Riverbed Cascade Gateway REST API. v1.2

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/1.2/ping
```

Authorization

This request requires authorization.

Response Body

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

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/1.2/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: 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/1.2/system/status
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "process_id": string,
    "process_name": string,
    "module_name": string,
    "status": string,
    "module_ipaddr": 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
SystemStatus	*<array of <object>>*	SystemStatus object.	
SystemStatus[SystemProcess]	*<object>*	SystemProcess object.	Optional
SystemStatus[SystemProcess].process_id	*<string>*	Process ID.	Optional
SystemStatus[SystemProcess].process_name	*<string>*	Process name.	
SystemStatus[SystemProcess].module_name	*<string>*	Module name. Available on Enterprise systems only.	Optional
SystemStatus[SystemProcess].status	*<string>*	Process status.	Values: Running, Stopped
SystemStatus[SystemProcess]. module_ipaddr	*<string>*	Module IP address. Available on Enterprise systems only.	Optional

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/1.2/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: 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/1.2/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: 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/1.2/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 (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/1.2/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.

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/1.2/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: 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/1.2/system/{module}/status
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "process_id": string,
    "process_name": string,
    "module_name": string,
    "status": string,
    "module_ipaddr": 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
SystemStatus	*<array of <object>>*	SystemStatus object.	
SystemStatus[SystemProcess]	*<object>*	SystemProcess object.	Optional
SystemStatus[SystemProcess].process_id	*<string>*	Process ID.	Optional
SystemStatus[SystemProcess].process_name	*<string>*	Process name.	
SystemStatus[SystemProcess].module_name	*<string>*	Module name. Available on Enterprise systems only.	Optional
SystemStatus[SystemProcess].status	*<string>*	Process status.	Values: Running, Stopped
SystemStatus[SystemProcess]. module_ipaddr	*<string>*	Module IP address. Available on Enterprise systems only.	Optional

---

System: Shutdown

Shutdown the system. The operation is asynchronous.

```
POST https://{device}/api/gateway/1.2/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: Reboot

Reboot the system. The operation is asynchronous.

```
POST https://{device}/api/gateway/1.2/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

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/1.2/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 (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/1.2/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.

Users: List users

Get a list of user accounts.

```
GET https://{device}/api/gateway/1.2/users
```

Authorization

This request requires authorization.

Response Body

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

JSON

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

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
Users	*<array of <object>>*	List of user accounts on the system.	
Users[User]	*<object>*	User account.	Optional
Users[User].enabled	*<string>*	Boolean flag indicating if the user account is enabled.	
Users[User].last_name	*<string>*	Last name of the user.	
Users[User].id	*<number>*	Numeric ID of the user that the system uses internally and in the API.	
Users[User].last_login	*<number>*	Time of last login. Unix time (epoch).	

Users[User].authentication_type	*<string>*	Type of authentication for the user, such as Local or RADIUS.	Values: Local, Remote
Users[User].username	*<string>*	User name (short name) that identifies the user to the system, such as 'admin'.	
Users[User].authorization_type	*<string>*	Type of authorization for the user, such as Local or RADIUS.	Values: Local, Remote
Users[User].role	*<string>*	Role of the user. Defines permissions.	Values: Developer, Administrator, Operator, Monitor, Event_Viewer, Dashboard_Viewer
Users[User].first_name	*<string>*	First name of the user.	
Users[User].last_access	*<number>*	Time of last access to the system. Unix time (epoch).	
Users[User].view_packet_details	*<string>*	Boolean flag indicating if the user has access to packet data.	Optional
Users[User].last_authentication	*<number>*	Time of last authentication. Unix time (epoch).	
Users[User].view_user_information	*<string>*	Boolean flag indicating if the user has access to identity information, such as Active Directory information.	Optional
Users[User].login_timeout	*<number>*	Timeout (in seconds) during which the user cannot log in to the system because of security policies.	

Users: Re-authenticate user

Re-authenticate user account. Requires basic authentication.

```
GET https://{device}/api/gateway/1.2/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

User account by user ID.

```
GET https://{device}/api/gateway/1.2/users/{user_id}
```

Authorization

This request requires authorization.

Response Body

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

JSON

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

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

Users: Test RADIUS user

Test a RADIUS user.

```
POST https://{device}/api/gateway/1.2/users/radius/test_user?password={string}&username={string}
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes

password	*<string>*	RADIUS password.	
username	*<string>*	RADIUS username.	

Request Body

Provide a request body with the following structure:

JSON

```
{
  "password": string,
  "username": string
}

Example:
{
  "username": "testusername",
  "password": "testpassword"
}
```

Property Name	Type	Description	Notes
RemoteTestUserRequest	*<object>*	RemoteTestUserRequest object.	
RemoteTestUserRequest.password	*<string>*	password.	
RemoteTestUserRequest.username	*<string>*	user name.	

Response Body

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

JSON

```
{
 "role_id": number,
 "error_message": string,
 "permission": string,
 "server_type": number,
 "role": string,
 "details": string,
 "permission_id": string,
 "server_ip": string,
 "authenticated": string,
 "attributes": [
  {
   [prop]: string
  }
 ],
 "authorized": string
}
```

Example:

```
{
 "error_message": "",
 "authenticated": true,
 "server_type": 2,
 "permission_id": "",
 "permission": "",
 "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_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

RemoteTestUserResponse	*<object>*	RemoteTestUserResponse object.	
RemoteTestUserResponse.role_id	*<number>*	Matched role ID.	
RemoteTestUserResponse.error_message	*<string>*	Error message.	
RemoteTestUserResponse.permission	*<string>*	Matched permission name.	
RemoteTestUserResponse.server_type	*<number>*	Indicates the type of the server being tested: RADIUS(2) or TACACS+(3).	
RemoteTestUserResponse.role	*<string>*	Matched role name.	
RemoteTestUserResponse.details	*<string>*	Remote user test details.	
RemoteTestUserResponse.permission_id	*<string>*	Matched permission ID.	
RemoteTestUserResponse.server_ip	*<string>*	Remote Server IP address.	
RemoteTestUserResponse.authenticated	*<string>*	Flag indicating if the remote user was authenticated.	
RemoteTestUserResponse.attributes	*<array of <object>>*	Attributes of Remote Test User Response.	Optional
RemoteTestUserResponse.attributes [RemoteAttributes]	*<object>*	Remote attribute.	Optional
RemoteTestUserResponse.attributes [RemoteAttributes][prop]	*<string>*	Remote attribute value.	Optional
RemoteTestUserResponse.authorized	*<string>*	Flag indicating if the remote user was authorized (as Administrator, Monitor, etc).	

Users: Test RADIUS server

Test the connection to a RADIUS server.

```
GET https://{device}/api/gateway/1.2/users/radius/test_server
```

Authorization

This request requires authorization.

Parameters

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

Response Body

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

JSON

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

Example:
{
  "message": "Connection attempt succeeded",
  "success": true
}
```

Property Name	Type	Description	Notes
RemoteTestServerResponse	*<object>*	RemoteTestServerResponse object.	
RemoteTestServerResponse.success	*<string>*	Flag indicating if the remote server test was successful.	
RemoteTestServerResponse.message	*<string>*	Response message.	

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.