

Riverbed Cascade Gateway REST API v1.1

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

Overview

Overview

The documentation pages in this section describe the RESTful APIs included with Cascade Gateway products. It is assumed that the reader has practical knowledge of RESTful APIs, so the documentation does not go into detail about what REST is and how to use it. Instead the documentation focuses on what data can be accessed and how to access it.

The following information can be accessed via the API:

- Perform System operations
- Information about system users

Details about REST resources can be found in the **Resources** section. This overview continues with how to run reports and retrieve data from them.

Authentication

All REST requests must be authenticated. The **Authentication** section of the Common 1.0 API describes which authentication methods are presently supported. There are also examples that show how to use each of the different authentication methods.

Resources

Ping: Ping

Simple test of service availability.

```
GET https://{device}/api/gateway/1.1/ping
```

Authorization

This request requires authorization.

Response Body

On success, the server does not provide any body in the responses.

Users: List users

Get a list of user accounts.

```
GET https://{device}/api/gateway/1.1/users
```

Authorization

This request requires authorization.

Response Body

On success, the server returns a response body with the following structure:

```
JSON
```

```
[
  {
    "authentication_type": string,
    "authorization_type": string,
    "enabled": boolean,
    "first_name": string,
    "id": number,
    "last_access": number,
    "last_authentication": number,
    "last_login": number,
    "last_name": string,
    "login_timeout": number,
    "role": string,
    "username": string,
    "view_packet_details": boolean,
    "view_user_information": boolean
  }
]
```

Example:

```
[
  {
    "username": "admin",
    "last_authentication": 1352313328,
    "first_name": "Jonh",
    "last_name": "Smith",
    "authorization_type": "Local",
    "enabled": true,
    "view_user_information": true,
    "authentication_type": "Local",
    "role": "Administrator",
    "login_timeout": 900,
    "last_login": 1352313328,
    "last_access": 1352313328,
    "id": 123
  },
  {
    "username": "admin2",
    "last_authentication": 1352313328,
    "first_name": "Mark",
    "last_name": "Greg",
    "authorization_type": "Local",
    "enabled": true,
    "view_user_information": true,
    "authentication_type": "Local",
    "role": "Administrator",
    "login_timeout": 900,
    "last_login": 1352313328,
    "last_access": 1352313328,
    "id": 124
  }
]
```

Property Name	Type	Description	Notes
<i>users</i>	<array of <object>>	List of user accounts on the system.	
<i>users[user]</i>	<object>	User account.	
<i>users[user].authentication_type</i>	<string>	Type of authentication for the user, such as Local or RADIUS.	Values: Local, RADIUS
<i>users[user].authorization_type</i>	<string>	Type of authorization for the user, such as Local or RADIUS.	Values: Local, RADIUS
<i>users[user].enabled</i>	<boolean>	Boolean flag indicating if the user account is enabled.	
<i>users[user].first_name</i>	<string>	First name of the user.	
<i>users[user].id</i>	<number>	Numeric ID of the user that the system uses internally and in the API.	
<i>users[user].last_access</i>	<number>	Time of last access to the system. Unix time (epoch).	
<i>users[user].last_authentication</i>	<number>	Time of last authentication. Unix time (epoch).	
<i>users[user].last_login</i>	<number>	Time of last login. Unix time (epoch).	
<i>users[user].last_name</i>	<string>	Last name of the user.	
<i>users[user].login_timeout</i>	<number>	Timeout (in seconds) during which the user cannot log in to the system because of security policies.	
<i>users[user].role</i>	<string>	Role of the user. Defines permissions.	Values: Developer, Administrator, Operator, Monitor, Event_Viewer, Dashboard_Viewer
<i>users[user].username</i>	<string>	User name (short name) that identifies the user to the system, such as 'admin'.	

<code>users[user].view_packet_details</code>	<code><boolean></code>	Boolean flag indicating if the user has access to packet data.	Optional
<code>users[user].view_user_information</code>	<code><boolean></code>	Boolean flag indicating if the user has access to identity information, such as Active Directory information.	Optional

Users: Test RADIUS user

Test a RADIUS user.

```
GET https://{device}/api/gateway/1.1/users/radius/test_user?username={string}&password={string}
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes
<code>username</code>	<code><string></code>	RADIUS username.	
<code>password</code>	<code><string></code>	RADIUS username password.	

Response Body

On success, the server returns a response body with the following structure:

```
JSON
```

```

{
  "attributes": [
    {
      [attr]: string
    }
  ],
  "authenticated": boolean,
  "authorized": boolean,
  "details": string,
  "error_message": string,
  "permission": string,
  "permission_id": string,
  "role": string,
  "role_id": number,
  "server_ip": string
}

```

Example:

```

{
  "authenticated": true,
  "permission": "",
  "permission_id": "",
  "error_message": "",
  "role_id": 0,
  "role": "",
  "authorized": false,
  "server_ip": "10.38.8.112:1812",
  "attributes": [
    {
      "25": "operatorClass"
    },
    {
      "25": "monitorClass"
    },
    {
      "25": "eventviewerClass"
    },
    {
      "17164": "unMappedRole"
    },
    {
      "17164": "monitorCascade"
    },
    {
      "17164": "eventviewerCascade"
    },
    {
      "17164": "dashboardCascade"
    },
    {
      "25": "DBAccess"
    },
    {
      "25": "dashboardClass"
    },
    {
      "17164": "AbC10~!@#%$%^&*()_+{|[]:;<>?/.'z"
    },
    {
      "17164": "operatorCascade"
    },
    {
      "LOGIN_RADIUS_SERVER": "10.38.8.112:1812"
    },
    {
      "25": "adminClass1"
    },
    {
      "25": "unMappedClass"
    },
    {
      "25": "eventviewerClass"
    },
    {
      "17164": "adminCascade"
    }
  ],
  "details": "Using 10.38.8.112:1812 - Unable to match a role."
}

```

Property Name	Type	Description	Notes
test_server	<object>	RadiusTestUserResponse object.	

<code>test_server.attributes</code>	<code><array of <object>></code>	Radius Attributes List object.	
<code>test_server.attributes[attribute]</code>	<code><object></code>	RADIUS attributes map.	Optional
<code>test_server.attributes[attribute][attr]</code>	<code><string></code>	RADIUS attribute value.	
<code>test_server.authenticated</code>	<code><boolean></code>	Flag indication if the RADIUS user was authenticated.	
<code>test_server.authorized</code>	<code><boolean></code>	Flag indication if the RADIUS user was authorized (as Administrator, Monitor, etc).	
<code>test_server.details</code>	<code><string></code>	RADIUS user test details.	
<code>test_server.error_message</code>	<code><string></code>	Error message.	
<code>test_server.permission</code>	<code><string></code>	Matched permission name.	
<code>test_server.permission_id</code>	<code><string></code>	Matched permission ID.	
<code>test_server.role</code>	<code><string></code>	Matched role name.	
<code>test_server.role_id</code>	<code><number></code>	Matched role ID.	
<code>test_server.server_ip</code>	<code><string></code>	RADIUS Server IP Address.	

Users: Test RADIUS server

Test the connection to a RADIUS server.

```
GET https://{device}/api/gateway/1.1/users/radius/test_server
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes
<code>server</code>	<code><string></code>	RADIUS server identifier, example server=IP:PORT.	

Response Body

On success, the server returns a response body with the following structure:

JSON

```
{
  "message": string,
  "success": boolean
}
```

Example:

```
{
  "message": "RADIUS connection attempt succeeded",
  "success": true
}
```

Property Name	Type	Description	Notes
<code>test_server</code>	<code><object></code>	RadiusTestServerResponse object.	
<code>test_server.message</code>	<code><string></code>	Response message.	
<code>test_server.success</code>	<code><boolean></code>	Flag indication if the RADIUS server test was successful.	

Users: Re-authenticate user

Re-authenticate a user account. Requires basic authentication.

```
GET https://{device}/api/gateway/1.1/users/re_authenticate
```

Authorization

This request requires authorization.

Response Body

On success, the server does not provide any body in the responses.

Users: Get user

Get a user account by user ID.

```
GET https://{device}/api/gateway/1.1/users/{user_id}
```

Authorization

This request requires authorization.

Response Body

On success, the server returns a response body with the following structure:

JSON

```
{
  "authentication_type": string,
  "authorization_type": string,
  "enabled": boolean,
  "first_name": string,
  "id": number,
  "last_access": number,
  "last_authentication": number,
  "last_login": number,
  "last_name": string,
  "login_timeout": number,
  "role": string,
  "username": string,
  "view_packet_details": boolean,
  "view_user_information": boolean
}
```

Example:

```
{
  "username": "admin",
  "last_authentication": 1352313328,
  "first_name": "Jonh",
  "last_name": "Smith",
  "authorization_type": "Local",
  "enabled": true,
  "view_user_information": true,
  "authentication_type": "Local",
  "role": "Administrator",
  "login_timeout": 900,
  "last_login": 1352313328,
  "last_access": 1352313328,
  "id": 123
}
```

Property Name	Type	Description	Notes
<i>user</i>	<object>	User account.	
<i>user.authentication_type</i>	<string>	Type of authentication for the user, such as Local or RADIUS.	Values: Local, RADIUS
<i>user.authorization_type</i>	<string>	Type of authorization for the user, such as Local or RADIUS.	Values: Local, RADIUS
<i>user.enabled</i>	<boolean>	Boolean flag indicating if the user account is enabled.	
<i>user.first_name</i>	<string>	First name of the user.	
<i>user.id</i>	<number>	Numeric ID of the user that the system uses internally and in the API.	
<i>user.last_access</i>	<number>	Time of last access to the system. Unix time (epoch).	
<i>user.last_authentication</i>	<number>	Time of last authentication. Unix time (epoch).	
<i>user.last_login</i>	<number>	Time of last login. Unix time (epoch).	
<i>user.last_name</i>	<string>	Last name of the user.	
<i>user.login_timeout</i>	<number>	Timeout (in seconds) during which the user cannot log in to the system because of security policies.	
<i>user.role</i>	<string>	Role of the user. Defines permissions.	Values: Developer, Administrator, Operator, Monitor, Event_Viewer, Dashboard_Viewer
<i>user.username</i>	<string>	User name (short name) that identifies the user to the system, such as 'admin'.	
<i>user.view_packet_details</i>	<boolean>	Boolean flag indicating if the user has access to packet data.	Optional
<i>user.view_user_information</i>	<boolean>	Boolean flag indicating if the user has access to identity information, such as Active Directory information.	Optional

System: Stop all processes

Stop all system processes. The operation is asynchronous. Use "GET system/status" to poll for status. On Enterprise systems, stop system processes on all modules.

```
POST https://{device}/api/gateway/1.1/system/stop
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Kill all processes

Kill all system processes. The operation is asynchronous. Use "GET system/status" to poll for status. On Enterprise systems, kill system processes on all modules. Warning: this operation can result in data being corrupted.

```
POST https://{device}/api/gateway/1.1/system/kill
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Shutdown

Shutdown the system. The operation is asynchronous.

```
POST https://{device}/api/gateway/1.1/system/shutdown
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Get status of all processes

Get status of all system processes. On Enterprise systems, get system process statuses on all modules.

```
GET https://{device}/api/gateway/1.1/system/status
```

Authorization

This request requires authorization.

Response Body

On success, the server returns a response body with the following structure:

```
JSON
```



```
[
  {
    "module_ipaddr": string,
    "module_name": string,
    "process_id": string,
    "process_name": string,
    "status": string
  }
]
```

Example:

```
[
  {
    "process_id": "25096",
    "process_name": "memmonitor",
    "status": "Running"
  },
  {
    "process_name": "healthd",
    "status": "Stopped"
  },
  {
    "process_id": "25092",
    "process_name": "diskmon",
    "status": "Running"
  },
  {
    "process_id": "25123",
    "process_name": "dispatcher",
    "status": "Running"
  },
  {
    "process_name": "analyzer",
    "status": "Stopped"
  }
]
```

Property Name	Type	Description	Notes
<i>status</i>	<array of <object>>	SystemStatus object.	
<i>status[process]</i>	<object>	SystemProcess object.	
<i>status[process].module_ipaddr</i>	<string>	Module IP Address. Available on Enterprise systems only.	Optional
<i>status[process].module_name</i>	<string>	Module name. Available on Enterprise systems only.	Optional
<i>status[process].process_id</i>	<string>	Process ID.	Optional
<i>status[process].process_name</i>	<string>	Process name.	
<i>status[process].status</i>	<string>	Process status.	Values: Running, Stopped

System: Start all processes

Start all system processes. The operation is asynchronous. Use "GET system/status" to poll for status. On Enterprise systems, start system processes on all modules.

```
POST https://{device}/api/gateway/1.1/system/start
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Restart all processes

Restart all system processes. The operation is asynchronous. Use "GET system/status" to poll for status. On Enterprise systems, stop system processes on all modules.

```
POST https://{device}/api/gateway/1.1/system/restart
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Reboot

Reboot the system. The operation is asynchronous.

```
POST https://{device}/api/gateway/1.1/system/reboot
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Stop all processes (one module)

Stop all system processes on one module on Enterprise systems. The operation is asynchronous. Use "GET system/{module}/status" to poll for status. The {module} can be either the IP Address or the module name.

```
POST https://{device}/api/gateway/1.1/system/{module}/stop
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Kill all processes (one module)

Kill all system processes on one module on Enterprise systems. The operation is asynchronous. Use "GET system/{module}/status" to poll for status. The {module} can be either the IP Address or the module name. Warning: This operation can result in data being corrupted.

```
POST https://{device}/api/gateway/1.1/system/{module}/kill
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Get status of all processes (one module)

Get status of all system processes on one module on Enterprise systems. The {module} can be either the IP

Address or the module name.

```
GET https://{device}/api/gateway/1.1/system/{module}/status
```

Authorization

This request requires authorization.

Response Body

On success, the server returns a response body with the following structure:

JSON

```
[
  {
    "module_ipaddr": string,
    "module_name": string,
    "process_id": string,
    "process_name": string,
    "status": string
  }
]
```

Example:

```
[
  {
    "process_id": "25096",
    "process_name": "memmonitor",
    "status": "Running"
  },
  {
    "process_name": "healthd",
    "status": "Stopped"
  },
  {
    "process_id": "25092",
    "process_name": "diskmon",
    "status": "Running"
  },
  {
    "process_id": "25123",
    "process_name": "dispatcher",
    "status": "Running"
  },
  {
    "process_name": "analyzer",
    "status": "Stopped"
  }
]
```

Property Name	Type	Description	Notes
<i>status</i>	<array of <object>>	SystemStatus object.	
<i>status[process]</i>	<object>	SystemProcess object.	
<i>status[process].module_ipaddr</i>	<string>	Module IP Address. Available on Enterprise systems only.	Optional
<i>status[process].module_name</i>	<string>	Module name. Available on Enterprise systems only.	Optional
<i>status[process].process_id</i>	<string>	Process ID.	Optional
<i>status[process].process_name</i>	<string>	Process name.	
<i>status[process].status</i>	<string>	Process status.	Values: Running, Stopped

System: Start all processes (one module)

Start all system processes on one module on Enterprise systems. The operation is asynchronous. Use "GET system/{module}/status" to poll for status. The {module} can be either the IP Address or the module name.

```
POST https://{device}/api/gateway/1.1/system/{module}/start
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

System: Restart all processes (one module)

Restart all system processes on one module on Enterprise systems. The operation is asynchronous. Use "GET system/{module}/status" to poll for status. The {module} can be either the IP Address or the module name.

```
POST https://{device}/api/gateway/1.1/system/{module}/restart
```

Authorization

This request requires authorization.

Request Body

Do not provide a request body.

Response Body

On success, the server does not provide any body in the responses.

Error Codes

In the event that an error occurs while processing a request, the server will respond with appropriate HTTP status code and additional information in the response body:

```
{
  "error_id": "{error identifier}",
  "error_text": "{error description}",
  "error_info": {error specific data structure, optional}
}
```

The table below lists the possible errors and the associated HTTP status codes that may returned.

Error ID	HTTP Status	Comments
INTERNAL_ERROR	500	Internal server error.
AUTH_REQUIRED	401	The requested resource requires authentication.
AUTH_INVALID_CREDENTIALS	401	Invalid username and/or password.
AUTH_INVALID_SESSION	401	Session ID is invalid.
AUTH_EXPIRED_PASSWORD	403	The password must be changed. Access only to password change resources.
AUTH_DISABLED_ACCOUNT	403	Account is either temporarily or permanently disabled.
AUTH_FORBIDDEN	403	User is not authorized to access the requested resource.
AUTH_INVALID_TOKEN	401	OAuth access token is invalid.
AUTH_EXPIRED_TOKEN	401	OAuth access token is expired.
AUTH_INVALID_CODE	401	OAuth access code is invalid.
AUTH_EXPIRED_CODE	401	OAuth access code is expired.
RESOURCE_NOT_FOUND	404	Requested resource was not found.
HTTP_INVALID_METHOD	405	Requested method is not available for this resource.
HTTP_INVALID_HEADER	400	An HTTP header was malformed.
REQUEST_INVALID_INPUT	400	Malformed input structure.
URI_INVALID_PARAMETER	400	URI parameter is not supported or malformed.
URI_MISSING_PARAMETER	400	Missing required parameter.