp` - To timestamp in format `MM/dd/yyyy HH:mm:ss`
- `lineCount` - Maximum number of log messages to be retrieved from each brick. E.g. If the value is 100, last 100 log messages from each brick are retrieved. Thus, if `brickName` = `ALL`, the response will contain a maximum of (100 * number of bricks) log messages.

Response XML:

```
<logMessages>
  <logMessage>
    <timeStamp>timestamp</timeStamp>
```



```

        <brick>serverName:brickDirectory</brick>
        <severity>severity</severity>
        <message>logMessage</message>
    </logMessage>
    ...
</logMessages>

```

HTTP Response Code: 200

8.10. Volume Logs Download - GET

Description: Use this operation to download log files of all bricks of the volume in a compressed format.

Resource URI:

https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/logs/download

Response XML:

- *.tar.gz* of log files from all bricks of the volume

HTTP Response Code: 200

9. API Operations for Bricks

This section describes the brick operations of the Gluster Management Gateway API.

- [Bricks - POST](#)
- [Brick - PUT](#)
- [Brick - DELETE](#)

9.1. Bricks - POST

Description: Use this operation to add bricks to the volume.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/bricks`

Parameters:

- *bricks* - comma separated list of bricks; such as *svr1:dir1, svr2:dir2, svr3:dir3*

HTTP Response Code: 201

9.2. Brick - PUT

Description: Use this operation to migrate a volume brick to a new one.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/bricks`

Parameters:

- *operation* - migrate-start/migrate-pause/migrate-abort/migrate-commit
- *source* - source brick that you want to migrate (*server:brickDirectory*)
- *target* - destination brick (*server:brickDirectory*)
- *force* - values = true/false; optional - false by default; indicates if force option is to be used with the operation.
- *autoCommit* = true/false; optional; false by default; applicable only when operation=migrate-start

HTTP Response Code:

- 202 - in case of long running operation; such as migrate-start

Note: In case of migrate-start, the URI of newly created brick migration *Task* will be returned in the HTTP header “location”

- 204 - for other operations

9.3. Brick - DELETE

Description: Use this operation to remove brick(s) from a volume by issuing a DELETE request to its URI.

Resource URI:

`https://server:port/glustermg/1.0.0/clusters/{clusterName}/volumes/{volumeName}/bricks`

Parameters:

- *bricks* - comma separated list of bricks; such as *svr1:dir1, svr2:dir2, svr3:dir3*
- *deleteData* - value = true/false; optional; false by default.

Note: If you set the parameter to *true*, it will physically delete all data of the brick. It can impact performance of the system while the deletion is in progress, and hence must be used with caution

HTTP Response Code: 204

10. API Operations for Tasks

This section describes tasks operations of the Gluster Management Gateway API.

- [Tasks - GET](#)
- [Task - GET](#)
- [Task - PUT](#)
- [Task - DELETE](#)

10.1. Tasks - GET

Description: Use this operation to get a list of all tasks running in the cluster.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/{clusterName}/tasks`

Response XML:

```
<tasks>
  <task>
    <id>taskId</id>
    <type>taskType</type>
    <reference>referenceEntityName</reference>
    <description>taskDescription</description>
    <pauseSupported>true/false</pauseSupported>
    <stopSupported>true/false</stopSupported>
    <commitSupported>>false</commitSupported>
    <status>
      <code>statusCode</code>
      <message>statusMessage</message>
      <percentageSupported>true/false</percentageSupported>
      <percentCompleted>%completed</percentCompleted>
    </status>
  </task>
  ...
</tasks>
```

HTTP Response Code: 200

10.2. Task - GET

Description: Use this operation to get the details of a specified task.

Resource URI: `https://server:port/glustermg/1.0.0/clusters/{clusterName}/tasks/{id}`

Response XML:

```
<task>
  <id>taskId<id>
  <type>taskType</type>
  <reference>referenceEntityName</reference>
  <description>taskDescription</description>
  <pauseSupported>true/false</pauseSupported>
  <stopSupported>true/false</stopSupported>
```

```

    <commitSupported>false</commitSupported>
    <status>
      <code>statusCode</code>
      <message>statusMessage</message>
      <percentageSupported>true/false</percentageSupported>
      <percentCompleted>%completed</percentCompleted>
    </status>
  </task>

```

HTTP Response Code: 200

10.3. Task - PUT

Description: Use this operation to pause or resume a task.

Resource URI: *https://server:port/glustermg/1.0.0/clusters/{clusterName}/tasks/{id}*

Parameters:

- operation - pause/resume

HTTP Response Code: 204

10.4. Task - DELETE

Description: Use this operation to abort a running task.

Resource URI: *https://server:port/glustermg/1.0.0/clusters/{clusterName}/tasks/{id}*

HTTP Response Code: 204