Riverbed Cascade Profiler REST API. v1.0

Created Jan 16, 2024 at 02:01 PM

Overview

Overview

The documentation pages in this section describe the RESTful APIs included with Cascade Profiler and Cascade Express 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 primary focus of the current version of the API is on providing access to reporting data. The following information can be accessed via the API:

- Reporting data (traffic data, service data, event list data, and user data)
- Information about network devices
- Control over SDN virtual network settings (SDN virtual network settings are a licensed feature)
- 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.

Running a report

The steps for running a report are described below. An easy way to learn how to run reports and get data is to use the **Demo Reports** section of this documentation. In that section a number of example reports are listed. If you click on any of the examples, the report will run along with a listing on the right side of the screen for each step in the process. It displays the REST resource, HTTP method, headers and body for both the request and response.

Follow these steps to run a report:

1. Create a new report by posting report criteria to the server.

A criteria structure in either JSON or XML is submitted to the server using the HTTP POST method. The resource to which the criteria structure is posted is called **/profiler/1.0/reporting/reports**. The details are described in the **Resources** section of this documentation.

A key part of the report criteria is the ID of the template that should be used to run the report. A special system template ID 184 that provides a high degree of flexibility is used in demo reports in this documentation. Additionally, any template that is saved via the user interface of the product can be used to run a report. In order to save a template, the ID of that template must be passed in the report criteria structure instead of 184. A template can be configured via the user interface, saved via the product and then used in the REST API to generate reports and retrieve them in a rendered form or in raw data form. Once a template is saved, its ID can be obtained via the **/api/profiler/1.0/reporting/templates** REST resource.

2. Poll report status until the report completes.

It may take a while for a report to complete. While the report is running, the client must poll for report status to know when the report completes. When the call to create a new report succeeds, it returns the URL for the newly created report. That URL may look similar to */profiler/1.0/reporting/reports/1000*, which is the ID of the new report.

The general way to describe this in REST documentation is **/profiler/1.0/reporting/reports/{id}** so this documentation uses that syntax throughout. Note that the client does not need to know that **{id}** is really an ID of a report. Instead the client should treat a given report, for example /profiler/1.0/reporting/reports/1000, as a REST resource without parsing the parts of the URL.

The status of a report can be obtained by executing a GET method on the report URL. The client should wait while the report is running and until the status turns to state.

3. Retrieve the report data.

Once the report completes, the client can retrieve its data or the rendered version of the report in a number of formats.

The following resources can be used to retrieve a rendered version of the report:

- /profiler/1.0/reporting/reports/{id}/view.pdf
- /profiler/1.0/reporting/reports/{id}/view.csv

These are for PDF and CSV versions respectively.

If the client is a script that needs access to raw data, the **/profiler/1.0/reporting/reports/{id}/queries** resource can be used with the GET method to obtain the list of elements (queries) first. The data shown in a typical report on the user interface may come from more than one query, which is why the report structure needs to be determined first. However, the system template 184 described above will have only one query and is easy to use for simple scripts.

Each query resource provides metadata about the query, such as the list of columns with descriptions of what the columns are.

Once the query is chosen, the **/profiler/1.0/reporting/reports/{id}/queries/{id}** resource can be used to get the report data.

The simple overview provided above cannot substitute for full documentation and it is not intended to do so. Please refer to **Demo Reports** section to see how reports are run. Look at the **Coding Examples** under **General Information** and explore **Resources** section of this documentation for more information.

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.

Running a report: Sample PHP script

Run a report to get bytes and packets for the top 20 hosts using the application WEB. Use BASIC Authentication.

```
<?php

define('HOST', '127.0.0.1'); // IP address of Profiler
define('BASIC_AUTH', 'admin:admin');

// Timeframe
$end   = time() - 3*60;
$start = $end   - 5*60;

// Lib functions

// HTTP POST
function do_POST($url, $string, &$info) {
  $curl = curl_init();
  curl_setopt($curl, CURLOPT_URL, $url);
  curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC ) ;
  curl_setopt($curl, CURLOPT_USERPWD, BASIC_AUTH);
  curl_setopt($curl, CURLOPT_SSLVERSION,3);
  curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
  curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 2);
  curl_setopt($curl, CURLOPT_HEADER, true);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
  curl_setopt($curl, CURLOPT_POST,           1);
  curl_setopt($curl, CURLOPT_POSTFIELDS,     $string);
  $output = curl_exec($curl);
  $info   = curl_getinfo($curl);
  curl_close($curl);

  $headers = substr($output, 0, $info['header_size']);
  $headers = explode("\n", $headers);
  $info['headers'] = $headers;
  $body = substr($output, $info['header_size']);
  return $body;
}

// HTTP GET
function do_GET($url, &$info) {
  $curl = curl_init();
  curl_setopt($curl, CURLOPT_URL, $url);
  curl_setopt($curl, CURLOPT_HTTPAUTH, CURLAUTH_BASIC ) ;
  curl_setopt($curl, CURLOPT_USERPWD, BASIC_AUTH);
  curl_setopt($curl, CURLOPT_SSLVERSION,3);
  curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, FALSE);
  curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 2);
  curl_setopt($curl, CURLOPT_HEADER, true);
  curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
  curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));
  curl_setopt($curl, CURLOPT_HTTPGET, true);
```

```
  $output = curl_exec($curl);
  $info   = curl_getinfo($curl);
  curl_close($curl);

  $headers = substr($output, 0, $info['header_size']);
  $headers = explode("\n", $headers);
  $info['headers'] = $headers;
  $body = substr($output, $info['header_size']);
  return $body;
}

// Finds and returns HTTP header
function get_header($headers, $header) {
  foreach($headers as $h) {
    if (strpos($h, $header . ':') !== false)
      return trim(substr($h, 10));
  }
  echo "Unable to find {$header} header!\n";
  exit;
}

// Locates a column by id and returns the name
function find_column_name_by_id($columns, $id) {
  foreach ($columns as $c) {
    if ($c['id'] == $id)
      return $c['name'];
  }
  return 'Unknown';
}

// CSV helper
function echo_csv($headers, $rows) {
  echo implode(',', $headers) . "\n";
  foreach ($rows as $row)
    echo implode(',', $row) . "\n";
}

// End lib functions

$struct =
  array('template_id' => 184,
        'criteria'    => Array('time_frame' => array('start' => $start,
                                                     'end'   => $end),
                               'traffic_expression' => 'app WEB',
                               'query' => array('realm'       => 'traffic_summary',
                                                'group_by'    => 'hos',
                                                'sort_column' => 33,
                                                'columns'     => array(6, 33, 34))));

$json    = json_encode($struct);
$columns = $struct['criteria']['query']['columns'];

// Post to run the report
$url = 'https://' . HOST . '/api/profiler/1.0/reporting/reports.json';
echo "Run report:\nPOST {$url}\n{$json}\n\n";
$info = array();
do_POST($url, $json, $info);
if ($info['http_code'] != 201) {
  echo "Unable to run report!\n";
  exit(1);
}
$location = get_header($info['headers'], 'Location');
echo "Generated: {$location}\n\n";
$status_url = 'https://' . HOST . '' . $location . '.json';

// Wait for it to complete
echo "Please wait\n";
while (true) {
  $info = array();
  $output = do_GET($status_url, $info);
  $s = json_decode($output, 1);
  print " Percent completed {$s['percent']}, seconds remaining {$s['remaining_seconds']}...\n";
  if ($s['status'] == 'completed') {
    echo "Completed\n\n";
    break;
  }
  sleep(1);
}

// Get all quesries (In this exampe it is only one)
$queries_url = 'https://' . HOST . '' . $location . '/queries.json';
$output = do_GET($queries_url, $info);
```

```
$queries = json_decode($output, 1);

// Print the data from all queries
foreach ($queries as $q) {
  $query_id = $q['id'];
  $data_url = 'https://' . HOST . '' . $location . '/queries/' . $query_id . '.json?offset=0&limit=20&columns=' . join(',', $columns);
  $info = array();
  $output = do_GET($data_url, $info);
  $data = json_decode($output, 1);

  $h = array();
  foreach ($columns as $c)
    $h[] = find_column_name_by_id($q['columns'], $c);

  echo_csv($h, $data['data']);
  echo "\n";
}

?>
```

Running a report: Sample Python script

Run a report to get bytes and packets for the top 20 hosts using the application WEB. Use BASIC Authentication.

```
from urlparse import urlparse
import base64
import logging
import httplib
import json
import time
import sys

HOST       = '127.0.0.1'
BASIC_AUTH = 'admin:admin'

end   = int(time.time() - 3*60)
start = int(end - 5*60);

# Lib functions

def do_POST(url, string):
    '''HTTP POST'''

    conn = httplib.HTTPSConnection(HOST, 443)

    headers = {"Authorization"  : "Basic %s" % base64.b64encode(BASIC_AUTH),
               "Content-Length" : str(len(string)),
               "Content-Type"   : "application/json"}

    conn.request('POST', url, body=string, headers=headers)

    res = conn.getresponse()

    info = {"status"  : res.status,
            "headers" : res.getheaders()}

    data = res.read()
    conn.close()
    return data, info

def do_GET(url):
    '''HTTP GET'''

    conn = httplib.HTTPSConnection(HOST, 443)

    headers = {"Authorization"  : "Basic %s" % base64.b64encode(BASIC_AUTH),
               "Content-Length" : 0,
               "Content-Type"   : "application/json"}

    conn.request('GET', url, body="", headers=headers)

    res = conn.getresponse()

    info = {"status"  : res.status,
            "headers" : res.getheaders()}

    data = res.read()
    conn.close()
    return data, info
```

```
def get_header(headers, header):
  '''Finds and returns HTTP header'''
  for i in headers:
    if (i[0] == header):
      return i[1]
  return ""

def find_column_name_by_id(columns, cid):
  '''Locates a column by id and returns the name'''
  for c in columns:
    if (c['id'] == cid):
      return c['name']
  return 'Unknown'

def echo_csv(headers, rows):
  '''CSV helper'''
  print ','.join(headers)
  for row in rows:
    print ','.join(row)

# End lib functions

struct = {
    "template_id" : 184,
    "criteria" : {
        "time_frame" : {
            "start" : start,
            "end"   : end
            },
        "traffic_expression" : "app WEB",
        "query" : {
            "realm"      : "traffic_summary",
            "group_by"   : "hos",
            "sort_column": 33,
            "columns"    : [6, 33, 34]
            }
        }
    }

to_post = json.dumps(struct)
columns = struct["criteria"]["query"]["columns"]

# Post to run the report
url = "https://%s/api/profiler/1.0/reporting/reports.json" % HOST
print "Run report:"
print "POST %s" % url
print "%s" % to_post

output, info = do_POST(url, to_post)
if (info['status'] is not 201):
  print "Unable to run report"
  sys.exit(1)

location = get_header(info['headers'], 'location')
print ""
print "Generated: %s" % location
print ""

status_url = "https://%s%s.json" % (HOST, location)

# Wait for it to complete
print "Please wait"
while (True):
  output, info = do_GET(status_url)
  s = json.loads(output)
  print "Percent completed %s, seconds remaining %s..." % (s["percent"], s["remaining_seconds"])
  if (s["status"] == "completed"):
    print "Completed"
    break
  time.sleep(1)

# Get all quesries (In this exampe it is only one)
queries_url = "https://%s%s/queries.json" % (HOST, location)

output, info = do_GET(queries_url)
queries = json.loads(output)

# Print the data from all queries
for q in queries:
  query_id = q['id'];
  columns_str = ' '.join([repr(i) for i in columns])
```

```
  columns_str = ",".join([repr(i) for i in columns])
  data_url = "https://%s%s/queries/%s.json?offset=0&limit=20&columns=%s" % (HOST, location, query_id, columns_str)
  output, info = do_GET(data_url)
  data = json.loads(output)

  h = []
  for c in columns:
    h.append(find_column_name_by_id(q["columns"], c))

  print ""
  echo_csv(h, data["data"])
```

Running a report: Sample Perl script

Run a report to get bytes and packets for the top 20 hosts using the application WEB. Use BASIC Authentication.

```
#!/usr/bin/perl
use strict;
use warnings;

use LWP::UserAgent;
use HTTP::Request;
use List::MoreUtils qw(firstidx);
use JSON qw( encode_json decode_json );

use constant HOST     => '127.0.0.1';
use constant LOGIN    => 'admin';
use constant PASSWORD => 'admin';

our $ua = LWP::UserAgent->new;
$ua->agent("ProfilerScript/0.1");

our $API_BASE = "https://127.0.0.1";

sub _request($)
{
  my $req = shift;

  $req->header('Accept' => 'application/json');
  $req->authorization_basic(LOGIN, PASSWORD);

  my $res = $ua->request($req);

  return {
    code    => $res->code,
    status  => $res->status_line,
    headers => $res->headers(),
    data    => decode_json($res->content)
  };
}

sub GET($)
{
  my $req = HTTP::Request->new(GET => $API_BASE . shift);
  return _request($req);
}

sub POST($$)
{
  my $req = HTTP::Request->new(POST => $API_BASE . shift);
  $req->content_type('application/json');
  $req->content(encode_json(shift));

  return _request($req);
}

my $end   = time();
my $start = $end - 5 * 60;

my $struct = {
  template_id => 184,
  criteria    => {
    time_frame => {
      start => $start,
      end   => $end
    },
    traffic_expression => "app WEB",
    query => {
      realm        => "traffic_summary",
      group_by     => "hos"
```

```
    group_by    => "host",
    sort_column  => 33,
    columns      => [6, 33, 34]
   }
  }
};

print "Running report... ";

my $response = POST('/api/profiler/1.0/reporting/reports', $struct);
die "Unable to run report. $response->{data}->{error_text}" unless $response->{code} == 201;

my $loc = $response->{headers}->header('Location');

while (1)
{
  $response = GET($loc);
  printf "\rRunning report, %3d%% done, %d seconds remaining...   ",
    $response->{data}->{percent},
    $response->{data}->{remaining_seconds};

  last if $response->{data}->{status} eq 'completed';
  sleep(1);
};

print "\nLoading data...\n";

$response = GET($loc . '/queries');
die "Unable to load queries. $response->{data}->{error_text}" unless $response->{code} == 200;

foreach my $query (@{$response->{data}})
{
  my @columns = @{$struct->{criteria}->{query}->{columns}};
  my $columns = join ',', @columns;

  my $data_response = GET("$loc/queries/$query->{id}?offset=0&limit=20&columns=$columns");
  die "Unable to load data. $response->{data}->{error_text}" unless $response->{code} == 200;

  my @indices = map { my $id = $_; firstidx { $_->{id} == $id } @{$query->{columns}} } @columns;

  print join ",", map { qq~"$query->{columns}->[$_]->{name}"~; } @indices;
  print "\n";

  foreach my $row (@{$data_response->{data}->{data}}) {
    print join ",", @$row;
    print "\n";
  }
}
```

Running a report: Sample .Net/C# code

Run a report to get bytes and packets for the top 20 hosts using the application WEB. Use BASIC Authentication.

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Runtime.Serialization.Json;
using System.Text;
using System.IO;
using System.Net.Security;
using System.Security.Cryptography.X509Certificates;
using System.Linq;
using System.Threading;
using System.Web.Script.Serialization;

namespace CascadeRestClient
{
    public class ReportUpdate
    {
        public string status { get; set; }
        public string user_id { get; set; }
        public string name { get; set; }
        public string percent { get; set; }
        public string id { get; set; }
        public string remaining_seconds { get; set; }
        public string run_time { get; set; }
        public string saved { get; set; }
        public string template_id { get; set; }
```

```
    public string template_id { get; set; }
    public string size { get; set; }
}

public class Column
{
    public string strid { get; set; }
    public string metric { get; set; }
    public string rate { get; set; }
    public string statistic { get; set; }
    public int id { get; set; }
    public string unit { get; set; }
    public string category { get; set; }
    public string severity { get; set; }
    public string area { get; set; }
    public bool @internal { get; set; }
    public string role { get; set; }
    public string cli_srv { get; set; }
    public string type { get; set; }
    public bool available { get; set; }
    public string direction { get; set; }
    public string comparison { get; set; }
    public bool sortable { get; set; }
    public string name { get; set; }
    public string comparison_parameter { get; set; }
    public bool has_others { get; set; }
    public bool context { get; set; }
    public string name_type { get; set; }
}

public class QueryResult
{
    public string direction { get; set; }
    public string actual_log { get; set; }
    public int actual_t0 { get; set; }
    public bool sort_desc { get; set; }
    public string area { get; set; }
    public string metric { get; set; }
    public int sort_col { get; set; }
    public string parent_id { get; set; }
    public string rate { get; set; }
    public string group_by { get; set; }
    public string role { get; set; }
    public List&lt;Column&gt; columns { get; set; }
    public string statistic { get; set; }
    public string type { get; set; }
    public string id { get; set; }
    public string unit { get; set; }
    public int actual_t1 { get; set; }
}

public class QueryData
{
    public List&lt;List&lt;string&gt;&gt; data { get; set; }
    public int data_size { get; set; }
    public List&lt;string&gt; totals { get; set; }
}

class Program
{
    static string BASIC_AUTH = "admin:admin";

    // callback used to validate the self-gen certificate in an SSL conversation
    private static bool ValidateRemoteCertificate(object sender, X509Certificate cert, X509Chain chain, SslPolicyErrors policyErrors)
    {
        return true;
        /*
        X509Certificate2 certv2 = new X509Certificate2(cert);
        if (certv2.GetNameInfo(X509NameType.SimpleName,true) == "www.riverbed.com")
            return true;

        return false;
         */
    }

    static void Main(string[] args)
    {
        if (args.Length == 0 || string.IsNullOrWhiteSpace(args[0]))
        {
            Console.WriteLine("Usage: CascadeRestClient hostname");
            return;
        }
```

```
        try
        {
            //Code to allow run with self-signed certificates
            // validate cert by calling a function
            ServicePointManager.ServerCertificateValidationCallback += new RemoteCertificateValidationCallback(ValidateRemoteCertificate);

            //Starting to run rest
            string rootUrl = "https://" + args[0];
            string requestUrl = rootUrl + "/api/profiler/1.0/reporting/reports.json";
            string location;

            int start = (int)((DateTime.Now - new DateTime(1970, 1, 1).ToLocalTime()).TotalSeconds) - 8*60; //8 minutes before in unix time
            int end = start + 5*60; //3 minutes before in unix time

            var jsondata = new
            {
                template_id = 184,
                criteria = new
                {
                    time_frame = new
                    {
                        start = start,
                        end = end
                    },

                    traffic_expression = "app WEB",
                    query = new
                    {
                        realm = "traffic_summary",
                        group_by = "hos",
                        sort_column = 33,
                        columns = new List&lt;int&gt; { 6, 33, 34 }
                    }
                }
            };

            //Serialize anomymous type to json
            JavaScriptSerializer serializer = new JavaScriptSerializer();
            string postData = serializer.Serialize(jsondata);

            Console.WriteLine("Run report:");
            Console.WriteLine("POST " + requestUrl);
            Console.WriteLine(postData + Environment.NewLine);

            // Post to run the report
            var runReportResponse = MakeRequest&lt;ReportUpdate&gt;(requestUrl, WebRequestMethods.Http.Post, out location, postData);
            Console.WriteLine("Generated " + location + Environment.NewLine);
            requestUrl = rootUrl + location;
            Console.WriteLine("Please wait");
            while (runReportResponse.status != "completed")
            {
                runReportResponse = MakeRequest&lt;ReportUpdate&gt;(requestUrl + ".json", WebRequestMethods.Http.Get, out location);
                Console.WriteLine(string.Format("Percent completed {0}, seconds remaining {1}",runReportResponse.percent,
runReportResponse.remaining_seconds));
                Thread.Sleep(1000);
            }
            Console.WriteLine("Completed"+ Environment.NewLine);

            // Get all quesries (In this example it is only one)
            var getQueriesResponse = MakeRequest&lt;List&lt;QueryResult&gt;&gt;(requestUrl +"/queries.json", WebRequestMethods.Http.Get, out
location);
            string columns = jsondata.criteria.query.columns.Select(c=&gt;c.ToString()).Aggregate((i, j) =&gt; i + "," + j);
            // Print the data from all queries
            foreach (var query in getQueriesResponse) {
                var qr = MakeRequest&lt;QueryData&gt;(requestUrl + "/queries/" + query.id + ".json?offset=0&amp;limit=20&amp;columns=" + columns,
                                    WebRequestMethods.Http.Get, out location);
                string columnList = jsondata.criteria.query.columns.Select(c=&gt;query.columns.Where(col =&gt; col.id == c).First().name)
                                            .Aggregate((l,r) =&gt; l + "," + r);
                Console.WriteLine(columnList);

                foreach (var dr in qr.data)
                {
                    Console.WriteLine(dr.Aggregate((i, j) =&gt; i + ',' + j));
                }
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e.Message);
        }
    }
```

```
        private static string Base64Encode(string toEncode)
        {
            byte[] toEncodeAsBytes
            = System.Text.ASCIIEncoding.ASCII.GetBytes(toEncode);
            return System.Convert.ToBase64String(toEncodeAsBytes);
        }

        /// &lt;summary&gt;
        /// Make request
        /// &lt;/summary&gt;
        /// &lt;typeparam name="T"&gt;return type&lt;/typeparam&gt;
        /// &lt;param name="requestUrl"&gt;url for request&lt;/param&gt;
        /// &lt;param name="action"&gt;Http Verb, Get, Post etc&lt;/param&gt;
        /// &lt;param name="location"&gt;location returned from response header &lt;/param&gt;
        /// &lt;param name="requestData"&gt;Data posted&lt;/param&gt;
        /// &lt;returns&gt;&lt;/returns&gt;
        private static T MakeRequest&lt;T&gt;(string requestUrl, string action, out string location, string requestData = null) where T : class
        {
            HttpWebRequest request = WebRequest.Create(requestUrl) as HttpWebRequest;
            request.Headers.Add("Authorization: Basic " + Base64Encode(BASIC_AUTH));
            request.ContentType = "application/json";
            request.Method = action;
            if (requestData == null)
            {
                request.ContentLength = 0;
            }
            else
            {
                ASCIIEncoding encoding = new ASCIIEncoding();
                byte[] byte1 = encoding.GetBytes(requestData);
                request.ContentLength = byte1.Length;
                using (Stream newStream = request.GetRequestStream())
                {
                    newStream.Write(byte1, 0, byte1.Length);
                }
            }

            using (HttpWebResponse response = request.GetResponse() as HttpWebResponse)
            {
                if (response.StatusCode != HttpStatusCode.OK &amp;amp; response.StatusCode != HttpStatusCode.Created)
                    throw new Exception(String.Format(
                    "Unable to run report! StatusCode={0}, Description={1}",
                    response.StatusCode,
                    response.StatusDescription));
                location = response.Headers[HttpResponseHeader.Location];
                DataContractJsonSerializer jsonSerializer = new DataContractJsonSerializer(typeof(T));
                object objResponse = jsonSerializer.ReadObject(response.GetResponseStream());
                return objResponse as T;
            }
        }
    }
}
```

CascadeRestClient.csproj:

```
&lt;?xml version="1.0" encoding="utf-8"?&gt;
&lt;Project ToolsVersion="4.0" DefaultTargets="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003"&gt;
  &lt;PropertyGroup&gt;
    &lt;Configuration Condition=" '$(Configuration)' == '' "&gt;Debug&lt;/Configuration&gt;
    &lt;Platform Condition=" '$(Platform)' == '' "&gt;x86&lt;/Platform&gt;
    &lt;ProductVersion&gt;8.0.30703&lt;/ProductVersion&gt;
    &lt;SchemaVersion&gt;2.0&lt;/SchemaVersion&gt;
    &lt;ProjectGuid&gt;{4ED69347-523B-46AB-B259-47EF60D4F13A}&lt;/ProjectGuid&gt;
    &lt;OutputType&gt;Exe&lt;/OutputType&gt;
    &lt;AppDesignerFolder&gt;Properties&lt;/AppDesignerFolder&gt;
    &lt;RootNamespace&gt;CascadeRestClient&lt;/RootNamespace&gt;
    &lt;AssemblyName&gt;CascadeRestClient&lt;/AssemblyName&gt;
    &lt;TargetFrameworkVersion&gt;v4.0&lt;/TargetFrameworkVersion&gt;
    &lt;TargetFrameworkProfile&gt;
    &lt;/TargetFrameworkProfile&gt;
    &lt;FileAlignment&gt;512&lt;/FileAlignment&gt;
  &lt;/PropertyGroup&gt;
  &lt;PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|x86' "&gt;
    &lt;PlatformTarget&gt;x86&lt;/PlatformTarget&gt;
    &lt;DebugSymbols&gt;true&lt;/DebugSymbols&gt;
    &lt;DebugType&gt;full&lt;/DebugType&gt;
    &lt;Optimize&gt;false&lt;/Optimize&gt;
    &lt;OutputPath&gt;bin\Debug\&lt;/OutputPath&gt;
    &lt;DefineConstants&gt;DEBUG;TRACE&lt;/DefineConstants&gt;
    &lt;ErrorReport&gt;prompt&lt;/ErrorReport&gt;
    &lt;WarningLevel&gt;4&lt;/WarningLevel&gt;
  &lt;/PropertyGroup&gt;
  &lt;PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|x86' "&gt;
    &lt;PlatformTarget&gt;x86&lt;/PlatformTarget&gt;
    &lt;DebugType&gt;pdbonly&lt;/DebugType&gt;
    &lt;Optimize&gt;true&lt;/Optimize&gt;
    &lt;OutputPath&gt;bin\Release\&lt;/OutputPath&gt;
    &lt;DefineConstants&gt;TRACE&lt;/DefineConstants&gt;
    &lt;ErrorReport&gt;prompt&lt;/ErrorReport&gt;
    &lt;WarningLevel&gt;4&lt;/WarningLevel&gt;
  &lt;/PropertyGroup&gt;
  &lt;ItemGroup&gt;
    &lt;Reference Include="System" /&gt;
    &lt;Reference Include="System.Core" /&gt;
    &lt;Reference Include="System.Runtime.Serialization" /&gt;
    &lt;Reference Include="System.Web.Extensions"&gt;
      &lt;HintPath&gt;..\..\..\..\..\..\..\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETFramework\v4.0\System.Web.Extensions.dll&lt;/HintPath&gt;
    &lt;/Reference&gt;
    &lt;Reference Include="System.Xml.Linq" /&gt;
    &lt;Reference Include="System.Data.DataSetExtensions" /&gt;
    &lt;Reference Include="Microsoft.CSharp" /&gt;
    &lt;Reference Include="System.Data" /&gt;
    &lt;Reference Include="System.Xml" /&gt;
  &lt;/ItemGroup&gt;
  &lt;ItemGroup&gt;
    &lt;Compile Include="Program.cs" /&gt;
    &lt;Compile Include="Properties\AssemblyInfo.cs" /&gt;
  &lt;/ItemGroup&gt;
  &lt;Import Project="$(MSBuildToolsPath)\Microsoft.CSharp.targets" /&gt;
  &lt;!-- To modify your build process, add your task inside one of the targets below and uncomment it.
       Other similar extension points exist, see Microsoft.Common.targets.
  &lt;Target Name="BeforeBuild"&gt;
  &lt;/Target&gt;
  &lt;Target Name="AfterBuild"&gt;
  &lt;/Target&gt;
  --&gt;
&lt;/Project&gt;
```

Resources

Vnis: List VNIs

Get a list of Virtual Network Identifiers.

```
GET https://{device}/api/profiler/1.0/vnis
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": number,
    "description": string,
    "name": string
  }
]

Example:
[
  {
    "description": "Customer A. Blue Network.",
    "name": "Blue_Network",
    "id": 100
  },
  {
    "description": "Customer B. Blue Network.",
    "name": "Red_Network",
    "id": 200
  }
]
```

Property Name	Type	Description	Notes
VNIs	*<array of <object>>*	List of VNIs (Virtual Network Identifiers of SDN setup).	
VNIs[VNI]	*<object>*	Object representing a VNI.	Optional
VNIs[VNI].id	*<number>*	ID of the VNI.	
VNIs[VNI].description	*<string>*	Description of the VNI.	Optional
VNIs[VNI].name	*<string>*	Name of the VNI.	Optional

Vnis: Delete VNI

Delete a Virtual Network Identifier.

```
DELETE https://{device}/api/profiler/1.0/vnis/{vni_id}
```

Authorization

This request requires authorization.

Response Body

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

Vnis: Update VNIs

Update one or many Virtual Network Identifiers.

```
PUT https://{device}/api/profiler/1.0/vnis
```

Authorization

This request requires authorization.

Request Body

Provide a request body with the following structure:

JSON

```
[
  {
    "id": number,
    "description": string,
    "name": string
  }
]

Example:
[
  {
    "description": "Customer A. Blue Network.",
    "name": "Blue_Network",
    "id": 100
  },
  {
    "description": "Customer B. Blue Network.",
    "name": "Red_Network",
    "id": 200
  }
]
```

Property Name	Type	Description	Notes
VNIs	*<array of <object>>*	List of VNIs (Virtual Network Identifiers of SDN setup).	
VNIs[VNI]	*<object>*	Object representing a VNI.	Optional
VNIs[VNI].id	*<number>*	ID of the VNI.	
VNIs[VNI].description	*<string>*	Description of the VNI.	Optional
VNIs[VNI].name	*<string>*	Name of the VNI.	Optional

Response Body

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

Vnis: Get VNI

Get a Virtual Network Identifier.

```
GET https://{device}/api/profiler/1.0/vnis/{vni_id}
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
{
  "id": number,
  "description": string,
  "name": string
}

Example:
{
  "description": "Customer A. Blue Network.",
  "name": "Blue_Network",
  "id": 100
}
```

Property Name	Type	Description	Notes
VNI	*<object>*	Object representing a VNI.	
VNI.id	*<number>*	ID of the VNI.	
VNI.description	*<string>*	Description of the VNI.	Optional
VNI.name	*<string>*	Name of the VNI.	Optional

Vnis: Update VNI

Update one Virtual Network Identifier.

PUT https://{device}/api/profiler/1.0/vnis/{vni_id}

Authorization

This request requires authorization.

Request Body

Provide a request body with the following structure:

JSON

```
{
 "id": number,
 "description": string,
 "name": string
}

Example:
{
 "description": "Customer A. Blue Network.",
 "name": "Blue_Network",
 "id": 100
}
```

Property Name	Type	Description	Notes
VNI	*<object>*	Object representing a VNI.	
VNI.id	*<number>*	ID of the VNI.	
VNI.description	*<string>*	Description of the VNI.	Optional
VNI.name	*<string>*	Name of the VNI.	Optional

Response Body

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

Devices: Get device

Get a device by IP Address.

GET https://{device}/api/profiler/1.0/devices/{device_ip}

Authorization

This request requires authorization.

Response Body

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

JSON

```
{
 "id": number,
 "type_id": number,
 "ipaddr": string,
 "name": string,
 "type": string,
 "version": string
}

Example:
{
 "name": "MyNetflowDevice",
 "type_id": 2,
 "ipaddr": "10.0.0.1",
 "version": "N/A",
 "type": "Netflow",
 "id": 123
}
```

Property Name	Type	Description	Notes
Device	*<object>*	Object representing a device.	
Device.id	*<number>*	Device identifier (ID). Used internally in the product and in the API.	

Device.type_id	*<number>*	Device type ID; a way to represent device type that is more friendly to programs.	
Device.ipaddr	*<string>*	Device IP address.	
Device.name	*<string>*	Device name, which usually comes from SNMP or DNS.	
Device.type	*<string>*	Device type, e.g. Cascade Gateway, Cascade Shark or Netflow device.	
Device.version	*<string>*	Version of the protocol used to communicate with the device.	

Devices: List devices

Get a list of devices.

```
GET https://{device}/api/profiler/1.0/devices?type_id={number}&cidr={string}
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes
type_id	*<number>*	Filter devices by device type.	Optional
cidr	*<string>*	Filter devices by IP or Subnet (e.g. 10.0.0.0/8).	Optional

Response Body

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

JSON

```
[
  {
    "id": number,
    "type_id": number,
    "ipaddr": string,
    "name": string,
    "type": string,
    "version": string
  }
]

Example:
[
  {
    "name": "MyNetflowDevice",
    "type_id": 2,
    "ipaddr": "10.0.0.1",
    "version": "N/A",
    "type": "Netflow",
    "id": 123
  },
  {
    "name": "MySensorDevice",
    "type_id": 1,
    "ipaddr": "10.0.0.2",
    "version": "M8.4",
    "type": "Sensor",
    "id": 124
  }
]
```

Property Name	Type	Description	Notes
Devices	*<array of <object>>*	List of network devices that report data to Profiler.	
Devices[Device]	*<object>*	One device from the list of devices that report data.	Optional
Devices[Device].id	*<number>*	Device identifier (ID). Used internally in the product and in the API.	
Devices[Device].type_id	*<number>*	Device type ID; a way to represent device type that is more friendly to programs.	
Devices[Device].ipaddr	*<string>*	Device IP address.	
Devices[Device].name	*<string>*	Device name, which usually comes from SNMP or DNS.	

Devices[Device].type	*<string>*	Device type, e.g. Cascade Gateway, Cascade Shark or Netflow device.	
Devices[Device].version	*<string>*	Version of the protocol used to communicate with the device.	

Devices: Delete device

Delete a device by IP Address. Warning: the device will be deleted in a few minutes after this call.

```
DELETE https://{device}/api/profiler/1.0/devices/{device_ip}
```

Authorization

This request requires authorization.

Response Body

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

Ping: Ping

Simple test of service availability.

```
GET https://{device}/api/profiler/1.0/ping
```

Authorization

This request requires authorization.

Response Body

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

Reporting: List reports

Get a list of reports with their respective status.

```
GET https://{device}/api/profiler/1.0/reporting/reports
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "run_time": number,
    "error_text": string,
    "remaining_seconds": number,
    "saved": string,
    "id": number,
    "status": string,
    "percent": number,
    "user_id": number,
    "size": number,
    "name": string,
    "template_id": number
  }
]

Example:
[
  {
    "status": "completed",
    "user_id": 1,
    "name": "",
    "percent": 100,
    "id": 1000,
    "remaining_seconds": 0,
    "run_time": 1352494550,
    "saved": false,
    "template_id": 952,
    "error_text": "",
    "size": 140
  },
  {
    "status": "completed",
    "user_id": 1,
    "name": "Host Information Report",
    "percent": 100,
    "id": 1001,
    "remaining_seconds": 0,
    "run_time": 1352494550,
    "saved": true,
    "template_id": 952,
    "error_text": "",
    "size": 140
  }
]
```

Property Name	Type	Description	Notes
ReportInfoList	*<array of <object>>*	List of report objects.	
ReportInfoList[ReportInfo]	*<object>*	Object representing report information.	Optional
ReportInfoList[ReportInfo].run_time	*<number>*	Time when the report was run (Unix time).	
ReportInfoList[ReportInfo].error_text	*<string>*	A report can be completed with an error. Error message may provide more detailed info.	Optional
ReportInfoList[ReportInfo]. remaining_seconds	*<number>*	Number of seconds remaining to run the report. Even if this number is 0, the report may not yet be completed, so check 'status' to make sure what the status is.	
ReportInfoList[ReportInfo].saved	*<string>*	Boolean flag indicating if the report was saved.	
ReportInfoList[ReportInfo].id	*<number>*	ID of the report. To be used in the API.	
ReportInfoList[ReportInfo].status	*<string>*	Status of the report.	Values: completed, running, waiting
ReportInfoList[ReportInfo].percent	*<number>*	Progress of the report represented by percentage of report completion.	
ReportInfoList[ReportInfo].user_id	*<number>*	ID of the user who owns the report.	
ReportInfoList[ReportInfo].size	*<number>*	Size of the report in kilobytes.	
ReportInfoList[ReportInfo].name	*<string>*	Name of the report. Could be given by a user or automatically generated by the system.	Optional
ReportInfoList[ReportInfo].template_id	*<number>*	ID of the template that the report is based on.	

Reporting: List directions

Get a list of directions that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/directions
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "inn",
    "name": "in"
  },
  {
    "id": "out",
    "name": "out"
  },
  {
    "id": "c2s",
    "name": "client to server"
  },
  {
    "id": "s2c",
    "name": "server to client"
  }
]
```

Property Name	Type	Description	Notes
Directions	*<array of <object>>*	List of directions.	
Directions[Direction]	*<object>*	Object representing a direction.	Optional
Directions[Direction].id	*<string>*	ID of a direction. To be used in the API.	
Directions[Direction].name	*<string>*	Human-readable name of a direction.	

Reporting: List categories

Get a list of categories that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/categories
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "idx",
    "name": "index"
  },
  {
    "id": "key",
    "name": "key"
  }
]
```

Property Name	Type	Description	Notes
Categories	*<array of <object>>*	List of categories.	
Categories[Category]	*<object>*	Object representing a category.	Optional
Categories[Category].id	*<string>*	ID of a category. To be used in the API.	
Categories[Category].name	*<string>*	Human-readable name of a category.	

Reporting: List statictics

Get a list of statistics that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/statistics
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "tot",
    "name": "total"
  },
  {
    "id": "avg",
    "name": "average"
  },
  {
    "id": "pek",
    "name": "peak"
  },
  {
    "id": "min",
    "name": "min"
  }
]
```

Property Name	Type	Description	Notes
Statistics	*<array of <object>>*	List of statistics.	
Statistics[Statistic]	*<object>*	Object representing a statistic.	Optional
Statistics[Statistic].id	*<string>*	ID of a statistic. To be used in the API.	

Statistics[Statistic].name	*<string>*	Human-readable name of a statistic.	

Reporting: Delete report

Delete a report.

```
DELETE https://{device}/api/profiler/1.0/reporting/reports/{report_id}
```

Authorization

This request requires authorization.

Response Body

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

Reporting: List realms

Get a list of realms.

```
GET https://{device}/api/profiler/1.0/reporting/realms
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
   "id": string,
   "name": string
  }
]

Example:
[
  {
   "id": "traffic_summary",
   "name": "traffic summary"
  },
  {
   "id": "traffic_flow_list",
   "name": "traffic flow list"
  },
  {
   "id": "traffic_overall_time_series",
   "name": "traffic overall time series"
  }
]
```

Property Name	Type	Description	Notes
Realms	*<array of <object>>*	List of type of report queries (realms) that could be requested and run.	
Realms[Realm]	*<object>*	Object representing a realm.	Optional
Realms[Realm].id	*<string>*	ID of a realm. To be used in the API.	
Realms[Realm].name	*<string>*	Human-readable name of a realm.	

Reporting: List centricities

Get a list of centricities that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/centricities
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "hos",
    "name": "host"
  },
  {
    "id": "int",
    "name": "interface"
  }
]
```

Property Name	Type	Description	Notes
Centricities	*<array of <object>>*	List of centricities.	
Centricities[Centricity]	*<object>*	Object representing a centricity.	Optional
Centricities[Centricity].id	*<string>*	ID of a centricity. To be used in the API.	
Centricities[Centricity].name	*<string>*	Human-readable name of a centricity.	

Reporting: List units

Get a list of units that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/units
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "byt",
    "name": "bytes"
  },
  {
    "id": "pkt",
    "name": "packet"
  },
  {
    "id": "con",
    "name": "conntions"
  }
]
```

Property Name	Type	Description	Notes
Units	*<array of <object>>*	List of units.	

Units[Unit]	*<object>*	Object representing a unit.	Optional
Units[Unit].id	*<string>*	ID of a unit. To be used in the API.	
Units[Unit].name	*<string>*	Human-readable name of a unit.	

Reporting: List severities

Get a list of severities that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/severities
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "nav",
    "name": "not available"
  },
  {
    "id": "nml",
    "name": "normal"
  },
  {
    "id": "low",
    "name": "low"
  },
  {
    "id": "med",
    "name": "medium"
  },
  {
    "id": "hgh",
    "name": "high"
  },
  {
    "id": "all",
    "name": "all"
  }
]
```

Property Name	Type	Description	Notes
Severities	*<array of <object>>*	List of severities.	
Severities[Severity]	*<object>*	Object representing a severity.	Optional
Severities[Severity].id	*<string>*	ID of a severity. To be used in the API.	
Severities[Severity].name	*<string>*	Human-readable name of a severity.	

Reporting: Get query data

Get data for one or many columns from this query.

```
GET https://{device}/api/profiler/1.0/reporting/reports/{report_id}/queries/{query_id}?columns={string}&offset={number}&limit={number}
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes
columns	*<string>*	Comma-separated list of column ids.	Optional
offset	*<number>*	Start row.	Optional
limit	*<number>*	Number of rows to be returned.	Optional

Response Body

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

JSON

```
{
 "data": [
  [
   string
  ]
 ],
 "data_size": number,
 "totals": [
  string
 ]
}

Example:
{
 "data": [
  [
   "10.38.8.202|",
   "6878717.15556",
   "7041.06111111"
  ],
  [
   "10.38.9.152|",
   "1996165.24167",
   "2049.01388889"
  ]
 ],
 "data_size": 3744,
 "totals": [
  "",
  "20293913.8417",
  "23577.3055556"
 ]
}
```

Property Name	Type	Description	Notes
DataResults	*<object>*	Object representing a 2-dimensional array of query data and totals.	
DataResults.data	*<array of <array of <string>>>*	Two-dimensional data array.	
DataResults.data[Row]	*<array of <string>>*	One row in the list of rows.	Optional
DataResults.data[Row][item]	*<string>*	One value datum.	Optional
DataResults.data_size	*<number>*	Number of rows in the data array.	Optional
DataResults.totals	*<array of <string>>*	Object representing a row of total values (totals).	Optional
DataResults.totals[item]	*<string>*	One total datum.	Optional

---

Reporting: List group bys

Get a list of reporting summarizations (group bys).

```
GET https://{device}/api/profiler/1.0/reporting/group_bys
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "hos",
    "name": "host"
  },
  {
    "id": "hop",
    "name": "host pair"
  },
  {
    "id": "gro",
    "name": "host group"
  },
  {
    "id": "gpp",
    "name": "host group pair"
  }
]
```

Property Name	Type	Description	Notes
GroupBys	*<array of <object>>*	List of reporting summarizations (group-bys).	
GroupBys[GroupBy]	*<object>*	Object representing a group by.	Optional
GroupBys[GroupBy].id	*<string>*	ID of a group by. To be used in the API.	
GroupBys[GroupBy].name	*<string>*	Human-readable name of a group by.	

Reporting: List columns

Get a list of columns.

```
GET https://{device}/api/profiler/1.0/reporting/columns?metric={string}&statistic={string}&severity={string}&role={string}&category={string}&group_by={string}&direction={string}&area={string}&centricity={string}&unit={string}&rate={string}&realm={string}
```

Authorization

This request requires authorization.

Parameters

Property Name	Type	Description	Notes
metric	*<string>*	Filter the list of columns by metric.	Optional
statistic	*<string>*	Filter the list of columns by statistic.	Optional
severity	*<string>*	Filter the list of columns by severity.	Optional
role	*<string>*	Filter the list of columns by role.	Optional
category	*<string>*	Filter the list of columns by category.	Optional
group_by	*<string>*	Filter the list of columns by group by.	Optional
direction	*<string>*	Filter the list of columns by direction.	Optional
area	*<string>*	Filter the list of columns by area.	Optional
centricity	*<string>*	Filter the list of columns by centricity.	Optional
unit	*<string>*	Filter the list of columns by unit.	Optional
rate	*<string>*	Filter the list of columns by rate.	Optional
realm	*<string>*	Filter the list of columns by realm.	Optional

Response Body

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

JSON

```
[
  {
    "metric": string,
    "cli_srv": string,
    "comparison_parameter": string,
    "internal": string,
    "id": number,
    "strid": string,
    "statistic": string,
    "severity": string,
    "role": string,
    "category": string,
    "name": string,
    "comparison": string,
    "sortable": string,
    "type": string,
    "direction": string,
    "available": string,
    "context": string,
    "area": string,
    "has_others": string,
    "unit": string,
    "name_type": string,
    "rate": string
  }
]
```

Example:

```
[
  {
    "strid": "ID_TOTAL_BYTES",
    "metric": "net_bw",
    "rate": "count",
    "statistic": "total",
    "id": 30,
    "unit": "bytes",
    "category": "data",
    "severity": "none",
    "area": "none",
    "internal": false,
    "role": "none",
    "cli_srv": "none",
    "type": "int",
    "available": false,
    "direction": "none",
    "comparison": "none",
    "sortable": true,
    "name": "Total Bytes",
    "comparison_parameter": "",
    "has_others": false,
    "context": false,
    "name_type": "colname_parts"
  },
  {
    "strid": "ID_TOTAL_PKTS",
    "metric": "net_bw",
    "rate": "count",
    "statistic": "total",
    "id": 31,
    "unit": "pkts",
    "category": "data",
    "severity": "none",
    "area": "none",
    "internal": false,
    "role": "none",
    "cli_srv": "none",
    "type": "int",
    "available": false,
    "direction": "none",
    "comparison": "none",
    "sortable": true,
    "name": "Total Packets",
    "comparison_parameter": "",
    "has_others": false,
    "context": false,
    "name_type": "colname_parts"
  }
]
```

Property Name	Type	Description	Notes
Columns	*<array of <object>>*	List of reporting query columns.	
Columns[Column]	*<object>*	A column for reporting query.	Optional

Columns[Column].metric	*<string>*	Column 'metric'. See 'reporting/metrics'.	
Columns[Column].cli_srv	*<string>*	Text flag indicating if the column is for the clients or servers.	
Columns[Column].comparison_parameter	*<string>*	Parameter for column comparison.	
Columns[Column].internal	*<string>*	Boolean flag indicating if the column is internal to the system.	
Columns[Column].id	*<number>*	System ID for the column. Used in the API.	
Columns[Column].strid	*<string>*	String ID for the column. Not used by the API, but easier for the human user to see.	
Columns[Column].statistic	*<string>*	Column 'statistic'. See 'reporting/statistics'.	
Columns[Column].severity	*<string>*	Column 'severity'. See 'reporting/severities.	
Columns[Column].role	*<string>*	Column 'role'. See 'reporting/roles'.	
Columns[Column].category	*<string>*	Column 'category'. See 'reporting/categories'.	
Columns[Column].name	*<string>*	Column name. Format used for column names is similar to the format used for column data.	
Columns[Column].comparison	*<string>*	Column 'comparison'. See 'reporting/comparisons'.	
Columns[Column].sortable	*<string>*	Boolean flag indicating if this data can be sorted on this column when running the template.	
Columns[Column].type	*<string>*	Type of the column data. See 'reporting/types'.	
Columns[Column].direction	*<string>*	Column 'direction'. See 'reporting/directions'.	
Columns[Column].available	*<string>*	Boolean flag indicating that the data for the column is available without the need to re-run the template.	
Columns[Column].context	*<string>*	Internal flag used for formatting certain kinds of data.	
Columns[Column].area	*<string>*	Column 'area'. See 'reporting/area'.	
Columns[Column].has_others	*<string>*	Boolean flag indicating if the column's 'other' row can be computed.	
Columns[Column].unit	*<string>*	Column 'unit'. See 'reporting/units'.	
Columns[Column].name_type	*<string>*	Type of the column name. See 'reporting/types'.	
Columns[Column].rate	*<string>*	Column 'rate'. See 'reporting/rates'.	

Reporting: Get report

Get information for report. Includes progress information for running reports.

```
GET https://{device}/api/profiler/1.0/reporting/reports/{report_id}
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
{
  "run_time": number,
  "error_text": string,
  "remaining_seconds": number,
  "saved": string,
  "id": number,
  "status": string,
  "percent": number,
  "user_id": number,
  "size": number,
  "name": string,
  "template_id": number
}

Example:
{
  "status": "completed",
  "user_id": 1,
  "name": "Host Information Report",
  "percent": 100,
  "id": 1001,
  "remaining_seconds": 0,
  "run_time": 1352494550,
  "saved": true,
  "template_id": 952,
  "error_text": "",
  "size": 140
}
```

Property Name	Type	Description	Notes
ReportInfo	*<object>*	Object representing report information.	
ReportInfo.run_time	*<number>*	Time when the report was run (Unix time).	
ReportInfo.error_text	*<string>*	A report can be completed with an error. Error message may provide more detailed info.	Optional
ReportInfo.remaining_seconds	*<number>*	Number of seconds remaining to run the report. Even if this number is 0, the report may not yet be completed, so check 'status' to make sure what the status is.	
ReportInfo.saved	*<string>*	Boolean flag indicating if the report was saved.	
ReportInfo.id	*<number>*	ID of the report. To be used in the API.	
ReportInfo.status	*<string>*	Status of the report.	Values: completed, running, waiting
ReportInfo.percent	*<number>*	Progress of the report represented by percentage of report completion.	
ReportInfo.user_id	*<number>*	ID of the user who owns the report.	
ReportInfo.size	*<number>*	Size of the report in kilobytes.	
ReportInfo.name	*<string>*	Name of the report. Could be given by a user or automatically generated by the system.	Optional
ReportInfo.template_id	*<number>*	ID of the template that the report is based on.	

Reporting: List templates

Get a list of templates.

```
GET https://{device}/api/profiler/1.0/reporting/templates
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "schedule_type": string,
    "id": number,
    "scheduled": string,
    "user_id": number,
    "name": string,
    "next_run": number
  }
]

Example:
[]
```

Property Name	Type	Description	Notes
ReportTemplates_1_1	*<array of <object>>*	List of templates available on the system.	
ReportTemplates_1_1[ReportTemplate_1_1]	*<object>*	One template in the list of templates.	Optional
ReportTemplates_1_1[ReportTemplate_1_1]. schedule_type	*<string>*	Type of template scheduling.	Optional; Values: Once, Hourly, Daily, Weekly, Monthly, Quarterly
ReportTemplates_1_1[ReportTemplate_1_1]. id	*<number>*	ID of the template.	
ReportTemplates_1_1[ReportTemplate_1_1]. scheduled	*<string>*	Flag indicating that the template is scheduled.	
ReportTemplates_1_1[ReportTemplate_1_1]. user_id	*<number>*	ID of the user who owns the template.	Optional
ReportTemplates_1_1[ReportTemplate_1_1]. name	*<string>*	Human-readable name of the template.	
ReportTemplates_1_1[ReportTemplate_1_1]. next_run	*<number>*	Next run time for the template if the template is scheduled to run.	Optional

Reporting: Get report config

Get configuration information for one report. Includes criteria, layout and GUI attributes.

```
GET https://{device}/api/profiler/1.0/reporting/reports/{report_id}/config
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
{
  "criteria": {
    "traffic_expression": string,
    "time_frame": {
      "resolution": string,
      "end": number,
      "expression": string,
      "start": number
    },
    "query": {
      "columns": [
        number
      ],
      "role": string,
      "group_by": string,
      "host_group_type": string,
      "direction": string,
      "sort_column": number,
      "area": string,
      "centricity": string,
      "realm": string
    },
    "deprecated": {
      [prop]: string
    }
  },
  "attributes": {
    [prop]: string
  },
```

```
  "sections": [
    {
      "widgets": [
        {
          "query_id": string,
          "id": string,
          "type": string,
          "attributes": {
            [prop]: string
          }
        }
      ],
      "layout": [
        {
          "items": [
            {
              "id": string
            }
          ],
          "wrappable": string,
          "full_width": string,
          "item_spacing": string,
          "line_spacing": string
        }
      ],
      "attributes": {
        [prop]: string
      }
    }
  ]
}
```

Example:

```
{
  "attributes": {
    "title": "Report name"
  },
  "sections": [
    {
      "widgets": [
        {
          "type": "table",
          "query_id": "0:sum_hos_non_non_non_non_non_non_non_33_d_0",
          "id": "sum_hos_non_non_non_non_non_non_non_tbl_0_0",
          "attributes": {
            "page_size": "20",
            "sort_col": "33",
            "col_order": "6,33,34"
          }
        }
      ],
      "attributes": {
        "sort_desc": "1"
      },
      "layout": [
        {
          "items": [
            {
              "id": "sum_hos_non_non_non_non_non_non_non_tbl_0_0"
            }
          ]
        }
      ]
    }
  ],
  "criteria": {
    "time_frame": {
      "start": 1352319891,
      "end": 1352320191
    },
    "query": {
      "realm": "traffic_summary",
      "sort_column": 33,
      "centricity": "hos",
      "group_by": "hos",
      "columns": [
        6,
        33,
        34
      ]
    }
  }
}
```

Property Name	Type	Description	Notes

ReportConfig	*<object>*	Object representing report configuration.	
*ReportConfig.*criteria	*<object>*	Report criteria.	
*ReportConfig.*criteria.traffic_expression	*<string>*	Traffic expression.	Optional
*ReportConfig.*criteria.time_frame	*<object>*	Time frame object.	Optional
*ReportConfig.*criteria.time_frame.resolution	*<string>*	Report data resolution. It can be one of: 1min, 15min, hour, 6hour, day, week, month. If not specified a resolution will automatically be selected based on time frame of the report.	Optional
*ReportConfig.*criteria.time_frame.end	*<number>*	Report end time (unix time).	Optional
*ReportConfig.*criteria.time_frame.expression	*<string>*	Traffic expression.	Optional
*ReportConfig.*criteria.time_frame.start	*<number>*	Report start time (unix time).	Optional
*ReportConfig.*criteria.query	*<object>*	Query object.	Optional
*ReportConfig.*criteria.query.columns	*<array of <number>>*	Query columns. Can be many of GET /reporting/columns.	Optional
*ReportConfig.*criteria.query.columns[item]	*<number>*	Query column.	Optional
*ReportConfig.*criteria.query.role	*<string>*	Query role. Can be one of /reporting/roles.	Optional
*ReportConfig.*criteria.query.group_by	*<string>*	Query group_by. Can be one of GET /reporting/group_bys.	Optional
*ReportConfig.*criteria.query.host_group_type	*<string>*	Query host group type. Required for "host group (gro)" "host group pairs (gpp)" and "host group pairs with ports (gpr)" queries.	Optional
*ReportConfig.*criteria.query.direction	*<string>*	Query direction. Can be one of GET /reporting/directions.	Optional
*ReportConfig.*criteria.query.sort_column	*<number>*	Query sort column. Can be one of GET /reporting/columns.	Optional
*ReportConfig.*criteria.query.area	*<string>*	Query area. Can be one of GET /reporting/areas.	Optional
*ReportConfig.*criteria.query.centricity	*<string>*	Query centricity. Can be one of GET /reporting/centricities.	Optional
*ReportConfig.*criteria.query.realm	*<string>*	Query realm. Can be one of GET /reporting/realms.	
*ReportConfig.*criteria.deprecated	*<object>*	Map with legacy criteria attributes that will not be supported soon.	Optional
*ReportConfig.*criteria.deprecated[prop]	*<string>*	ReportDeprecatedFilters map value.	Optional
*ReportConfig.*attributes	*<object>*	Report attributes.	
*ReportConfig.*attributes[prop]	*<string>*	Report attributes value.	Optional
*ReportConfig.*sections	*<array of <object>>*	Report sections.	
*ReportConfig.*sections[ReportSection]	*<object>*	One section of a report.	Optional
*ReportConfig.*sections[ReportSection].widgets	*<array of <object>>*	List of section widgets.	
*ReportConfig.*sections[ReportSection].widgets[ReportWidget]	*<object>*	One widget from a list of widgets.	Optional
*ReportConfig.*sections[ReportSection].widgets[ReportWidget].query_id	*<string>*	Query ID for the query that the widget is based on.	
*ReportConfig.*sections[ReportSection].widgets[ReportWidget].id	*<string>*	Widget ID used to reference a widget from the API.	
*ReportConfig.*sections[ReportSection].widgets[ReportWidget].type	*<string>*	Visual type of the widget.	
*ReportConfig.*sections[ReportSection].widgets[ReportWidget].attributes	*<object>*	Widget attributes.	
*ReportConfig.*sections[ReportSection].widgets[ReportWidget].attributes[prop]	*<string>*	Attribute value in the map.	Optional
*ReportConfig.*sections[ReportSection].layout	*<array of <object>>*	Section widget layout.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine]	*<object>*	One horizontal line of widgets.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].items	*<array of <object>>*	List of items (widgets) on the line.	
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].items[ReportLayoutItem]	*<object>*	One item in the list of layout items.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].items[ReportLayoutItem].id	*<string>*	ID of the layout item.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].wrappable	*<string>*	Flag allowing wrapping.	Optional

*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].full_width	*<string>*	Flag representing width of the layout line.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].item_spacing	*<string>*	Item spacing between widgets.	Optional
*ReportConfig.*sections[ReportSection].layout[ReportLayoutLine].line_spacing	*<string>*	Line spacing.	Optional
*ReportConfig.*sections[ReportSection].attributes	*<object>*	Section attributes.	
*ReportConfig.*sections[ReportSection].attributes[prop]	*<string>*	Attribute value in the map.	Optional

Reporting: List areas

Get a list of areas that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/areas
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "wan",
    "name": "wan"
  },
  {
    "id": "lan",
    "name": "lan"
  }
]
```

Property Name	Type	Description	Notes
Areas	*<array of <object>>*	List of areas.	
Areas[Area]	*<object>*	Object representing an area.	Optional
Areas[Area].id	*<string>*	ID of an area. To be used in the API.	
Areas[Area].name	*<string>*	Human-readable name of a area.	

Reporting: List metrics

Get a list of metrics that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/metrics
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "nbw",
    "name": "net bandwidth"
  },
  {
    "id": "nrt",
    "name": "net rtt"
  },
  {
    "id": "rtm",
    "name": "response time"
  }
]
```

Property Name	Type	Description	Notes
Metrics	*<array of <object>>*	List of metrics.	
Metrics[Metric]	*<object>*	Object representing a metric.	Optional
Metrics[Metric].id	*<string>*	ID of a metric. To be used in the API.	
Metrics[Metric].name	*<string>*	Human-readable name of a metric.	

Reporting: List rates

Get a list of rates that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/rates
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]

Example:
[
  {
    "id": "cnt",
    "name": "count"
  },
  {
    "id": "psc",
    "name": "per second"
  }
]
```

Property Name	Type	Description	Notes
Rates	*<array of <object>>*	List of rates.	
Rates[Rate]	*<object>*	Object representing a rate.	Optional
Rates[Rate].id	*<string>*	ID of a rate. To be used in the API.	
Rates[Rate].name	*<string>*	Human-readable name of a rate.	

Reporting: Get template

Reporting: Get template

Get a template.

```
GET https://{device}/api/profiler/1.0/reporting/templates/{template_id}
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
{
  "schedule_type": string,
  "id": number,
  "scheduled": string,
  "user_id": number,
  "name": string,
  "next_run": number
}
```

Property Name	Type	Description	Notes
ReportTemplate_1_1	*<object>*	A template for running reports.	
ReportTemplate_1_1.schedule_type	*<string>*	Type of template scheduling.	Optional; Values: Once, Hourly, Daily, Weekly, Monthly, Quarterly
ReportTemplate_1_1.id	*<number>*	ID of the template.	
ReportTemplate_1_1.scheduled	*<string>*	Flag indicating that the template is scheduled.	
ReportTemplate_1_1.user_id	*<number>*	ID of the user who owns the template.	Optional
ReportTemplate_1_1.name	*<string>*	Human-readable name of the template.	
ReportTemplate_1_1.next_run	*<number>*	Next run time for the template if the template is scheduled to run.	Optional

---

Reporting: Get report view (PDF, CSV)

Get GUI view of a report (PDF, CSV).

```
GET https://{device}/api/profiler/1.0/reporting/reports/{report_id}/view
```

Authorization

This request requires authorization.

Response Body

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

Reporting: List roles

Get a list of roles that this version of the API supports.

```
GET https://{device}/api/profiler/1.0/reporting/roles
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "id": string,
    "name": string
  }
]
```

Example:

```
[
  {
    "id": "cli",
    "name": "client"
  },
  {
    "id": "srv",
    "name": "server"
  }
]
```

Property Name	Type	Description	Notes
Roles	*<array of <object>>*	List of roles.	
Roles[Role]	*<object>*	Object representing a roles.	Optional
Roles[Role].id	*<string>*	ID of a role. To be used in the API.	
Roles[Role].name	*<string>*	Human-readable name of a role.	

Reporting: Get report queries

Get information for all queries run as part of this report. Each query has a list of columns.

```
GET https://{device}/api/profiler/1.0/reporting/reports/{report_id}/queries
```

Authorization

This request requires authorization.

Response Body

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

JSON

```
[
  {
    "metric": string,
    "actual_log": string,
    "columns": [
      {
        "metric": string,
        "cli_srv": string,
        "comparison_parameter": string,
        "internal": string,
        "id": number,
        "strid": string,
        "statistic": string,
        "severity": string,
        "role": string,
        "category": string,
        "name": string,
        "comparison": string,
        "sortable": string,
        "type": string,
        "direction": string,
        "available": string,
        "context": string,
        "area": string,
        "has_others": string,
        "unit": string,
        "name_type": string,
        "rate": string
      }
    ],
    "id": string,
    "statistic": string,
    "role": string,
    "group_by": string,
    "actual_t0": number,
    "parent_id": string,
```

```
  "parent_id": string,
  "actual_t1": number,
  "type": string,
  "sort_col": number,
  "direction": string,
  "sort_desc": string,
  "area": string,
  "unit": string,
  "rate": string
 }
]
```


Example:


```
[
 {
  "direction": "none",
  "actual_log": "flow",
  "actual_t0": 1352319840,
  "sort_desc": true,
  "area": "none",
  "metric": "none",
  "sort_col": 33,
  "parent_id": "",
  "rate": "none",
  "group_by": "hos",
  "role": "none",
  "columns": [
   {
    "strid": "ID_AVG_BYTES",
    "metric": "net_bw",
    "rate": "persec",
    "statistic": "avg",
    "id": 33,
    "unit": "bytes",
    "category": "data",
    "severity": "none",
    "area": "none",
    "internal": false,
    "role": "none",
    "cli_srv": "none",
    "type": "float",
    "available": true,
    "direction": "none",
    "comparison": "none",
    "sortable": true,
    "name": "Avg Bytes/s",
    "comparison_parameter": "",
    "has_others": false,
    "context": false,
    "name_type": "colname_parts"
   },
   {
    "strid": "ID_AVG_BYTES_RTX",
    "metric": "rtx",
    "rate": "persec",
    "statistic": "avg",
    "id": 391,
    "unit": "bytes",
    "category": "data",
    "severity": "none",
    "area": "none",
    "internal": false,
    "role": "none",
    "cli_srv": "none",
    "type": "float",
    "available": false,
    "direction": "none",
    "comparison": "none",
    "sortable": true,
    "name": "Avg Retrans Bytes/s",
    "comparison_parameter": "",
    "has_others": false,
    "context": false,
    "name_type": "colname_parts"
   }
  ],
  "statistic": "none",
  "type": "summary",
  "id": "0:sum_hos_non_non_non_non_non_non_non_33_d_0",
  "unit": "none",
  "actual_t1": 1352320200
 }
]
```


Property Name	Type	Description	Notes

Queries	*<array of <object>>*	List of queries. Query is one tabular unit of report data.	
Queries[Query]	*<object>*	A query.	Optional
Queries[Query].metric	*<string>*	Query 'metric'. See 'reporting/metrics'.	
Queries[Query].actual_log	*<string>*	Type of data log file that was used to get data for the query.	
Queries[Query].columns	*<array of <object>>*	List of columns that consitute the query. See 'reporting/columns'.	
Queries[Query].columns[Column]	*<object>*	A column for reporting query.	Optional
Queries[Query].columns[Column].metric	*<string>*	Column 'metric'. See 'reporting/metrics'.	
Queries[Query].columns[Column].cli_srv	*<string>*	Text flag indicating if the column is for the clients or servers.	
Queries[Query].columns[Column].comparison_parameter	*<string>*	Parameter for column comparison.	
Queries[Query].columns[Column].internal	*<string>*	Boolean flag indicating if the column is internal to the system.	
Queries[Query].columns[Column].id	*<number>*	System ID for the column. Used in the API.	
Queries[Query].columns[Column].strid	*<string>*	String ID for the column. Not used by the API, but easier for the human user to see.	
Queries[Query].columns[Column].statistic	*<string>*	Column 'statistic'. See 'reporting/statistics'.	
Queries[Query].columns[Column].severity	*<string>*	Column 'severity'. See 'reporting/severities.	
Queries[Query].columns[Column].role	*<string>*	Column 'role'. See 'reporting/roles'.	
Queries[Query].columns[Column].category	*<string>*	Column 'category'. See 'reporting/categories'.	
Queries[Query].columns[Column].name	*<string>*	Column name. Format used for column names is similar to the format used for column data.	
Queries[Query].columns[Column].comparison	*<string>*	Column 'comparison'. See 'reporting/comparisons'.	
Queries[Query].columns[Column].sortable	*<string>*	Boolean flag indicating if this data can be sorted on this column when running the template.	
Queries[Query].columns[Column].type	*<string>*	Type of the column data. See 'reporting/types'.	
Queries[Query].columns[Column].direction	*<string>*	Column 'direction'. See 'reporting/directions'.	
Queries[Query].columns[Column].available	*<string>*	Boolean flag indicating that the data for the column is available without the need to re-run the template.	
Queries[Query].columns[Column].context	*<string>*	Internal flag used for formatting certain kinds of data.	
Queries[Query].columns[Column].area	*<string>*	Column 'area'. See 'reporting/area'.	
Queries[Query].columns[Column].has_others	*<string>*	Boolean flag indicating if the column's 'other' row can be computed.	
Queries[Query].columns[Column].unit	*<string>*	Column 'unit'. See 'reporting/units'.	
Queries[Query].columns[Column].name_type	*<string>*	Type of the column name. See 'reporting/types'.	
Queries[Query].columns[Column].rate	*<string>*	Column 'rate'. See 'reporting/rates'.	
Queries[Query].id	*<string>*	ID for the query. Used in the API.	
Queries[Query].statistic	*<string>*	Query 'statistic'. See 'reporting/statistics'.	
Queries[Query].role	*<string>*	Query 'role'. See 'reporting/roles.'.	
Queries[Query].group_by	*<string>*	Grouping of data in the query. See 'reporting/group_bys'.	
Queries[Query].actual_t0	*<number>*	Actual start time for data in the query. This could be different from the requested start time because of time interval snapping and other similar features.	
Queries[Query].parent_id	*<string>*	Query ID of the query that preceeded this query and influenced data selection for it. For example, if one runs a query that returns time-series data for top 10 protocols in the network, the first query that would need to run is the one to pick top 10 protocols. That query would be the parent one to the follow-up query to get time-series data for those selected 10 protocols.	
Queries[Query].actual_t1	*<number>*	Actual end time for the data in the query. See 'actual_t0' for more detail.	
Queries[Query].type	*<string>*	Internal value. Reserved.	
Queries[Query].sort_col	*<number>*	ID of that column that was used to sort the query when it ran.	
Queries[Query].direction	*<string>*	Query 'direction. See 'reporting/directions'.	
Queries[Query].sort_desc	*<string>*	Boolean flag indicating if the sorting was in the descending order.	

Queries[Query].area	*<string>*	Query 'area'. See 'reporting/areas'.	
Queries[Query].unit	*<string>*	Query 'unit'. See 'reporting/units'.	
Queries[Query].rate	*<string>*	Query 'rate'. See 'reporting/rates.	

Reporting: Create report

Generate a new report.

POST https://{device}/api/profiler/1.0/reporting/reports

Authorization

This request requires authorization.

Request Body

Provide a request body with the following structure:

JSON

```
{
 "criteria": {
  "traffic_expression": string,
  "time_frame": {
   "resolution": string,
   "end": number,
   "expression": string,
   "start": number
  },
  "query": {
   "columns": [
    number
   ],
   "role": string,
   "group_by": string,
   "host_group_type": string,
   "direction": string,
   "sort_column": number,
   "area": string,
   "centricity": string,
   "realm": string
  },
  "deprecated": {
   [prop]: string
  }
 },
 "name": string,
 "template_id": number
}

Example:
{
 "criteria": {
  "traffic_expression": "app WEB",
  "time_frame": {
   "start": 1352314764,
   "end": 1352315064
  },
  "query": {
   "realm": "traffic_summary",
   "sort_column": 33,
   "group_by": "hos",
   "columns": [
    6,
    33,
    34
   ]
  }
 },
 "template_id": 184,
 "name": "Bytes and packets for the top 20 hosts using application WEB"
}
```

Property Name	Type	Description	Notes
ReportInputs	*<object>*	ReportInputs object.	
*ReportInputs.*criteria	*<object>*	Criteria neeed to run the report.	Optional
*ReportInputs.*criteria.traffic_expression	*<string>*	Traffic expression.	Optional

*ReportInputs.*criteria.time_frame	*<object>*	Time frame object.	Optional
*ReportInputs.*criteria.time_frame.resolution	*<string>*	Report data resolution. It can be one of: 1min, 15min, hour, 6hour, day, week, month. If not specified a resolution will automatically be selected based on time frame of the report.	Optional
*ReportInputs.*criteria.time_frame.end	*<number>*	Report end time (unix time).	Optional
*ReportInputs.*criteria.time_frame.expression	*<string>*	Traffic expression.	Optional
*ReportInputs.*criteria.time_frame.start	*<number>*	Report start time (unix time).	Optional
*ReportInputs.*criteria.query	*<object>*	Query object.	Optional
*ReportInputs.*criteria.query.columns	*<array of <number>>*	Query columns. Can be many of GET /reporting/columns.	Optional
*ReportInputs.*criteria.query.columns[item]	*<number>*	Query column.	Optional
*ReportInputs.*criteria.query.role	*<string>*	Query role. Can be one of /reporting/roles.	Optional
*ReportInputs.*criteria.query.group_by	*<string>*	Query group_by. Can be one of GET /reporting/group_bys.	Optional
*ReportInputs.*criteria.query.host_group_type	*<string>*	Query host group type. Required for "host group (gro)" "host group pairs (gpp)" and "host group pairs with ports (gpr)" queries.	Optional
*ReportInputs.*criteria.query.direction	*<string>*	Query direction. Can be one of GET /reporting/directions.	Optional
*ReportInputs.*criteria.query.sort_column	*<number>*	Query sort column. Can be one of GET /reporting/columns.	Optional
*ReportInputs.*criteria.query.area	*<string>*	Query area. Can be one of GET /reporting/areas.	Optional
*ReportInputs.*criteria.query.centricity	*<string>*	Query centricity. Can be one of GET /reporting/centricities.	Optional
*ReportInputs.*criteria.query.realm	*<string>*	Query realm. Can be one of GET /reporting/realms.	
*ReportInputs.*criteria.deprecated	*<object>*	Map with legacy criteria attributes that will not be supported soon.	Optional
*ReportInputs.*criteria.deprecated[prop]	*<string>*	ReportDeprecatedFilters map value.	Optional
*ReportInputs.*name	*<string>*	Report name.	Optional
*ReportInputs.*template_id	*<number>*	Template ID. Can be one of GET /reporting/templates.	

Response Body

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

JSON

```
{
  "run_time": number,
  "error_text": string,
  "remaining_seconds": number,
  "saved": string,
  "id": number,
  "status": string,
  "percent": number,
  "user_id": number,
  "size": number,
  "name": string,
  "template_id": number
}

Example:
{
  "status": "completed",
  "user_id": 1,
  "name": "Host Information Report",
  "percent": 100,
  "id": 1001,
  "remaining_seconds": 0,
  "run_time": 1352494550,
  "saved": true,
  "template_id": 952,
  "error_text": "",
  "size": 140
}
```

Property Name	Type	Description	Notes
ReportInfo	*<object>*	Object representing report information.	
*ReportInfo.*run_time	*<number>*	Time when the report was run (Unix time).	

ReportInfo.error_text	*<string>*	A report can be completed with an error. Error message may provide more detailed info.	Optional
ReportInfo.remaining_seconds	*<number>*	Number of seconds remaining to run the report. Even if this number is 0, the report may not yet be completed, so check 'status' to make sure what the status is.	
ReportInfo.saved	*<string>*	Boolean flag indicating if the report was saved.	
ReportInfo.id	*<number>*	ID of the report. To be used in the API.	
ReportInfo.status	*<string>*	Status of the report.	Values: completed, running, waiting
ReportInfo.percent	*<number>*	Progress of the report represented by percentage of report completion.	
ReportInfo.user_id	*<number>*	ID of the user who owns the report.	
ReportInfo.size	*<number>*	Size of the report in kilobytes.	
ReportInfo.name	*<string>*	Name of the report. Could be given by a user or automatically generated by the system.	Optional
ReportInfo.template_id	*<number>*	ID of the template that the report is based on.	

Users: List users

Get a list of user accounts.

```
GET https://{device}/api/profiler/1.0/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/profiler/1.0/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/profiler/1.0/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/profiler/1.0/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

Do not provide a request body.

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/profiler/1.0/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.