

Hitachi Ops Center Analyzer Detail View

REST API Reference Guide

This document provides detailed information on the Ops Center Analyzer detail view REST API.

© 2025 Hitachi Vantara, Ltd. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including copying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., Hitachi Vantara, Ltd., or Hitachi Vantara Corporation (collectively "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara LLC at https://support.hitachivantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara LLC.

By using this software, you agree that you are responsible for:

1. Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals; and
2. Verifying that your data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

Hitachi and Lumada are trademarks or registered trademarks of Hitachi, Ltd., in the United States and other countries.

AlX, DB2, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, FlashCopy, GDPS, HyperSwap, IBM, OS/390, PowerHA, PowerPC, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z14, z15, z16, z/VM, and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, Microsoft Edge, the Microsoft corporate logo, the Microsoft Edge logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or website are properties of their respective owners.

Copyright and license information for third-party and open source software used in Hitachi Vantara products can be found in the product documentation, at <https://www.hitachivantara.com/en-us/company/legal.html>.

Contents

Preface.....	6
Product version.....	6
Accessing product documentation.....	6
Comments.....	6
Chapter 1: Overview.....	7
Analyzer detail view database schema.....	8
Common specification of the API functions.....	12
Communication protocols and port numbers.....	12
Request and response formats.....	12
Request format of the API.....	12
API request headers.....	12
URI format.....	13
API request URL.....	14
API request length restriction.....	15
API request process flow.....	15
Response format of the API.....	22
Response headers descriptions.....	22
HTTP status codes descriptions.....	23
Timestamps and time zones.....	24
Authorization for APIs.....	24
Authorization for APIs using the Analyzer detail view local and Active Directory users.....	24
Authorization for APIs using the Ops Center Portal users.....	25
API parameters.....	26
Chapter 2: Data subset API resources.....	27
Creating a data subset.....	27
Getting information about the data subset.....	31
Chapter 3: Attribute definitions.....	36
Creating a custom attribute definition.....	36
Getting information about an attribute definition.....	44
Updating attribute definition information.....	50
Deleting the custom attribute definition.....	56

Chapter 4: Resource definitions.....	62
Creating a custom resource definition.....	62
Getting information about a resource definition.....	71
Updating resource definition information.....	76
Deleting a resource definition.....	87
Chapter 5: Resource data.....	93
Creating or updating resource data.....	93
Locking the request at resource level.....	103
Time series data points.....	104
Deleting resource data.....	105
Getting resource data information.....	111
Deactivating the resource data.....	118
Chapter 6: Dataset API resources.....	125
Getting information about a dataset.....	125
Deleting the dataset.....	128
Chapter 7: Request API resources.....	132
Getting the request status.....	132
Getting an audit log.....	135
Stopping an API request.....	142
Getting the Analyzer detail view server information.....	144
Chapter 8: Database API resources.....	148
Configuring database and API properties.....	148
Editing database properties.....	148
Editing API properties.....	151
Configuring the number of worker threads for the application web server (Jetty).....	155
Increasing the maximum number of open file handles.....	155
Viewing the existing schema using REST APIs.....	156
Chapter 9: Query API resources.....	159
Running the MQL query.....	159
Best practices for query APIs.....	169
Chapter 10: Alert API resources.....	172
Alert API design.....	172
Creating an alert definition.....	174
Getting information about alert definitions.....	186
Getting information about alert condition violations.....	194
Subscribing to push notifications.....	199

Websocket.....	202
Subscribing to alert notifications.....	202
Unsubscribing an alerts notification.....	207
Pausing an alert definition evaluation.....	209
Restarting an alert definition evaluation.....	210
Deleting an alert definition.....	212
Updating an alert threshold.....	214
Getting the status of an alert request.....	218
Getting an alert audit log.....	218
Evaluating alert violations.....	223
Selecting the data interval.....	223
Data hole consideration in the alert violation computation.....	224
Editing the threshold value.....	225
Chapter 11: User, license, and product URL API resources.....	227
Access rights for API functions.....	227
API authentication.....	228
License key authentication.....	228
User authentication token.....	228
Authentication details for API functions.....	228
API request headers.....	229
User management API design.....	229
Creating a user.....	229
Updating specific user information.....	236
Changing a password.....	239
Deleting a specific user from the Analyzer detail view.....	242
Registering a license.....	244
Getting the URL of the Analyzer detail view.....	247
Appendix A: API server response error codes and messages.....	250
API server response error codes and messages.....	250
Alert API response error codes and messages.....	269
Appendix B: References.....	272

Preface

This document provides detailed information on the Analyzer detail view REST API. It also provides an overview of the Analyzer detail view database and discusses API server architecture.

Product version

This document revision applies to Ops Center Analyzer detail view version 11.0.4-00 or later.

Accessing product documentation

Product user documentation is available on: <https://docs.hitachivantara.com>. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Comments

Please send comments to doc.feedback@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Chapter 1: Overview

The Hitachi Ops Center Analyzer detail view (Analyzer detail view) database is a high-performance, size-optimized, and NoSQL database that offers highly efficient storage and retrieval capabilities. It can store time series data at a high granularity, including minute-level data for a long period. You can query this data using an expressive and powerful yet concise query language (MQL). MQL is uniquely designed, which makes it different from the commonly used Structured Query Language (SQL).

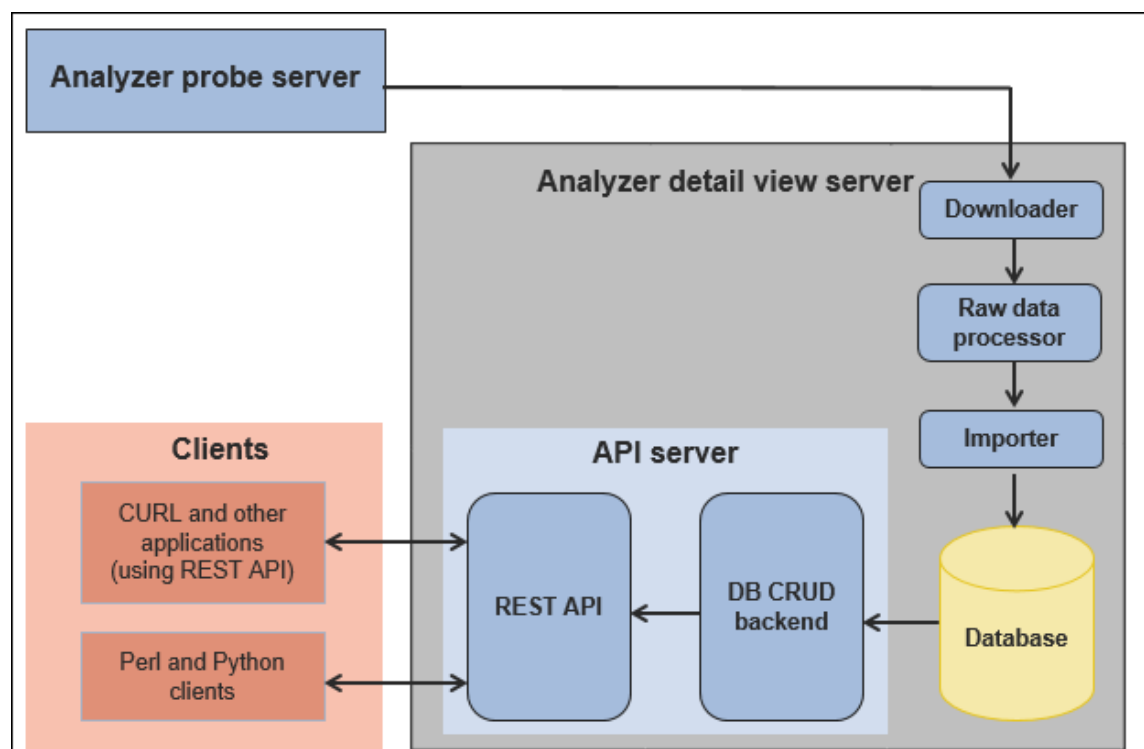
The Analyzer detail view REST API allows you to build an application around the Analyzer detail view database.

The API functions of Analyzer detail view conform to the REST (Representational State Transfer) architecture.

The API requests are handled by the API server, which in turn passes them to the Analyzer detail view database through the CRUD backend (Create, Retrieve, Update, and Delete, the common database operations backend). The CRUD backend uses resource management techniques to efficiently route and process multiple API requests.

The Analyzer probe server collects the data and stores it in the Analyzer detail view database. You can also store the data in the Analyzer detail view database using the third-party application.

The following diagram shows the architecture of the Analyzer detail view REST API server.



The Analyzer detail view REST API architecture consists of the following:

- **Analyzer detail view REST API:** This interface is powered by the Jetty web server and is responsible for accepting incoming HTTP requests, parsing and passing them to the Analyzer detail view CRUD interface.
- **Analyzer detail view CRUD interface:** This interface accepts each request from the REST API and adds it to an in-memory waiting queue.
- **Scheduler:** Continuously monitors the waiting queue and moves runnable requests to the running queue. The scheduler decides on runnable requests among all the waiting requests on the basis of request-level locks. The running queue is bounded by the in-memory queue and its size is equal to the number of threads configured in the request processor.
- **Multi-threaded request processor:** Instead of processing each API request on a separate thread, which can be inefficient on resources, the multi-threaded request processor uses the pool of worker threads to throttle the API requests processing.

The number of worker threads can be tuned to process requests in parallel. Each request from a running queue is picked up by a worker thread. The request processor interacts with the underlying database.

- **Analyzer detail view database:** Consists of schema and data. The schema consists of attribute and resource definitions, whereas the data is of two types: scalar and time series. The Analyzer detail view database has a separate store for schema, scalar, and time series data in the Analyzer detail view virtual file system.

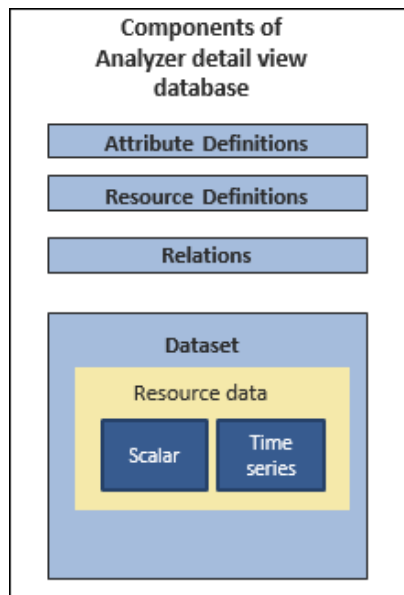
The schema is stored in an unbounded cache. The scalar data is stored in a bounded cache up to a maximum of 1000 elements, where an element is an index of stored scalar value. The least recently used (LRU) policy is used to delete elements from the scalar data cache.

You can view the schema in the Analyzer detail view GUI.

Analyzer detail view database schema

The *schema* in a Analyzer detail view database consists of resource types, attribute definitions, and resource definitions.

The following diagram shows the components of the Analyzer detail view database.



The Analyzer detail view database is organized in dataset and data subsets that consist of scalar and time series data:

Attribute and resource definitions

Attribute and resource definitions together define the schema. The Analyzer detail view server contains the predefined schema, attribute definitions, and resource definitions. You can view this predefined schema using APIs; see [Viewing the existing schema using REST APIs \(on page 156\)](#).

You can also view the existing schema in `Schema Reference.xlsx` provided with this document.

You can also create your own schema.

- **Attribute definition:** Defines how the attribute is shown and consists of the following:
 - **ID:** A unique identifier for attribute definition within a given dataset; for example, name, ipAddress, and cpuUsage.
 - **Name:** Display name for the attribute definition.
 - **Type:** Type can be scalar or time series.
 - **Unit:** Specifies the unit in which the value is stored (percent, MBps, or KBps). For more information, see [Creating a custom resource definition \(on page 62\)](#).
- **Attribute:** A property attached to a resource that provides more information about a resource. Two types of attributes are:
 - **Scalar attributes:** These attributes do not change frequently; for example, host name or IP address of a host, size of a storage volume.
 - **Timeseries attributes:** These attributes are expected to change frequently; for example, CPU usage of a host, number of IOPS occurring on a storage volume.
- **Resource definition:** Defines how the resource is shown and consists of the following:
 - **Type:** A unique identifier for resource definition within a given dataset; for example, LHost represents a Linux Host, and LHostCPU represents the CPUs in a host.
 - **Scalar attributes:** Can exist on that resource; for example, name, ipAddress, capacity, and so on.
 - **Timeseries attributes:** Can exist on that resource; for example, cpuUsage, memUsage, readIOPS, and so on.
 - **Relations:** The relation between resources; for example, LHost is related to LHostCPU, and so on.
- **Resource data:** Resource data consists of either scalar or timeseries data, or both in the Analyzer detail view database. Each resource data instance is associated with a timestamp when it is captured. Resources are real-world entities, such as a host, CPU, storage volume, storage system, NIC card, and virtual machine.
 - **Scalar data:** The resource could have zero or more scalar attributes.
 - **Timeseries data:** The resource could have zero or more time series attributes.

Relations

Relations define resource hierarchies. A resource does not operate independently and is either dependent on another resource or contains more resources. For example, a VM is dependent on the host it runs, while the host contains a number of VMs running on it. Two resources are considered related if one resource depends on another resource or one resource contains the other resource.

Data subset

A data subset is a logical grouping of the resource data. For example, VMware data is associated with one data subset and NetApp data is associated with another data subset.

Dataset

A dataset includes the schema and data. It is similar to the database of the relational database management system (RDBMS).



Note: The Analyzer detail view contains a built-in default dataset called *defaultDs*. All requests must use this default *defaultDs*.

Analyzer detail view and traditional relational database similarities

The Analyzer detail view database is based on concepts that are different from the traditional relational database; however, there are some similarities. The various constructs in relational databases and their conceptual equivalents in the Analyzer detail view database are listed in the following table.

Relational database	Analyzer detail view database
Database	Dataset
Table Name	Resource Type
Table Definition	Resource Definition
Table Column	Attribute Definition
Table Row	Resource Data
Primary Key	Each resource data has an implicit attribute called <i>signature</i> that automatically becomes a Primary Key.
Foreign Key	Relations define hierarchies and are specified using signatures of other (related) resources.
SQL	Uses MQL, specifically designed for operations on the Analyzer detail view database. For more information, see the <i>Hitachi Ops Center Analyzer Detail View Query Language User Guide</i> .
JOIN	Sub-path traversal in MQL query.
WHERE, GROUP BY, HAVING clauses	MQL supports almost all SQL constructs and offers a more expressive and powerful syntax for complex operations as well as filters that support regex-based filtering, aggregation, rollups, grouping, and so on. For more information, see the <i>Hitachi Ops Center Analyzer Detail View Query Language User Guide</i> .

Common specification of the API functions

This section explains the common specification of the API functions provided by Analyzer detail view.

Communication protocols and port numbers

The Analyzer detail view server uses the following ports:

- **HTTPS:** By default, the REST API server listens for HTTPS requests on TCP port 8443.
- **SSL:** The REST API server redirects the HTTP requests to the HTTPS port using the SSL port (default: 8443).

You can change the SSL or HTTPS port numbers. To change these port numbers, you must change the port numbers specified in the definition file, and then open the newly configured port on the Analyzer detail view server for the communication.

Refer to the "Changing the SSL or HTTPS port number of the Analyzer detail view server" section in the *Hitachi Ops Center Analyzer Installation and Configuration Guide* for more information.

Request and response formats

The JSON format is required as the data format for the POST requests and responses. Specify a data format in the request header. Use UTF-8 character encoding for the input and output format.

Request format of the API

A request format contains a URI, method, request line, and request header.

API request headers

The following table describes the applicable request headers for all API requests.

Request header	Description	Mandatory
Authorization	Authorize the request	Yes
Content-Type	This header defines the format of the request body, allowing the server to process it correctly. For example: Content-Type: application/json	Yes (for the POST requests)

Request header	Description	Mandatory
x-waitTime	<p>Time interval for which the request processor in the API server waits before sending a job accepted status or response.</p> <p>Unit: Seconds</p> <p>Default: 10 seconds</p>	No

URI format

Use URIs to specify resources for API functions. A URI contains a basic URI (from <schema> to <version>) and a relative URI (<domain> and followings). Resources are identified with the relative URI (starting from <domain>).

Format of the URI:

```
<scheme>://<host>:<port>/<object-type>[?<query-string>]
```

Or

```
<scheme>://<host>:<port>/<object-type>/actions/[<action-identifier>[/invoke]]
```

In this manual, "<scheme>://<host>:<port>/" is transcribe into baseURL.

The following table describes the URI components.

Item	Description
<scheme>	Protocol. Specify HTTP or HTTPS.
<host>	Host name or IP address of the Analyzer detail view host that is accessible from an API server.
<port>	Port number of the Analyzer detail view host that is accessible from an API server. The default port number is 8443 for HTTPS.
<object-type>	Indicates the resource or service.
<query-string>	Query string that specifies a query parameter. This specifies a condition for reducing or sorting <object-type>. The available query parameters depend on resources and can be in any order.
<action-identifier>	Indicates the action for the resource.



Note: The components and other details of URIs vary for each API function. For more information, see the descriptions of the applicable API function.

API request URL

For HTTP GET requests, API parameters can be specified in a request URL. If an API parameter contains URL reserved characters (such as #), those characters must be replaced with their percent encoding. For example, the # character must be replaced with %23.

Character	Encoding
!	%21
#	%23
\$	%24
&	%26
'	%27
(%28
)	%29
*	%2A
+	%2B

Character	Encoding
,	%2C
/	%2F
:	%3A
;	%3B
=	%3D
?	%3F
@	%40
[%5B
]	%5D

API request length restriction

The Analyzer detail view REST API imposes content length restriction on all API requests. If the API request exceeds the allowed length, the following error is shown: 413 REQUEST ENTITY TOO LARGE.

The restriction on the content length is defined by the following two properties:

`app.db.api.request.max.content.length` and
`app.db.api.request.max.content.length.factor`.

Allowed content length is the smallest of the following two:

- `app.db.api.request.max.content.length`
- JVM memory divided by maximum content length factor (JVM memory / `app.db.api.request.max.content.length.factor`)

API request process flow

This section describes about how the request scheduling and throttling are managed by REST API.

Request scheduling and throttling

The Analyzer detail view REST API request scheduler contains the following limitations while scheduling the request:

- The schema and non-schema read requests are processed in parallel.
- The schema and non-schema read requests and non-schema write requests for a dataset are processed in parallel.
- If the schema write request is in progress, then requests such as the schema read request, non-schema read request, and write request are not processed in parallel with a schema write request.



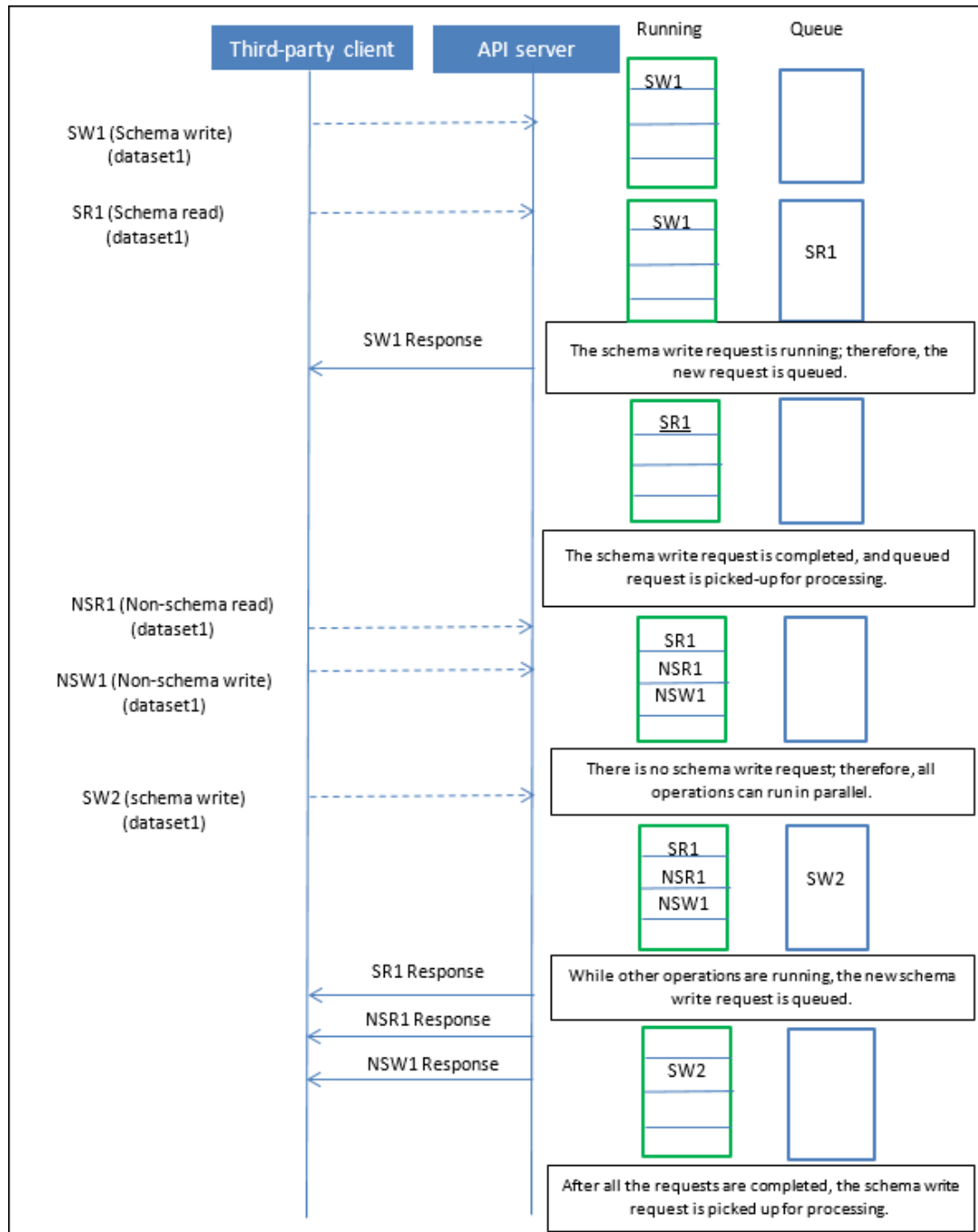
Note: Schema operations include read and write operations on the attribute definition, resource definition, dataset, and data subset. Non-schema operations include read and write operations on resource data and query.

Any request that cannot be processed due to these limitations is queued in the request scheduler. All queued requests are processed by the request processor in first in, first out (FIFO) order.

Instead of processing each API request on a separate thread, which could be inefficient on resources, the request processor uses the pool of worker threads to throttle the API requests processing.

The number of worker threads can be tuned to process number of requests in parallel.

The following diagram explains the request queueing and scheduling.



The following example considers the four worker threads:

1. Initially, all worker threads are available and the queue is empty
2. When the schema write request SW1 is received, it is selected by a worker thread for processing.
3. When another request SR1 is received, it is pushed to the queue. (If a schema write request is in progress, then no other request can run in parallel.)
4. Request SW1 completes and its status is sent in the response message. Request SR1 processing begins.
5. While request SR1 is in progress, additional requests NSR1 and NSW1 are received. These requests can run in parallel; the available worker threads select these request for processing. (All schema read, non-schema read, and non-schema write requests for a dataset can be processed in parallel).
6. While these requests are in progress, another schema write request SW2 is received. This request cannot be processed in parallel and is added to the queue.
7. When all requests are completed, the SW2 request is selected from the queue for processing.

Asynchronous request processing

When an API request is received, Analyzer detail view CRUD interface submits it to the request scheduler and the request is available in the queue. The request is selected by the request processor for processing. The request processor waits for a certain time frame (specified in `x-waitTime`) to complete the processing. If the processing is not completed in the specified time frame, then the request processor returns the status code and request ID to the client (the client used to invoke the REST API calls) indicating that the request is accepted as the asynchronous job and is being processed. In this case, the client should periodically check the status using the request ID provided in the response.

This approach provides the solution to two issues that generally arise with synchronous processing:

- **Timeout:** With synchronous processing, the API server waits for an operation to complete before sending any response back to the client. If the connection drops due to network connectivity issues, API server timeout, or client restart, then the client cannot get back the response to its request. By sending the immediate response with the request ID, the API server ensures that the client can track requests in future irrespective of the health and status of the current connection.
- **Incremental results:** Asynchronous processing enables the client to seek the incremental response from the API server when it is available.

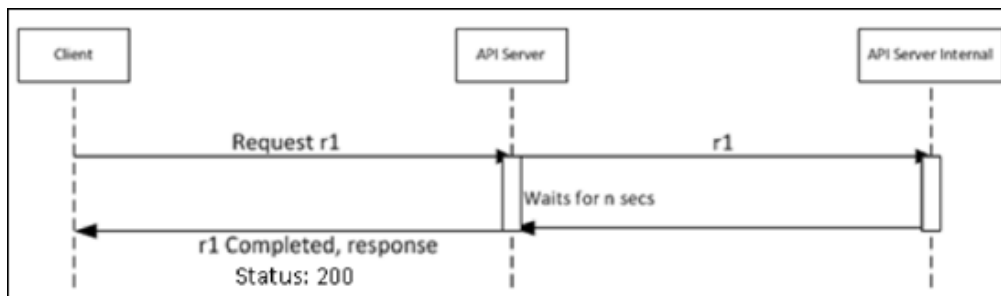
A time frame for which the API request processor waits before sending the `Job Accepted` response can be configured in request by setting the `x-waitTime` header. The value for wait the time frame should be in seconds, and the value 0 makes the request processor return a response immediately. If the `x-waitTime` is not specified or has a negative value, then the default wait time frame of 10 seconds is used.

```
POST action=retrieveDataset&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
x-waitTime:15
```

The API server waits for request completion or 15 seconds, whichever occurs earlier, before sending the response.

Use case 1: Request completes in x-waitTime

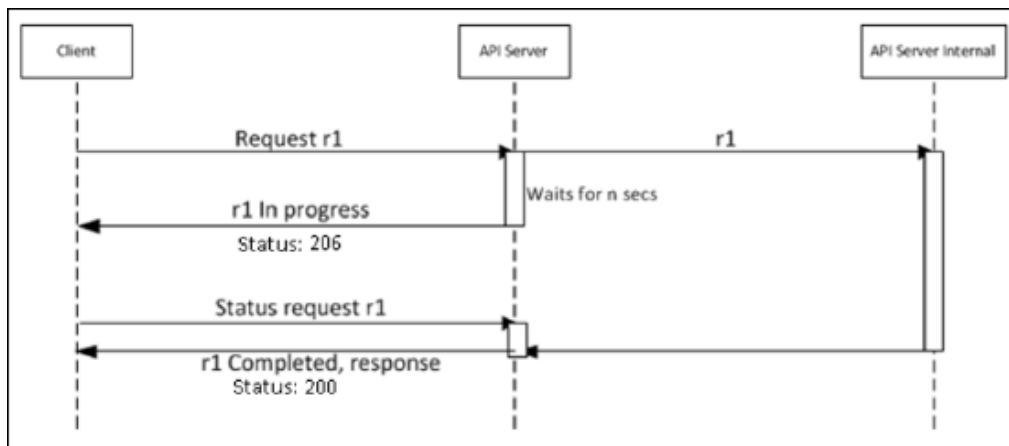
The following diagram shows the asynchronous request processing workflow.



1. The client sends `Request r1` to the API server.
2. The API server schedules the request and waits for `x-waitTime` before sending the response to the client.
3. If the request completes within `x-waitTime`, then the API server sends complete response to the client.

Use case 2: Request is not completed in x-waitTime; client comes back for response

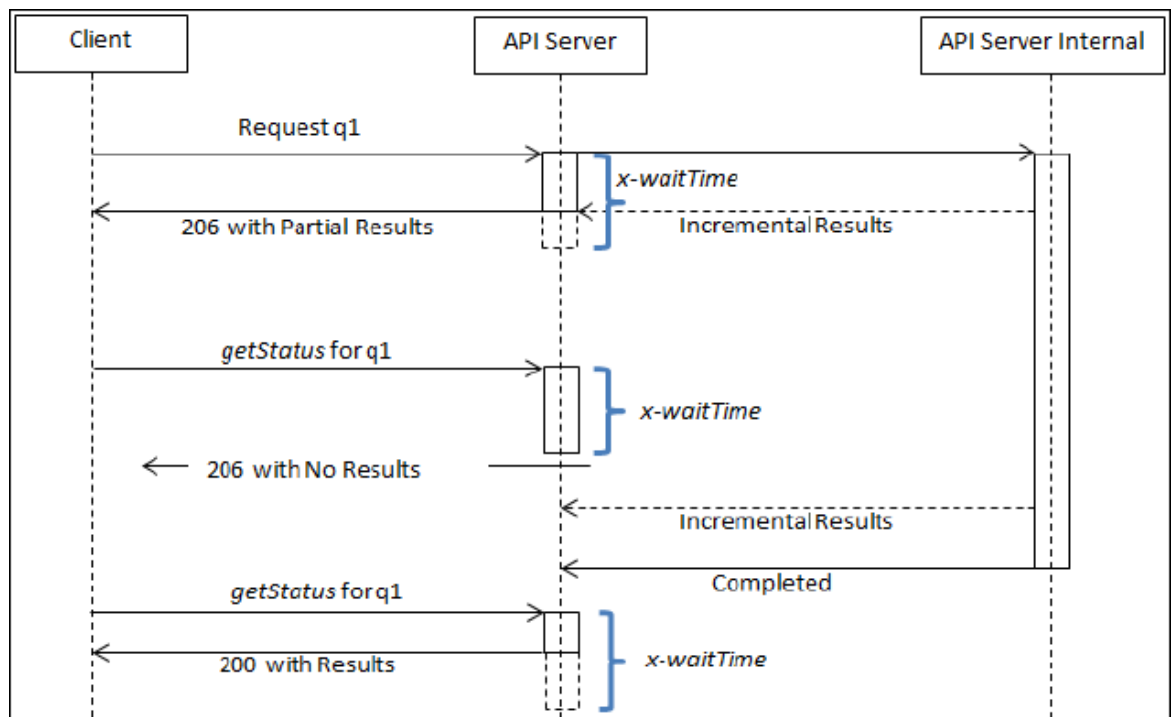
The following diagram shows the asynchronous request processing workflow, when the request is not completed in the `x-waitTime`.



1. The client sends `Request r1` to the API server.
2. The API server schedules the request and waits for `x-waitTime` before sending a response to the client.
3. If request is not completed within `x-waitTime`, then the API server sends a `206 (REQUEST IN PROGRESS)` response code along with the request ID to the client.
4. The client uses the request ID to poll the API server for status of the request.
5. If request is still being processed, then the API server sends the `REQUEST IN PROGRESS` response.
6. If request is completed, then the API server sends the complete response to client along with the request complete response code.
7. For a request in progress, client periodically polls the API server for status until it gets a request completed response.

Use case 2: Asynchronous query request processing flow

The following diagram shows the asynchronous query request processing workflow.



1. The client sends `Query Request q1` at time (t_1) to the API server.
2. The API server schedules the request and responds on two events, whichever occurs first:
 - **Any result is available:** The server responds with either response code 206 with partial results or response code 200 with complete results.
 - **x-waitTime elapses:** The server responds with response code 206 without any result.
3. If the API server responds with response code 206, then the client has to poll the server for more query results using `getStatus` API until it gets complete results.

Caching asynchronous requests

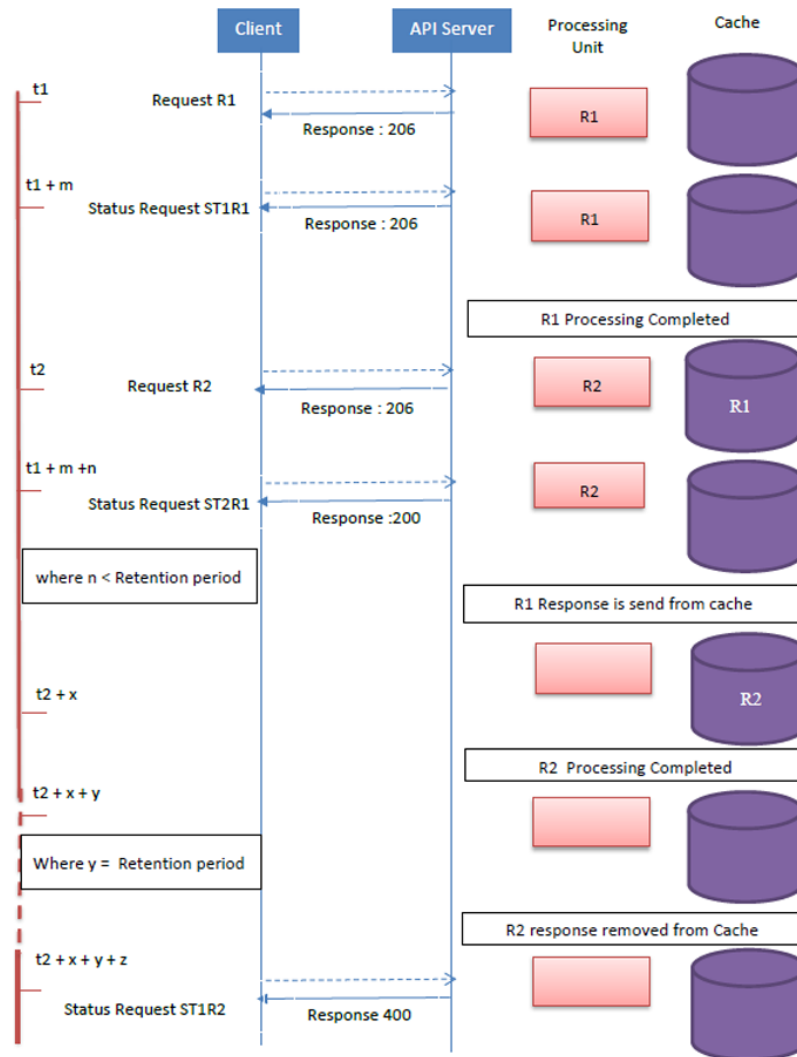
The response of an asynchronous job is cached in the API server. To control the memory consumption, the responses are cached for 3600 seconds (1 hour). The response cached time can be configured through the property:

```
app.db.api.response.cache.time.limit.in.secs.
```

The following retention policy is for a cached response:

- Cached response are saved for 1 hour after the request processing is completed.
- The response are deleted from cache after the response is sent to the client on receiving the `getStatus` request.

A `getStatus` request on deleted response returns the 404 NOT FOUND error.



1. The client sends `Request R1` at time (t_1) to the API server.
2. The API server starts processing the request, after the time frame that is specified in the `x-waitTime`, and the request is processed asynchronously. The API server returns response with status code 206.
3. While request R1 is not processed completely, the client sends a status request for R1 at time ($t_1 + m$) will get back response with status code 206.

4. When the request R1 is completed, the response for request will be cached and same will be returned when client sends a status request for R1.
5. The client sends status request for R1 at time $(t1 + m + n)$, where n is less than retention time frame then the response is returned from the cache. After returning the response to the client, response is deleted from the cache.
6. Similarly, the client sends request R2 at $t2$ and the request is completed at time $(t2 + x)$, then response data is added in the cache.
7. At time $(t2 + x + y)$, the cached response is deleted from the cache, as it exceeds the retention policy time frame ($y \geq \text{Retention time frame}$).
8. The client sends the `getStatus` request for request R2 at time $(t2 + x + y + z)$ and the API server returns the error 400, because the response for request R2 is deleted from cache.

Response format of the API

A response for a request contains a status line, response header, and response body.

Status descriptions

A status line contains the elements listed in the following table.

Item	Description
Protocol	Displays the communication protocol used by the API.
Protocol version	Displays the version of the communication protocol used by the API.
Status code	Display the result of the request in a status code.
Message	Displays the content of the status code.

See the status codes of each API. For details about the status codes from errors prior to running the API, see [HTTP status codes descriptions \(on page 23\)](#).

Response headers descriptions

The following table lists the applicable response headers.

Response header	Description
Content-type	Content-type for every response: <code>application/json; charset=utf-8</code>
x-requestId	Request ID of the submitted request

HTTP status codes descriptions

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
409	CONFLICT	Resource already exists.
413	REQUEST ENTITY TOO LARGE	Server cannot process the request because the request entity is too large for processing.
500	SERVER ERROR	API operation request failed. Check the response body for details.
501	NOT IMPLEMENTED	API server does not support the specified action.

Status code	Message	Description
503	SERVICE UNAVAILABLE	API server cannot process the request, possibly because the API server instance is not running.

For a list of common error codes, see [API server response error codes and messages](#).

Timestamps and time zones

The Analyzer detail view REST API uses the format `yyyyMMdd_HH:mm:ss` for timestamp inputs and outputs.

For example, 23rd March 2015 10:00:00 is represented in the above format as `20150323_100000`. For more information on format, refer to `java SimpleDateFormat`.

The valid range for timestamp input is from `19020101_000000` to `20371231_235959`. Any timestamp input outside of this range is treated as an invalid value.

The daylight saving time settings are handled implicitly by `java.util.TimeZone`, which is a core Java class. Internally, new time zone settings are applied when JDK is updated. Java gets its time zone information from the IANA time zone database.

If you are using Oracle's JDK, the Oracle Timezone Updater tool is used to update only the time zone.



Note: In Alert APIs, all input and output timestamps are considered to be in the UTC time zone. Optionally, you can explicitly specify the time zone for API input timestamps in UTC + offset format; for example, UTC+01:00, UTC-08:00.

Authorization for APIs

APIs can be authorized using the Analyzer detail view local and the Active Directory users or by Ops Center Portal users. Refer to the following sections for more information:

- [Authorization for APIs using the Analyzer detail view local and Active Directory users \(on page 24\)](#)
- [Authorization for APIs using the Ops Center Portal users \(on page 25\)](#)

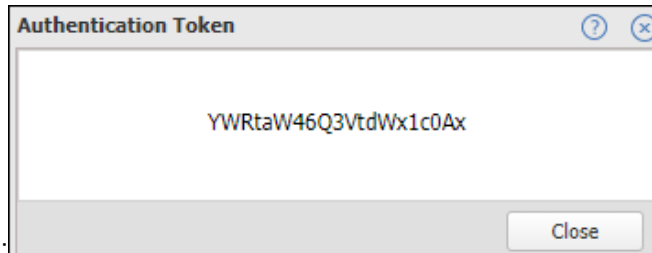
Authorization for APIs using the Analyzer detail view local and Active Directory users

The API request uses HTTP basic authorization. To authorize the request, set the Authorization header in the request:

```
Authorization: Basic<space><Authentication Token>
```

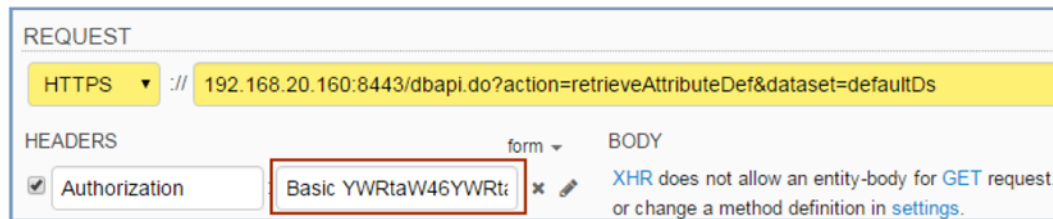

You can generate the authentication token using the local or Active Directory users in the Analyzer detail view UI. To generate the authentication token, click Manage > Administration > Authentication Token link.

The Authentication Token window shows the token as follows:



Using the authentication token

After you generate the authentication token, it should be used for each API request, as shown in the following example.



Changing the authentication token

When you change the user password of the Analyzer detail view, the authentication token is also changed.

Authorization for APIs using the Ops Center Portal users

The API request uses bearer authorization for Ops Center Portal users. To authorize the request, set the Authorization header in the request as follows:

```
Authorization: Bearer<space><Authorization Token>
```

The expression must be: Content-type: application/json

Use the Common Services APIs to generate the authorization token.



Note:

- The authorization token is refreshed after every 5 minutes. You must generate the new authorization token after 5 minutes.
- The User Management APIs of Analyzer detail view are not supported for Ops Center Portal users.

API parameters

Some of the API parameters are optional. If the optional parameter value is specified as JSON type `null`, then the API simply ignores the optional parameter.

Chapter 2: Data subset API resources

A data subset indicates the logical grouping of the resource data. Apart from maintaining the logical separation of data, it helps in updating the resource lifecycle when a user sends snapshot data.

Creating a data subset

You can create a new data subset. Each data subset must have a unique ID within the same dataset. If the data subset ID already exists in the dataset, then an error message is received in a response message.

Request line

```
Post baseUrl/dbapi.do?action=createDataSubset&dataset=<Datasetname>
```

Request body

```
{
  "dataSubsetId": "",
  "dataSubsetType": ""
}
```

Request example

```
POST /dbapi.do?action=createDataSubset&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "dataSubsetId": "US-NorthAmerica",
  "dataSubsetType": "VMware"
}
```

Request parameters

Parameter	Type	Description
action	String	{createDataSet} Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
dataSubsetId	String	The data subset is created with this ID. The ID must be unique within the dataset. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
dataSubsetType	String	Data subset type is used to group the related data subsets. Multiple data subsets can have the same data subset type. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Response body

```
{
  "dataSubset":""
}
{
  "code": "",
  "error": ""
}
```

Response example

```
{
  "dataSubset": "US-NorthAmerica"
}
```

Response parameters

Parameter	Type	Description
dataSubset	String	Created data subset ID.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
409	CONFLICT	Resource already exists.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
210000E4	Mandatory parameter {dataset} is not specified.	Parameter <code>dataset</code> is not specified.
21010002	Invalid JSON Request.	Specified JSON input is invalid.
21010008	Mandatory parameter {dataSubsetId} is not specified.	Parameter <code>dataSubsetId</code> is not specified.
2101000F	Invalid value specified for parameter {dataSubsetId}. It must be of type string.	Parameter <code>dataSubsetId</code> does not belong to the type string.
2102000F	Invalid value specified for parameter {dataSubsetId}. The {dataSubsetId} length must be between 2 to 32 characters.	Parameter <code>dataSubsetId</code> is less than 2 character or more than 32 characters.
2102000F	Invalid value specified for parameter {dataSubsetId}. The {dataSubsetId} contains an invalid characters. The valid characters are [alphanumeric, - and _].	Parameter <code>dataSubsetId</code> contains an invalid character.
21010008	Mandatory parameter {dataSubsetType} is not specified.	Parameter <code>dataSubsetId</code> is not specified.
2101000F	Invalid value specified for parameter {dataSubsetType}. It must be of type string.	Parameter <code>dataSubsetId</code> does not belong to the type string.
2102000F	Invalid value specified for parameter {dataSubsetType}. The {dataSubsetType} length must be between 2 to 32 characters.	Parameter <code>dataSubsetId</code> is less than 2 character or more than 32 characters.

Response code	Message	Condition
2102000F	Invalid value specified for parameter {dataSubsetType}. The {dataSubsetType} contains an invalid characters. The valid characters are [alphanumeric, -, and _].	Parameter dataSubsetType contains an invalid characters.
2101000C	Parameter {dataset} is not supported.	Parameter dataset is not supported.
210000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
210000E9	Authorization needed.	Request is sent with an invalid authentication token.
210000E4	Dataset with {name} [ds1] does not exist.	Request sent for a dataset does not exist.
210000E0	HTTP method [GET] is not supported for action [createDataSubset].	Request is sent with an HTTP method other than POST.
21050004	DataSubset with {dataSubsetId} [US-NorthAmerica] already exists.	Data subset already exists with the same subset ID.
210100E2	Application encountered an internal error.	API server encountered an unknown error.

Getting information about the data subset

You can obtain data subset information such as data subset ID, data subset type, and data availability.

Request line

```
GET baseUrl/dbapi.do?
action=retrieveDataSubset&dataset=<Datasetname>&dataSubsetId=<DataSubsetId>
```

Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	{retrieveDataSubset} Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
dataSubsetId	String	(Optional) Data subset ID. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed. If not specified, all data subset IDs are returned.
utcOffset	String	(Optional) Offset from the coordinated universal time. The UTC offset is applicable for API request and response. UTC+/-HH:MM Example: UTC+09:00 Default: Coordinated Universal Time (UTC).

Request example

```
GET /dbapi.do?action=retrieveDataSubset&dataset=defaultDs&dataSubsetId=US-NorthAmerica HTTP /1.1 Authorization:BasicZnJlZDpmcmVkkx
```

Response body

```
{
  "dataSubsetId": "",
  "dataSubsetType": "",
  "dataAvailability": {
    "startTime": "",
    "endTime": ""
  }
}
{
  "code": "",
```



```
"error":""
}
```

Response example

```
{
  "dataSubsetId": "US-NorthAmerica",
  "dataSubsetType": "VMware",
  "dataAvailability": {
    "startTime": "20141211_120000",
    "endTime": "20141213_120000"
  }
}
```

Response parameters



Note: If the `dataSubsetId` is not specified in the request, then all `dataSubsets` are returned as an array of objects defined in the following table.

Parameter	Type	Description
<code>dataSubsetId</code>	String	Data subset ID.
<code>dataSubsetType</code>	String	Data subset type.
<code>dataAvailability</code>	Object	Data availability for the given data subset.
<code>dataAvailability.startTime</code>	String	Start time of the data is in the following format: yyyyMMdd_HH:mm:ss Time zone: UTC
<code>dataAvailability.endTime</code>	String	End time of the data is in the following format: yyyyMMdd_HH:mm:ss Time zone: UTC
<code>code</code>	String	Eight-digit hexadecimal error code.
<code>error</code>	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
220000E4	Mandatory parameter {dataset} is not specified.	Parameter <code>dataset</code> is not specified.
22040008	Mandatory parameter {dataSubsetId} is not specified	Parameter <code>dataSubsetId</code> is not specified.
2202000F	Invalid value is specified for parameter {dataSubsetId}. The {dataSubsetId} length must be between 2 to 32 characters.	Parameter <code>dataSubsetId</code> is less than 2 characters or more than 32 characters.
2202000F	Invalid value is specified for parameter {dataSubsetId}. The {dataSubsetId} contains an invalid character. The valid characters are [alphanumeric, -, and _].	Parameter <code>dataSubsetId</code> contains an invalid character.
220000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
220000E9	Authorization needed.	Request is sent with an invalid authentication token.
220000E4	Dataset with {name};[ds1] does not exist.	Request is sent for a dataset that does not exist.
22050005	DataSubset with {dataSubsetId} [US-NorthAmerica] does not exist.	Request is sent for a data subset 'US-NorthAmerica' that does not exist.
220000E0	HTTP method [POST] is not supported for action [retrieveDataSubset].	Request is sent with an HTTP method other than GET.
220400E2	Application encountered an internal error.	API server encountered an unknown error.

Chapter 3: Attribute definitions

Analyzer detail view server includes default attribute definition for the data that is collected using the Analyzer probe server. You can also create custom attribute definitions. This section describes how to create, get, update, and delete the custom attribute definition.

Creating a custom attribute definition

You can create a new attribute definition. The attribute definition must have a unique ID. If the attribute definition ID already exists, then an error message is received. You can create multiple attribute definitions in a single request.

Request line

```
Post baseUrl/dbapi.do?action=createAttributeDef&dataset=<Datasetname>
```

Request body

```
[
  {
    "id": "",
    "name": "",
    "type": ""
  },
  {
    "id": "",
    "name": "",
    "type": ""
  },
  {
    "dynamic": ,
    "id": "",
    "name": "",
    "type": "",
    "unit": "",
    "bandFactor":
  }
]
```

Request example

```
POST /dbapi.do?action=createAttributeDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
[
  {
    "id":"numCores",
    "name":"Number Of Cpu cores",
    "type":"scalar"
  },
  {
    "id":"cpuUsage",
    "name":"cpuUsage",
    "type":"timeseries"
  },
  {
    "dynamic":true,
    "id":"^byteFlow_(\\d{1,5})_(\\d{1,5})_([A-Za-z]*[0-9]*)$",
    "name":" Network Traffic",
    "type":"timeseries",
    "unit":"Bytes",
    "bandFactor":1
  }
]
```

Request parameters

Parameter	Type	Description
action	String	createAttributeDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
id	Integer	Unique attribute definition ID. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Parameter	Type	Description
name	String	(Optional) Attribute definition name. From 2 to 128 alphanumeric characters are allowed. Default: Attribute ID
unit	String	(Optional) Unit of measure. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.. Default: Blank
dynamic	Boolean	(Optional) Specify if this attribute is dynamic or not. true or false Default: false
bandFactor	Float	(Optional) Decimal precision for attribute value. A positive decimal number. Default: 1

A resource can have multiple combinations of the same attribute. For example, in the network flow analysis between a pair of IP addresses, a usage metric can be collected for all combinations of source port, destination port, and protocol (UDP or TCP). Although multiple combinations are possible, only a single definition is maintained for an attribute. These types of attributes are dynamic attributes.

Parameter	Description
dynamic	Indicates that this definition is for a dynamic attribute.
id	Dynamic attribute definition ID consists of two parts: static and dynamic, separated by an underscore (_). The dynamic part is optional. Dynamic attribute definition ID must begin with the caret (^) character, followed by the static part. The static part must not have an underscore (_), but can have from 2 to 32 alphanumeric and hyphen (-) characters.

Parameter	Description
	<p>The dynamic part must be a valid regular expression following the Java pattern class.</p> <p>For example, <code>^byteFlow_(\\d{1,5})_(\\d{1,5})_([A-Za-z]*[0-9]*)\$</code></p> <p>Where, <code>byteFlow</code> is the static part, and <code>(\\d{1,5})_(\\d{1,5})_([A-Za-z]*[0-9]*)\$</code> is the dynamic part.</p> <p>The following instances match the above attribute ID pattern:</p> <ul style="list-style-type: none"> byteFlow_2213_3454_udp byteFlow_8843_6753_tcp

Response body

```
{
  "created": [
    ""
  ],
  "failed": [
    {
      "code": "",
      "id": "",
      "error": ""
    }
  ]
}
```

Response example

```
{
  "created": [
    "^byteFlow_(\\d{1,5})_(\\d{1,5})_([A-Za-z]*[0-9]*)$"
  ],
  "failed": [
    {
      "code": "41050004",
      "id": "numCores",
      "error": "Attribute definition with {id}: [numCores] already exists"
    },
    {
      "code": "4102000F",
      "id": "cpuUsage%",
      "error": "Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are [alphanumeric, - and _]."

```

```

    }
  ]
}

```

Response parameters

Parameter	Type	Description
created	String[]	List of created attribute IDs.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:id	String	Failed attribute ID.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.

Status code	Message	Description
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all the specified attribute definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
41050004	Attribute definition with {id}: [cpuUsage] already exists.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The attribute definition ID already exists in the database.
4102000F	Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The specified attribute definition ID contains invalid characters.
4102000F	Invalid value specified for parameter {id}. Dynamic attributeld must start with [^].	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The specified dynamic attribute definition ID does not start with ^ .

Response code	Message	Condition
4102000F	Invalid value specified for parameter {id}. Specified regular expression is invalid.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The specified dynamic attribute definition ID contains invalid regex.
41050004	Attribute definition with {id}: [cpuUsage] already exists.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The specified attribute definition ID is already present with the same ID as the static part of the dynamic attribute definition ID.
41050004	Attribute definition with {id}: [^byteFlow_(\\d{1,5})_(\\d{1,5})_([A-Za-z]+[0-9]+)\$] already exists.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The dynamic attribute definition is already present with a static part of the dynamic attribute ID, but with different regex.
4102000F	Invalid value specified for parameter {id}. {id} length must be between 2-32 characters.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The attribute definition ID length is less than 2 or more than 32.
41020008	Mandatory parameter {type} not specified.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The parameter <i>type</i> is not specified.

Response code	Message	Condition
4102000F	Invalid value specified for parameter {type}. Valid values: scalar and timeseries.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The attribute definition contains an invalid value for the parameter <i>type</i> .
41020007	No regular expression is specified.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The dynamic attribute definition contains no regex.
4102000F	Invalid value specified for parameter {bandFactor}. Band factor must be one of [1.0, 0.1, 0.01, 0.001, 0.0001].	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The attribute definition contains an invalid value for the parameter <i>bandFactor</i> .
4102000F	Invalid value specified for parameter {unit}. {unit} length must be between 2-32 characters.	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The attribute unit length is less than 2 or more than 32.
4102000F	Invalid value specified for parameter {unit}. {unit} contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one attribute definition is created and the creation of one or more attribute definitions failed. The specified <i>unit</i> parameter contains invalid characters.
4102000F	Invalid value specified for parameter {name}. {name} length must be between 2-128 characters.	Minimum of one attribute definition is created and one or more attribute definitions failed as the specified attribute name is not valid in terms of allowed length.
4102000C	Parameter {bandFactor} not supported. Parameter {bandFactor} is only supported for timeseries attributes.	The parameter <i>bandFactor</i> is specified for the scalar attribute.

Response code	Message	Condition
4102000F	Invalid value specified for parameter {id}. Dynamic attributeld must start with [^].	Invalid value is specified for the dynamic attribute in the <i>id</i> parameter.
4102000C	Parameter {dynamic} not supported. Parameter {dynamic} is only supported for timeseries attributes.	The <i>dynamic</i> parameter is specified for the scalar attribute.
41040007	No Attribute definition is specified.	The attribute definition is not specified for insertion.
41010002	Invalid JSON Request.	Specified JSON input is invalid.
41050008	Mandatory parameter {id} not specified.	Attribute definition ID is not specified in the <i>id</i> parameter.
410000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
410000E9	Authorization needed.	Request is sent with an invalid authentication token.
410000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
410000E0	HTTP method [PUT] is not supported for action [createAttributeDef].	Request is sent with an HTTP method other than POST.
410200E2	Application encountered an internal error.	API server encountered an unknown error.

Getting information about an attribute definition

You can obtain the following information about an attribute definition: attribute ID, attribute type, attribute name, and attribute unit of measure. If the requested attribute does not exist, then an error message is received in the response message.

Request line

```
Post baseUrl/dbapi.do?action=retrieveAttributeDef&dataset=<Datasetname>
```



Note: You can send the HTTP-POST or GET request. HTTP-POST is useful when the request length exceeds 1024 characters, the maximum allowed request size for the HTTP-GET request.

The request body for HTTP-POST must be in JSON format.

To get the dynamic attribute definitions, use HTTP-POST and specify the attribute definition ID in the request body.

Request body

```
{
  "attrIds": [
    "",
    "",
    ""
  ]
}
```

Request example using GET

```
GET /dbapi.do?action=retrieveAttributeDef&dataset=defaultDs&attrIds=numCPU,cpuUsage,
memory HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

Request example using POST

```
POST /dbapi.do?action=retrieveAttributeDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "attrIds": [
    "numCPU",
    "cpuUsage",
    "memory"
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	retrieveAttributeDef Specify the API function to be invoked.

Parameter	Type	Description
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrlds	String[]	<p>(Optional) Attribute IDs are involved in request URL for GET method and in request body for POST method.</p> <p>From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.</p> <p>GET: One or more comma-separated attribute IDs.</p> <p>Example: A,B,C</p> <p>POST: JSON format</p> <p>Example:</p> <pre>{ "attrlds":["A","B","C"] }</pre>

Response body

```
{
  "retrieved": [
    {
      "id": "",
      "name": "",
      "unit": "",
      "type": "",
      "bandFactor":
      "dynamic":
    }
  ],
  "failed": [
    {
      "code": "",
      "id": "",
      "error": ""
    }
  ]
}
```

Response example

```
{
  "retrieved": [
    {
      "id": "cpuUsage",
      "name": "cpuUsage",
      "unit": "null",
      "type": "timeseries",
      "bandFactor": 1.000000
      "dynamic": false
    }
  ],
  "failed": [
    {
      "code": "42050005",
      "id": "numCPU",
      "error": "Attribute definition with {id}: [numCPU] does not exist"
    },
    {
      "code": "42050005",
      "id": "memory",
      "error": "Attribute definition with {id}: [memory] does not exist"
    }
  ]
}
```

Response parameters

Parameter	Type	Description
retrieved	Object[]	Retrieved attribute definition.
retrieved:id	String	Attribute definition ID.
retrieved:name	String	Attribute definition name
retrieved:unit	String	Attribute definition unit. It is blank if unit is not specified at the time of creation.
retrieved:type	String	Attribute definition type. It is either scalar or time series.

Parameter	Type	Description
retrieved:bandFactor	Float	Attribute definition band factor. Precision of time series data point is controlled by band factor value.
retrieved:dynamic	Boolean	Specifies weather attribute definition is dynamic or not. If attribute is not dynamic, this information is not displayed.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:id	String	Failed attribute ID.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.

Status code	Message	Description
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified attribute definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
42050005	Attribute definition with {id}: [cpuUsage] does not exist.	Minimum of one attribute definition is obtained and getting one or more attribute definitions failed. Specified attribute definition does not exist.
4202000F	Invalid value specified for parameter {id}. {id} contains invalid characters. The valid characters are alphanumeric, -, , and _.	Minimum of one attribute definition is obtained and getting one or more attribute definitions failed. The static part of the attribute definition ID contains invalid characters.

Response code	Message	Condition
4 202000F	Invalid value specified for parameter {id}. {id} length must be between 2 to 32 characters.	Minimum of one attribute definition is obtained and getting one or more attribute definitions failed. The length of the static part of the attribute ID exceeds the allowed length.
42020007	No regular expression is specified.	Minimum of one attribute definition is obtained and getting one or more attribute definitions failed. The regular expression is not specified in the dynamic attribute ID.
42010002	Invalid JSON Request.	Specified JSON input is invalid.
420000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
420000E9	Authorization needed.	Request is sent with an invalid authentication token.
420000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
420000E0	HTTP method [PUT] is not supported for action [retrieveAttributeDef].	Request is sent with any HTTP method other than GET or POST.
420100E2	Application encountered an internal error.	API server encountered an unknown error.

Updating attribute definition information

You can update the display name and unit of an attribute definition. If the requested attribute does not exist, then an error message is received in the response message. You can update multiple attribute definitions in a single request. You cannot update the predefined attribute definitions.

Request line

```
Post baseUrl/dbapi.do?action=updateAttributeDef&dataset=<Datasetname>
```

Request body

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

Request example

```
POST /dbapi.do?action=updateAttributeDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
[
  {
    "id": "numCores",
    "name": "Number Of Cpu"
  },
  {
    "id": "cpuUsage",
    "name": "Cpu Usage",
    "unit": "Percent"
  }
]
```

Request parameters

Parameter	Type	Description
action	String	updateAttributeDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Parameter	Type	Description
id	String	Attribute definition ID. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
name	String	(Optional) Attribute definition name. From 2 to 128 alphanumeric characters are allowed. Default: id
unit	String	(Optional) Unit of measure. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed. Default: Blank

Response body

```
{
  "updated": [
    ""
  ],
  "failed": [
    {
      "code": "",
      "id": "",
      "error": ""
    }
  ]
}
```

Response example

```
{
  "updated": [
    "numCores"
  ],
  "failed": [
    {
      "code": "43050005",
      "id": "memUsage",
      "error": "Attribute definition with {id}: [memUsage] does not exist"
    }
  ]
}
```

```

    ]
  }

```

Response parameters

Parameter	Type	Description
updated	String[]	List of updated attribute IDs.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:id	String	Failed attribute ID.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.

Status code	Message	Description
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified attribute definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
43050005	Attribute definition with {id}: [cpuUsage] does not exist.	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The specified attribute definitions do not exist.
4302000F	Invalid value specified for parameter {unit}. {unit} contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The attribute unit contains invalid characters.
4302000F	Invalid value specified for parameter {unit}. {unit} length must be between 2 to 32 characters.	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The attribute unit length is less than 2 or more than 32.

Response code	Message	Condition
4302000F	Invalid value specified for parameter {name}. {name} length must be between 2 to 128 characters.	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The specified attribute name exceeds the allowed length.
4302000F	Invalid value specified for parameter {id}. {id} length must be between 2 to 32 characters.	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The length of a static part of the attribute definition ID exceeds the allowed length.
43050014	Cannot update the read-only 'Attribute definition'.	Minimum of one attribute definition is updated and one or more attribute definitions failed to update. The attribute definition corresponding to the specified attribute definition ID is predefined.
43040007	No Attribute definition is specified.	The attribute definition is not specified for update.
43010002	Invalid JSON Request.	The specified JSON input is invalid.
43050008	Mandatory parameter {id} is not specified.	The parameter <code>id</code> is not specified.
420000E4	Mandatory parameter {dataset} is not specified.	The parameter <code>dataset</code> is not specified.
430000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
430000E9	Authorization needed.	Request is sent with an invalid authentication token.
430000E4	Dataset with {name};[ds1] does not exist.	Request is sent for a <code>dataset</code> that does not exist.
430000E0	HTTP method [GET] is not supported for action [updateAttributeDef].	Request is sent with any HTTP method other than POST.

Response code	Message	Condition
430500E2	Application encountered an internal error.	API server encountered an unknown error.

Deleting the custom attribute definition

You can delete a specific attribute definition or multiple attribute definitions. If the requested attribute is being used by any other resource definition or if it does not exist, then an error message is received in a response message. You can delete multiple attribute definitions in a single request.

The data of the deleted attribute definition is not deleted from the database. If you need to create the same attribute definition, then you can retrieve the old data. You cannot delete predefined attribute definitions.

Request line

Post `baseUrl/dbapi.do?action=deleteAttributeDef&dataset=<Datasetname>`



Note: To delete the dynamic attribute definitions, use HTTP-POST and specify the attribute definition IDs in the request body.

Request body

```
{
  "attrIds": [
    "",
    "",
    ""
  ]
}
```

Request example using GET

```
GET /dbapi.do?action=deleteAttributeDef&dataset=defaultDs&attrIds=numCPU,cpuUsage,
memory HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

Request example using POST

```
Post /dbapi.do?action=deleteAttributeDef&dataset=defaultDs memory HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "attrIds": [
```



```

    "numCPU",
    "cpuUsage",
    "memory"
  ]
}

```

Request parameters

The following table describes the values for the supported parameters of a query string:

Parameter	Type	Description
action	String	deleteAttributeDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrIds	String[]	Attribute IDs are involved in request URL for GET method and in request body for POST method. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed. GET: One or more comma-separated attribute IDs. Example: A,B,C POST: JSON format Example: <pre> { "attrIds":["A","B","C"] } </pre>

To delete the dynamic attribute definition, you must provide the attribute definition ID that you specified while creating it. The following example deletes the dynamic attribute definition for the specified ID. The request to delete the dynamic attributes that contain the comma-separated attribute definitions IDs must be called with the POST method.

Response body

```

{
  "deleted": [

```

```

    ""
  ],
  "failed": [
    {
      "code": "",
      "id": "",
      "error": ""
    },
    {
      "code": "",
      "id": "",
      "error": ""
    }
  ]
}

```

Response example

```

{
  "deleted": [
    "cpuUsage"
  ],
  "failed": [
    {
      "code": "44050005",
      "id": "memory",
      "error": "Attribute definition with {id}: [memory] does not exist"
    },
    {
      "code": "44050005",
      "id": "numCPU",
      "error": "Attribute definition with {id}: [numCPU] does not exist"
    }
  ]
}

```

Response parameters

Parameter	Type	Description
deleted	String[]	List of deleted attribute IDs.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:id	String	Failed attribute ID.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified attribute definitions fail, then the `500 SERVER ERROR` message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
44050005	Attribute definition with {id}: [cpuUsage] does not exist.	Minimum of one attribute definition is deleted and the deletion of one or more attribute definitions failed. The specified attribute definitions do not exist.
4402000F	Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are alphanumeric and -.	Minimum of one attribute definition is obtained and getting one or more attribute IDs failed. The static part of the attribute ID contains the invalid characters.
4402000F	Invalid value specified for parameter {id}. {id} length must be between 2 to 32 characters.	Minimum of one attribute definition is obtained and getting one or more attribute IDs failed. The length of the static part of the attribute ID exceeds the allowed length.
44050011	Attribute definition deletion failed due to dependency on one or more Resource definition.	Minimum of one attribute definition is deleted and the deletion of one or more attribute definitions failed. There is a dependency on one or more resource definitions.
44050014	Cannot delete read-only 'Attribute definition'.	Minimum of one attribute definition is deleted and the deletion of one or more attribute definitions failed. The attribute definition corresponding to the specified attribute ID is predefined.
44010002	Invalid JSON Request.	Specified JSON input is invalid.
44040007	No Attribute definition is specified.	No attribute definition is specified for the deletion.
440000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.

Response code	Message	Condition
440000E9	Authorization needed.	Request is sent with an invalid authentication token.
440000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a <code>dataset</code> that does not exist.
440000E0	HTTP method [PUT] is not supported for action [deleteAttributeDef].	Request is sent with any HTTP method other than GET or POST.
440400E2	Application encountered an internal error.	API server encountered an unknown error.

Chapter 4: Resource definitions

Analyzer detail view server includes default resource definitions for the resources that are collected using the Analyzer probe server. You can also create custom resource definitions. This section describes how to create, get, update, and delete a resource definition.

Creating a custom resource definition

You can create a new resource definition. If the resource definition already exists, then an error message is received in the response message. You can create multiple resource definitions in a single request.

Request line

```
POST baseUrl/dbapi.do?action=createResourceDef&dataset=<Datasetname>
```

Request body

```
[
  {
    "type": "",
    "name": "",
    "attributes": [
      ""
    ],
    "relations": [
      "",
      ""
    ]
  },
  {
    "type": "",
    "name": "",
    "attributes": [
      "",
      "",
      ""
    ],
    "relations": [
      "",
      ""
    ]
  }
]
```

```
}
]
```

Request example

```
POST /dbapi.do?action=createResourceDef&dataset=defaultDs HTTP /1.1 Authorization:
Basic ZnJlZDpmcmVh
```

```
[
  {
    "type": "vm",
    "name": "VM",
    "attributes": [
      "name"
    ],
    "relations": [
      "host",
      "vDisk"
    ]
  },
  {
    "type": "host",
    "name": "VM",
    "attributes": [
      "name",
      "cpuUsage",
      "memUsage"
    ],
    "relations": [
      "vCpu",
      "vDisk"
    ]
  }
]
```

Request parameters

Parameter	Type	Description
action	String	createResourceDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Parameter	Type	Description
type	String	A unique resource type. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
name	String	(Optional) Resource definition name. From 1 to 32 alphanumeric characters are allowed. Default: type
attributes	String[]	(Optional) List of scalar and timeseries attribute IDs. Default: Blank
relations	String[]	(Optional) List of valid related resource types . Default: Blank
additionalProperties	Object	(Optional) Additional key value pairs to be stored with the resource definition. Note: Properties type, name, relations, and attributes cannot be specified in <code>additionalProperties</code> as they constitute a resource definition. JSON object having valid key value pairs. Maximum of 128 key value pairs can be specified. Key: From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed. Value: \, ", & , > , < , and ' are not allowed. From 1 to 1024 alphanumeric characters are allowed.

Response body

```
{
  "created": [
  ],
  "failed": [
    {
      "code": "",
      "type": "",
      "error": ""
    }
  ]
}
```


Response example

```
{
  "created": [
  ],
  "failed": [
    {
      "code": "31020012",
      "type": "vm",
      "error": "createResourceDef failed for {type}: [vm]. Create attribute definition for [name]."
    },
    {
      "code": "31020012",
      "type": "host",
      "error": "createResourceDef failed for {type}: [host]. Create attribute definition for [name, cpuUsage, memUsage]."
    }
  ]
}
```

Response parameters

Parameter	Type	Description
created	String[]	List of created resource types.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:type	String	Failed resource type.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.

Status code	Message	Description
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified resource definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
31050004	Resource definition with {type}:[vm] already exists.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The requested resource definition already exists.
31010008	Mandatory parameter {type} is not specified.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The resource type is not specified.
3101000F	Invalid value is specified for parameter {type}. It must be of type string.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified resource type is not of type string.
3101000F	Invalid value is specified for parameter {type}. Resource {type}:[] length must be between 1 to 32 characters.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified resource type is less than 1 character or more than 32 character.
3101000F	Invalid value is specified for parameter {type}. The {<?<} contains an invalid characters. The valid characters are alphanumeric, -, and _ .	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified resource type contains an invalid characters.
3101000F	Invalid value is specified for parameter {name}. It must be of type string.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified resource name is not of the type string.

Response code	Message	Condition
3101000F	Invalid value is specified for parameter {name}. The {name} length must be between 1 to 32 characters.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The resource name is not the allowed length.
3101000F	Invalid value is specified for parameter {attributes}. It must be of type JSON list of string.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified attribute is not of type JSON.
3101000F	createResourceDef failed for {type}:[{host}]. Create attribute definition for [cpuUsage].	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified attribute IDs are not defined.
3101000F	Invalid value specified for parameter {relations}. It must be of type JSON list of string.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified relation is not of type JSON.
3101000F	Invalid value is specified for parameter {relations}. The {<?>} contains an invalid characters. The valid characters are alphanumeric, -, and _.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The relation contains an invalid character.
3101000F	Invalid value is specified for parameter {relations}. Resource {type}:[] length must be between 1 to 32 characters.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified relations are not in the allowed length
3101000F	Invalid value is specified for parameter {additionalProperties}. It must be of type JSON object.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified additionalProperties are not of type JSON object.

Response code	Message	Condition
3101000F	Invalid value is specified for parameter {additionalProperties}. The maximum 128 additional properties can be provided in the resource definition.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The properties specified in the <code>additionalProperties</code> are more than 128.
3101000F	Invalid value is specified for parameter {additionalProperties}. Additional properties key cannot be one of the following: type, configAttributes, displayName, and directRelations.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. Invalid properties are specified in the <code>additionalProperties</code> .
3101000F	Invalid value is specified for parameter {additionalProperties}. The {@@} contains invalid characters. The valid characters are alphanumeric, -, and _.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The key specified in <code>additionalProperties</code> contains an invalid character.
3101000F	Invalid value is specified for parameter {additionalProperties}. The {} length must be between 1 to 32 characters.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The key specified in <code>additionalProperties</code> is not in allowed length.
3101000F	Invalid value is specified for parameter {additionalProperties values}. It must be of type string.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified value for a key in <code>additionalProperties</code> is not of type string.

Response code	Message	Condition
3101000F	Invalid value is specified for parameter {additionalProperties}. {key} length must be between 1 to 32 characters.	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The specified value for a key in <code>additionalProperties</code> is not the allowed length.
3101000F	Invalid value specified for parameter {additionalProperties}. The {>>} contains invalid characters. The valid characters are alphanumeric, -, and _.	Minimum of one resource definition is created and one or more resource definitions failed. The value specified for a key in <code>additionalProperties</code> contains invalid character.
31020012	<code>createResourceDef</code> failed for {type}:[host]. Remove self association from resource definition [host].	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The resource definition is associated to itself.
31020012	<code>createResourceDef</code> failed for {type}:[host]. Resource type [host] is already associated with [vm].	Minimum of one resource definition is created and the creation of one or more resource definitions failed. The resource definition contains <code>relations</code> that is already defined through related resource type.
31010002	Invalid JSON Request.	Specified input JSON is invalid.
310000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
310000E9	Authorization needed.	Request is sent with an invalid authentication token.
310000E4	Dataset with {name}:[ds1] does not exist.	Request sent for a dataset does not exist.
310000E0	HTTP method [PUT] is not supported for action [<code>createResourceDef</code>].	Request is sent with any HTTP method other than POST.

Response code	Message	Condition
310000E2	Application encountered an internal error.	API server encountered an unknown error.

Getting information about a resource definition

You can obtain information about a resource definition. If the requested resource definition does not exist, then an error message is received in a response message. You can request information for multiple resource definitions in a single request.

Request line

```
baseUrl/dbapi.do?action=retrieveResourceDef&dataset=<Datasetname>
```

Request body

```
{
  "types": [
    "",
    "",
    ""
  ]
}
```

Request example using GET

```
GET /dbapi.do?action=retrieveResourceDef&dataset=defaultDs&types=host,vm,disk
HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

Request example using POST

```
POST /dbapi.do?action=retrieveResourceDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "types": [
    "host",
    "vm",
    "disk"
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	retrieveResourceDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
types	String[]	(Optional) One or more comma-separated resource types. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed. Default: Retrieves all resource definitions.

Response body

```
{
  "retrieved": [
    {
      "type": "",
      "name": "",
      "attributes": [
        "",
        "",
        ""
      ],
      "relations": [
        ""
      ]
    }
  ],
  "failed": [
    {
      "code": "",
      "type": "",
      "error": ""
    }
  ]
}
```


Response example

```
{
  "retrieved": [
    {
      "type": "host",
      "name": "Host",
      "attributes": [
        "name",
        "cpuUsage",
        "memUsage"
      ],
      "relations": [
        "disk"
      ]
    }
  ],
  "failed": [
    {
      "code": "32040005",
      "type": "disk",
      "error": "Resource definition with {type}:[disk] does not exist"
    },
    {
      "code": "32040005",
      "type": "vm",
      "error": "Resource definition with {type}:[vm] does not exist"
    }
  ]
}
```

Response parameters

Parameter	Type	Description
retrieved	Object[]	List of resource definitions.
retrieved.type	String	Resource type.
retrieved.name	String	Resource definition name.
retrieved.attributes	String[]	List of scalar and timeseries attribute IDs.
retrieved.relations	String[]	List of related resource types.
retrieved.additionalProperties	Object	All defined additional properties.
failed	Object[]	List of failed entities.

Parameter	Type	Description
failed:code	String	Eight-digit hexadecimal error code.
failed:type	String	Failed resource type.
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified resource definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
32040005	Resource definition with {type}:[host] does not exist.	Minimum of one resource definition is obtained and getting one or more resource definition types failed. Specified resource definition type does not exist.
3202000F	Invalid value specified for parameter {type}. Resource {type}:[] contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one resource definition is obtained and getting one or more resource definition types failed. Specified resource type contains invalid characters.
32040008	Mandatory parameter {type} not specified.	Minimum of one resource definition is obtained and getting one or more resource definition types failed. Specified resource type is blank.
3202000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.	Minimum of one resource definition is obtained and getting one or more resource definition types failed. Specified resource type is not valid in terms of allowed length.
32010002	Invalid JSON Request.	Specified JSON input is invalid.
320000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
320000E9	Authorization needed.	Request is sent with an invalid authentication token.

Response code	Message	Condition
320000E4	Dataset with {name}:[ds1] does not exist .	Request is sent for a dataset that does not exist.
320000E0	HTTP method [PUT] is not supported for action [retrieveResourceDef.]	Request is sent with any HTTP method other than POST.
320000E2	Application encountered an internal error.	API server encountered an unknown error.

Updating resource definition information

You can update the resource definitions in a dataset. If the requested resource definition does not exist, then an error message is received in the response message. You can update multiple resource definitions in a single request.

You cannot update the predefined resource definitions.

Request line

```
POST baseUrl/dbapi.do?action=updateResourceDef&dataset=<Datasetname>
```

Request body

```
[
  {
    "type": "",
    "name": "",
    "attributes": {
      "add": [
        ""
      ],
      "remove": [
        ""
      ]
    }
  },
  {
    "type": "",
    "name": "",
    "attributes": {
      "completeList": [
        " ",
        ""
      ]
    }
  }
],
```

```

    "relations":{
      "add":[
        ""
      ]
    }
  }
}
]

```

Request example

```

POST /dbapi.do?action=updateResourceDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk

```

```

[
  {
    "type":"vm",
    "name":"VM Updated",
    "attributes":{
      "add":[
        "cpuUsage"
      ],
      "remove":[
        "memUsage"
      ]
    }
  },
  {
    "type":"host",
    "name":"Host Updated",
    "attributes":{
      "completeList":[
        "cpuCores ",
        "cpuUsage"
      ]
    },
    "relations":{
      "add":[
        "vm"
      ]
    }
  }
]

```

Request parameters

Parameter	Type	Description
action	String	updateResourceDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
type	String	A unique resource definition type. From 1 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
name	String	(Optional) Resource definition name. From 1 to 32 alphanumeric characters are allowed. Default: type
attributes	Object[]	(Optional) Scalar and timeseries attributes. An existing attribute ID. Default: Blank
relations	Object[]	(Optional) Related resource definition types. A valid resource definition type. Default: Blank
additionalProperties	String[]	(Optional) Additional key value pairs to be stored the with resource definition. This adds new properties in resource definition and updates already existing additional properties.

Parameter	Type	Description
		<p>Note: Properties type, name, relations and attributes cannot be specified in <code>additionalProperties</code> as they constitute a resource definition.</p> <p>JSON object having valid key value pairs. Maximum 128 key value pairs can be specified.</p> <p>Key: From 2 to 32 alphanumeric, underscore (<code>_</code>), and hyphen (<code>-</code>) characters are allowed.</p> <p>Value: From 1 to 1024 alphanumeric characters are allowed, <code>\</code>, <code>"</code>, <code>&</code>, <code><</code>, <code>></code>, and <code>'</code> are not allowed.</p>

Parameter attributes can have `add`, `remove`, and `completeList`:

- **add:** Contains a list of scalar and timeseries attributes to be added.
- **remove:** Contains a list of scalar and time series attributes to be removed.
- **completeList:** Contains a complete list of scalar and timeseries attributes. If this list is present, it will overwrite all existing attributes.



Note: If `completeList` is specified, then `add` and `remove` are ignored.

Similarly, parameter relations can also have `add`, `remove`, or `completeList`.

Response body

```
{
  "updated": [
    ""
  ],
  "failed": [
    {
      "code": "",
      "type": "",
      "error": ""
    }
  ]
}
```

```

]
}

```

Response example

```

{
  "updated": [
    "vm"
  ],
  "failed": [
    {
      "code": "33020010",
      "type": "host",
      "error": "Resource definition update failed due to already existing relation.
Resource type [ host ] already has relation with [ vm ]."
    }
  ]
}

```

Response parameters

Parameter	Type	Description
updated	String[]	List of updated resource definition types.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:type	String	Failed resource type.
failed:error	String	Error message.

Return codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified resource definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
33010008	Mandatory parameter {type} not specified.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The resource definition type is not specified or the specified resource type is not of parameter type string.

Response code	Message	Condition
3301000F	Invalid value specified for parameter {name}. It must be of type string.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The resource definition name is not of parameter type string.
3301000E	[add1] is not a valid key in parameter {{attributes}}.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>attributes</i> contains an invalid key.
3301000F	Invalid value specified for parameter {attributes}. Atleast one correct key must be specified in {attributes}.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>attributes</i> does not contain any key.
3301000F	Invalid value specified for parameter {attributes}. It must be of type JSON object.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>attributes</i> is not a JSON object.
3301000F	Invalid value specified for parameter {add}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>add</i> in <i>attributes</i> is not of type JSON array.
3301000F	Invalid value specified for parameter {completeList}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>completeList</i> in <i>attributes</i> is not of type JSON array.

Response code	Message	Condition
3301000F	Invalid value specified for parameter {remove}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>remove</i> in <i>attributes</i> is not of type JSON array.
3301000E	[add1] is not a valid key in parameter {{relations}}.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>relations</i> contains an invalid key.
3301000F	Invalid value specified for parameter {relations}. At least one correct key must be specified in relations.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>relations</i> does not contain the key.
3301000F	Invalid value specified for parameter {relations}. It must be of type JSON object.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>relations</i> is not a JSON object.
3301000F	Invalid value specified for parameter {add}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>add</i> in <i>relations</i> is not of type JSON array.
3301000F	Invalid value specified for parameter {completeList}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>completeList</i> in <i>relations</i> is not of type JSON array.

Response code	Message	Condition
3301000F	Invalid value specified for parameter {remove}. It must be of type JSON array.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>remove</i> in <i>relations</i> is not of type JSON array.
3301000F	Invalid value specified for parameter {additionalProperties}. It must be of type JSON object.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The parameter <i>additionalProperties</i> is not of type JSON object.
3301000A	Nothing specified for update.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The request does not contain any parameter.
3302000F	Invalid value specified for parameter {type}. Resource {type}:[!] contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified resource type contains invalid characters.
3302000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified resource type is not valid in terms of allowed length.
3302000F	Invalid value specified for parameter {name}. {name} length must be between 2-32 characters.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified resource name is not valid in terms of allowed length.

Response code	Message	Condition
3302000F	Invalid value specified for parameter {type}. Resource {type}:[] length must be between 1-32 characters.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified relation is not valid in terms of allowed length.
3302000F	Invalid value specified for parameter {type}. Resource {type}:[!@] contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified relations contain invalid characters.
33020010	Resource definition update failed due to self relation. Remove self relation from resource definition [host].	Resource definition contains relation to itself.
33020010	Resource definition update failed due to already existing relation. Resource type [host] already has relation with [vm].	Minimum of one resource definition is updated and one or more resource definitions failed to update. The specified relation is already defined through related resource type.
33020012	updateResourceDef failed for {type}:[h]. Number of additional properties can not be more than 128 in a resource definition.	Minimum of one resource definition is updated and one or more resource definitions failed to update. More than 128 <i>additionalProperties</i> are specified.
33020012	updateResourceDef failed for {type}:[h]. Additional properties key cannot be one of type, configAttributes, displayName or directRelations.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The not allowed property is specified in the <i>additionalProperties</i> .
33020012	updateResourceDef failed for {type}:[h]. Invalid keys for additional properties. Keys : [!@]	Minimum of one resource definition is updated and one or more resource definitions failed to update. The key specified in the <i>additionalProperties</i> contains invalid characters.

Response code	Message	Condition
33020012	updateResourceDef failed for {type}:[h]. Invalid keys for additional properties. Keys : [{}]	Minimum of one resource definition is updated and one or more resource definitions failed to update. The key specified in <i>additionalProperties</i> is not valid in terms of allowed length.
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]	Minimum of one resource definition is updated and one or more resource definitions failed to update. The value specified for a key in <i>additionalProperties</i> is not of parameter type string.
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]	Minimum of one resource definition is updated and one or more resource definitions failed to update. The value specified for a key in the <i>additionalProperties</i> is not valid in terms of allowed length.
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]	Minimum of one resource definition is updated and one or more resource definitions failed to update. The value specified for a key in <i>additionalProperties</i> contains an invalid character.
33050014	Cannot update read-only 'Resource definition'.	Minimum of one resource definition is updated and one or more resource definitions failed to update. The resource definition corresponding to the specified resource definition type is pre-defined.

Response code	Message	Condition
33050005	Resource definition with {type}:[host] does not exist.	Minimum of one resource definition is updated and one or more resource definitions failed to update. Specified resource definitions do not exist.
33010002	Invalid JSON Request.	Specified JSON input is invalid or the resource definition is not specified for update.
330000E9	License is not valid.	License is not uploaded, or uploaded license is invalid or expired.
330000E9	Authorization needed.	Request is sent with an invalid authentication token.
330000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
330000E0	HTTP method [GET] is not supported for action [updateResourceDef].	Request is sent with any HTTP method other than the POST.
330000E2	Application encountered an internal error.	API server encountered an unknown error.

Deleting a resource definition

You can delete a resource definition. If the resource definition type is being used by another resource definition, then it is not deleted. If the resource definition does not exist, then an error message is received in the response message. You can delete multiple resource definitions in a single request.

You cannot delete resource definitions that are in a circular reference relation.

For example, resources A, B, and C are circularly referencing each other as follows: A=>B, B=>C, C=>A.

Here, A =>B implies that resource definition A is referenced by resource definition B. In this case, neither resource definition can be deleted.

Request line

```
baseUrl/dbapi.do?action=deleteResourceDef&dataset=<Datasetname>
```

Request body

```
{
  "types": [
    "",
    "",
    ""
  ]
}
```

Request example using GET

```
GET /dbapi.do?action=deleteResourceDef&dataset=defaultDs&types=host,vm,storage
HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

Request example using POST

```
POST /dbapi.do?action=deleteResourceDef&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "types": [
    "host",
    "vm",
    "storage"
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	deleteResourceDef Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
types	String[]	One or more comma-separated resource types. From 1 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Response body

```
{
  "deleted": [
    ""
  ],
  "failed": [
    {
      "code": "",
      "type": "",
      "error": ""
    }
  ]
}
```

Response example

```
{
  "deleted": [
    "vm"
  ],
  "failed": [
    {
      "code": "34050011",
      "type": "host",
      "error": "Resource definition deletion failed due to dependency on one or more other resources"
    },
    {
      "code": "34050005",
      "type": "storage",
      "error": "Resource definition with {type}: [storage] does not exist"
    }
  ]
}
```

Response parameters

Parameter	Type	Description
deleted	String[]	List of deleted resource types.
failed	Object[]	List of failed entities.
failed:code	String	Eight-digit hexadecimal error code.
failed:type	String	Failed resource type.

Parameter	Type	Description
failed:error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified resource definitions fail, then the 500 SERVER ERROR message is received in the response message.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
3402000F	Invalid value specified for parameter {type}. Resource {type}:[] contains invalid characters. Valid characters are [alphanumeric, - and _].	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. Specified resource type contains invalid characters.
34050008	Mandatory parameter {type} not specified.	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. The resource definition type is not specified as a request parameter.
3402000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. Specified resource definition type is not valid in terms of allowed length.
34050005	Resource definition with {type}:[] does not exist	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. Specified resource definitions type does not exist.
34050014	Cannot delete read-only 'Resource definition'	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. The resource definition corresponding to the specified resource definition type is pre-defined.

Response code	Message	Condition
34050011	Resource definition deletion failed due to dependency on one or more other resources	Minimum of one resource definition is deleted and the deletion of one or more resource definition types failed. Specified resource definition is dependent on some other resource definitions.
34030007	No Resource definition is specified	Parameter types are not specified or the specified parameter is not of type string.
34010002	Invalid JSON Request	Specified JSON input is invalid.
340000E9	License is not valid	License is not uploaded, or the uploaded license is invalid or expired.
340000E9	Authorization needed	Request is sent with an invalid authentication token.
34050005	Dataset with {name}:[ds1] does not exist	Request is sent for a dataset that does not exist.
340500E0	HTTP method [PUT] is not supported for action [deleteResourceDef]	Request is sent with any HTTP method other than POST.
340000E2	Application encountered an internal error	API server encountered an unknown error.

Chapter 5: Resource data

Resource data consists of either the scalar or timeseries data or both. Each resource data instance is associated with the time stamp when it is collected.

Creating or updating resource data

You can create resource data or update the existing resource data in a single request.

Request line

```
POST baseUrl/dbapi.do?action=createOrUpdateResourceData&dataset =<Datasetname>
```

Request body

```
{
  "meta":{
    "ts": "",
    "dataSubsetId": "",
    "fullSubsetData": ,
    "fullSubsetFor": [
      ""
    ]
  },
  "resources": [
    {
      "signature": "",
      "name": "",
      "memory": "",
      "cpuUsage": [
        {
          "from": "",
          "interval": ,
          "data": [
            , , ,
          ]
        }
      ],
      "relationsCompleteList": [
        ""
      ]
    },
  ],
}
```

```
{
  "signature": "",
  "name": "",
  "memory": "",
  "relationsAdded": [
    ""
  ]
}
```

Request example

```
POST /dbapi.do?action=createOrUpdateResourceData&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "meta":{
    "ts":"20140714_000000",
    "dataSubsetId":"DevOps_Lab",
    "fullSubsetData":true,
    "fullSubsetFor": [
      "vm"
    ]
  },
  "resources":[
    {
      "signature":"vm#devops-labvm1",
      "name":"devops-lab-vm1",
      "memory":"204800",
      "cpuUsage":[
        {
          "from":"20140714_003403",
          "interval":1,
          "data":[
            23.5, 33, 67.5, 4
          ]
        }
      ],
      "relationsCompleteList":[
        "h#devops-lab-host1"
      ]
    },
    {
      "signature":"vm#devops-labvm2",
      "name":"devops-lab-vm2",
      "memory":"204800",
      "relationsAdded":[
        "h#devops-lab-host2"
      ]
    }
  ]
}
```

```

    }
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	createOrUpdateResourceData. Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
meta	String	Contains meta information of the resource data.
meta.ts	String	Time in yyyyMMdd_HH:mm:ss format, at which resources specified in the body must be created or updated. Default time zone is UTC.
meta.dataSubsetId	String	The resource data subset ID logically categorizes the resource data.
meta.fullSubsetData	Boolean	This flag indicates if the request body is a snapshot. A snapshot contains all resources belonging to a data subset at a particular point in time. If the request body is a snapshot, the resources that are not in the snapshot request and are in the database are marked as deleted in the database. Default: False

Parameter	Type	Description
meta.fullSubsetFor	String[]	List of valid resource data types for the full snapshot request. If this parameter is specified, then <code>fullSubsetData</code> must be set to true.
resources	Object[]	Array of the resource data. Resource object includes a composite resource definition where both scalar and timeseries data can be specified.
resources.signature	String	<ul style="list-style-type: none"> Allowed characters for <resource-type> are: alphanumeric, <code>_</code>, and <code>-</code> Allowed characters for <unique-part> are: alphanumeric, <code>_</code>, <code>-</code>, <code>^</code>, <code>(</code>, <code>)</code>, <code>/</code>, <code>\</code>, <code>#</code>, <code>:</code>, <code>.</code> and space. Resource signature should be unique within a dataset. <p>Signature works as a unique key and is composed as <resource-type>#<unique-part>.</p> <p>For example: <resource-type> = vm <unique-part> = abc The signature is formed as vm#abc.</p>
resources<scalar attribute id>	String	Request body can have one or more scalar attribute data. From 1 to 4000 alphanumeric characters are allowed for scalar attribute data. Scalar attribute data is stored as a string value in the Analyzer detail view database. For example:

Parameter	Type	Description
		"name":"devops-lab-vm1", "memory":"204600",
resources.data <timeseries attribute Id>	String	The request body can have one or more timeseries attribute data. The timeseries data point is stored as a double value in the Analyzer detail view database. For example: "cpuUsage": [{ "from":"20140714_003403", "interval":60, "data":[23.5, 33, 67.5, 4] }],
resources.from	String	(Optional) Property that represents the start time in the UTC time zone of the time series sequence. If the <i>from</i> parameter is not specified, then <i>ts</i> is considered as the start time. Supported range of value is: 9223372036854775808 to 9223372036854775807 Values outside the range 4503599627370496 to 4503599627370496 might have precision loss.
resources.interval	Integer	Data interval, in seconds. A positive integer from 1 to 86400 is allowed. Data at a frequency of up to 1 day can be inserted.
resources.data <timeseries attribute Id> -> data	Double[]	Array of doubles.

Parameter	Type	Description
resources.<relations>	String[]	<p>Relation to a resource can be specified in three ways:</p> <ul style="list-style-type: none"> relationsCompleteList: List of resource signatures that are related to this resource. If this list is present, it overwrites all existing relations. relationsAdded: List of newly added relations. relationsRemoved: List of removed relations. <p>If relationsCompleteList is present then relationsAdded and relationsRemoved are not considered.</p>
utcOffset	String	<p>(Optional) Offset from the Coordinated Universal Time (UTC), is applicable for the API request and response.</p> <p>UTC+/-HH:MM</p> <p>For example: UTC+09:00</p> <p>Default: UTC</p>



Note: If you specify an invalid value type for any optional parameter, then that parameter is not considered.

The resource definition must have only scalar or time series data.

Response body

```
{
  "updated":,
  "failed":[
    ""
  ]
}
```

Response example

Resources data successfully created:

```
{
  "updated":2
}
```

One or more resources data successfully created:

```
{
  "updated":1,
  "failed":[
    "vm#devops-labvm2"
  ]
}
```

Failed to create all resources data:

```
{
  "updated":0,
  "failed":[
    "vm#devops-labvm2"
  ]
}
```

Response parameters

Parameter	Type	Description
updated	Integer	Number of created or updated resources
failed	Object[]	List of failed entities.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: If all of the specified resource data fail, then the 500 SERVER ERROR message is received in the response message.

For the 277 PARTIAL SUCCESS message (where a minimum of one resource data is created and the creation of one or more resource data failed), one of the following error messages is shown in the response body.

Message	Condition
{ "updated" : 1 , "failed" : ["null"] }	The resource signature is not specified for a resource.
{ "updated" : 1 , "failed" : [""] }	The resource data contains an empty resource signature.
{ "updated" : 1 , "failed" : ["vm#devops-lab-vm!!"] }	The resource signature contains invalid characters.
{ "updated" : 1 , "failed" : ["devops-lab-vm"] }	The resource signature is not prefixed with the resource type.

Message	Condition
{ "updated" : 1 , "failed" : ["disk#devops-lab-vm"] }	The resource definition is not present for the specified resource type.
{ "updated" : 1 , "failed" : ["vm#devops-lab-vm!!"] }	The related resource signature contains invalid characters.
{ "updated" : 1 , "failed" : ["vm#devops-lab-vm"] }	The scalar data value is not valid in terms of allowed length.
{ "updated" : 1 , "failed" : ["vm#devops-lab-vm"] }	The interval for any timeseries data is not in valid range.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

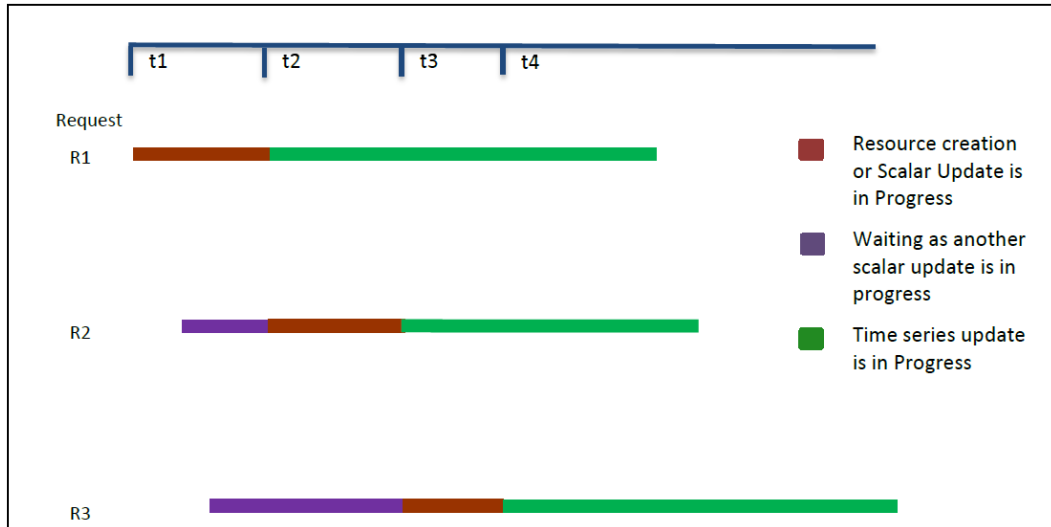
Response code	Message	Condition
51010002	Invalid JSON Request.	Specified JSON input is invalid.
51010008	Mandatory parameter {meta} not specified.	Parameter <code>meta</code> is not specified in the request body.
5101000F	Invalid value specified for parameter {meta}.	Specified <code>meta</code> is not of type JSON object.
5101000F	Invalid value specified for parameter {ts}.	Parameter <code>ts</code> is not specified or is invalid.
5101000F	Invalid value specified for parameter {ts}.	Specified date in <code>ts</code> is not in valid range.
5101000F	Invalid value specified for parameter {dataSubsetId}.	Parameter <code>dataSubsetId</code> is not specified or is not of type JSON string.
51010005	DataSubset with {id}:[ds11] does not exist.	Parameter <code>dataSubsetId</code> with the specified <code>dataSubsetId</code> does not exist.
5101000F	Invalid value specified for parameter {fullSubsetData}.	Specified <code>fullSubsetData</code> parameter is not of type boolean.

Response code	Message	Condition
5101000F	Invalid value specified for parameter {fullSubsetFor}.	Specified <code>fullSubsetFor</code> parameter is not of type string.
51010008	Mandatory parameter {resources} not specified.	Parameter <code>resources</code> is not specified.
5101000F	Invalid value specified for parameter {resources}.	Specified resources are not of type JSON list.
5101000F	Invalid value specified for parameter {fullSubsetData}. If {fullSubsetFor} is specified, then {fullSubsetData} must be true.	Parameter <code>fullSubsetFor</code> is specified but the parameter <code>fullSubsetData</code> is false or not specified.
5101000F	Invalid value specified for parameter {fullSubsetFor}. If {fullSubsetFor} is specified, then it must not be empty.	Specified <code>fullSubsetFor</code> is an empty JSON list.
51010006	Resource definition [vm] does not exist.	Specified resource type in <code>fullSubsetFor</code> is not defined.
5101000F	Invalid value specified for parameter {signature}.	Resource signature is not type string.
510000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
510000E9	Authorization needed.	Request is sent with an invalid authentication token.
510000E4	Dataset with {name};[ds1] does not exist.	Request is sent for a dataset that does not exist.
510000E0	HTTP method [GET] is not supported for action [createOrUpdateResourceData].	Request is sent with any HTTP method other than POST.
510000E2	Application encountered an internal error.	API server encountered an unknown error.

Locking the request at resource level

While processing multiple concurrent `createOrUpdateResourceData` requests, the Analyzer detail view REST API applies locking at the resource level.

The following diagram shows when three requests are processed concurrently and each request contains a different resource to create or update.

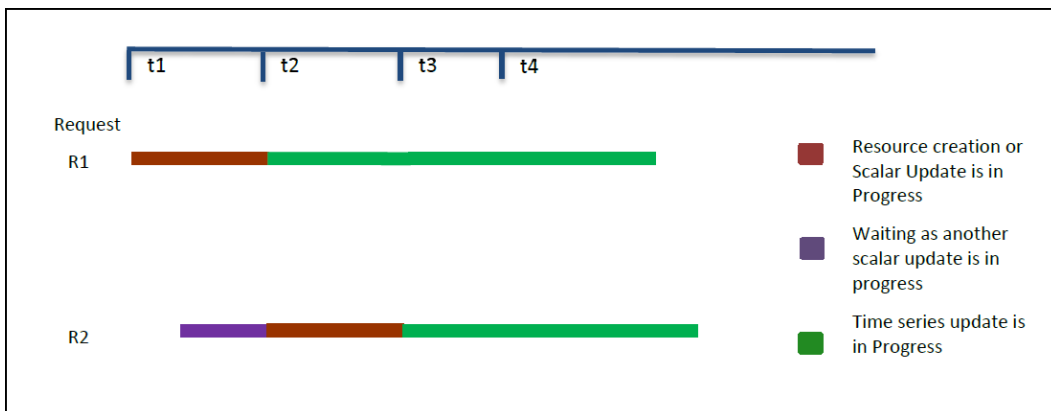


At time t1, the API server receives request R1 and starts creating the resource or updating its scalar data. At the same time, the API server receives requests R2 and R3, and while R1 is in progress, requests R2 and R3 are locked.

At time t2, when the resource creation and scalar update of request R1 is complete, the API server starts the time series update. At the same time the API server request R2 and starts creating the resource or updating the scalar data, request R3 is locked.

At time t3, when the resource creation and scalar update of request R2 is complete, the API server processes request R3 in parallel with the time series update of requests R1 and R2.

The following diagram shows when two requests are being processed concurrently and both requests contain the same resource to create or update.

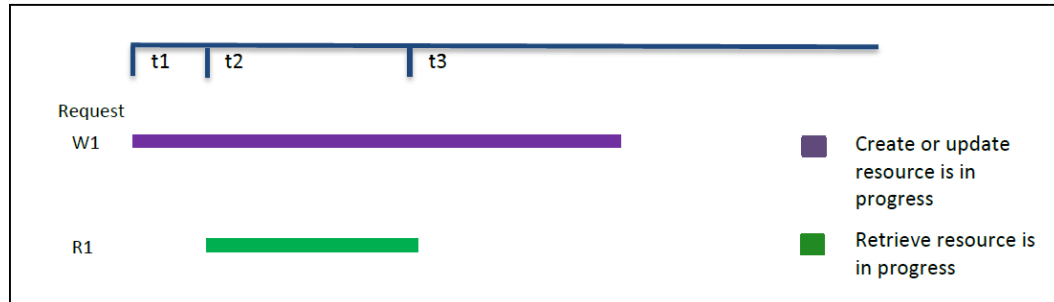


For this scenario, where both requests contain the same resource to create and update, the API server applies the lock as it did in the previous scenario.

At time t1, the API server receives request R1 and starts creating the resource or updating its scalar data. At the same time, request R2 is received. However, request R2 is locked while request R1 is in progress.

At time t2, when the resource creation and scalar update of request R1 is complete, the API server starts the time series update. At the same time, series of request R1, the API server starts request R2. As the resource is already created by request R1, request R2 only updates the scalar and time series data.

The following diagram shows when two requests are being processed concurrently. The first request contains a resource to create or update, while the second request attempts to retrieve the resource data for the same resource.



At time t1, the API server receives request W1 to create or update a resource data and it begins processing.

At time t2, the API server receives request R1 to retrieve the same resource, which is being created in another request. Request R1 is processed in parallel with request W1 and the API server attempts to retrieve the resource. In this case, partial results might be returned.

Time series data points

An attribute can have a positive value, a negative value, or both.

When creating the time series attribute definition, you can specify the data type that it can hold with the `bandFactor` parameter, which determines the precision (number of decimal places) of stored value.

The possible `bandFactor` values are 1, 0.1, 0.01, 0.001, and 0.0001.

The `bandFactor` is applied to the data value by dividing data value by `bandFactor`. The resulting number is then rounded to a whole number. This whole number is called a normalized value, and is stored in the database.

The data range and limitations for time series data value are:

- Supported range of normalized values is from 9223372036854775808 to 9223372036854775807.
- Values outside the range of 4503599627370496 to 4503599627370496 can have precision loss.

Any invalid time series data value (string or value beyond the supported range) is stored as a data hole in the Analyzer detail view database.

Time series data interval

The start time must be the multiple of the data interval. If not, then the next highest multiple of the data interval is considered as the start time.

For example, the data interval is 5 minutes and the start time is 20150322_051700, which is not a multiple of 5 minutes. In this case, the Analyzer detail view database shifts the start time to the next multiple of 5 minutes, which is 20150322_052000.

Data for the same resource and the same time series counter can be added with a different data interval in the same time range.

For example, insert data for different intervals (1 and 60) for the same resource, counter, and time:

```
{
  "signature": "vm#devops-labvm1",
  "name": "devops-lab-vm1",
  "cpuUsage": [
    {
      "from": "20141217_020000",
      "interval": 1,
      "data": [
        1, 2, 3, 0
      ]
    },
    {
      "from": "20141217_020000",
      "interval": 60,
      "data": [
        23.5, 33, 67.5, 4
      ]
    }
  ]
}
```

Deleting resource data

You can delete specific resource data by specifying it in the input criteria.

Request line

```
POST baseUrl/dbapi.do?action=deleteResourceData&dataset=<Datasetname>
```

Request body

```
{
  "dataSubsetId": "",
  "resType": "",
  "attrId": ""
}
```

```
"attrValue":""
}
```

Request example

```
POST /dbapi.do?action=deleteResourceData&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "resType":"host",
  "attrId":"name",
  "attrValue":"Host1"
}
```

Request parameters

Parameter	Type	Description
action	String	deleteResourceData Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
resSignatures	String[]	Optional) One or more comma-separated resource signatures.
dataSubsetId	String	(Optional) The resources data subset ID.
resType	String	(Optional) The resource data type. From 1 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrId	String	(Optional) Attribute ID of type scalar. From 2 to 3 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Parameter	Type	Description
attrValue	String	(Optional) Value for the specified attribute ID. From 1 to 4,000 alphanumeric characters are allowed.

You must specify at least one of following parameters:

- resSignatures
- dataSubsetId
- resType
- attrId and attrValue

If the `resSignatures` parameter is specified, then the following parameters are not considered: `dataSubsetId`, `resType`, `attrId`, and `attrValue`.

Response body

```
{
  "deleted":
}
{
  "code": "",
  "error": ""
}
```

Response example

```
{
  "deleted": 2
}
```

Response parameters

Parameter	Type	Description
deleted	Integer	If the resource data is deleted, shows the list of deleted resources.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
54010002	Invalid JSON Request.	Specified JSON input is invalid.

Response code	Message	Condition
54020009	At least one of {resType}, {attrId}, {attrValue}, {dataSubsetId}, or {resSignatures} parameters must be specified	Required parameters are specified, or the specified parameters are not sufficient.
5401000F	Invalid value specified for parameter {resSignatures}. It must be of type JSON list of string.	Specified <code>resSignatures</code> is not JSON list of string.
5402000F	Invalid value specified for parameter {resSignatures}.	Specified <code>resSignatures</code> is an empty list.
5402000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.	Parameter <code>attrId</code> is specified but <code>attrValue</code> is not specified.
5402000F	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.	Parameter <code>attrValue</code> is specified but <code>attrId</code> is not specified.
5402000F	Invalid value specified for parameter {resType}	Specified value for parameter <code>resType</code> is not of type string.
5402000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.	Specified value for <code>resType</code> contains an invalid length.
5402000F	Invalid value specified for parameter {resType}. Resource {type}:[#] contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified value for <code>resType</code> contains an invalid character.
5402000F	Invalid value specified for parameter {attrId}	Specified value for parameter <code>attrId</code> is not of type string.

Response code	Message	Condition
5402000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2-32 characters.	Specified value for attrId contains an invalid length.
5402000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified value for attrId contains an invalid character.
5402000F	Invalid value specified for parameter {attrValue}	Specified value for parameter attrValue is not of type string.
5402000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1-4000 characters.	Specified value for attrValue has an invalid length.
5402000F	Invalid value specified for parameter {dataSubsetId}.	Specified value for parameter dataSubsetId is not of type string.
5402000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} length must be between 1-512 characters.	Specified value for dataSubsetId contains an invalid length.
54000005	Resource definition with {resType}:[vm] does not exist.	Resource definition does not exist with the specified resType parameter.
54000005	Attribute definition with {attrId}:[name] does not exist.	Attribute definition does not exist with the specified attrId parameter.
5400000F	Invalid value specified for parameter {attrId}. Specified Attribute definition is not of type scalar.	Specified attrId is not of type scalar.
54050005	DataSubset with {dataSubsetId}:[ds1] does not exist.	DataSubset does not exist with the specified dataSubsetId parameter.

Response code	Message	Condition
540000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
540000E9	Authorization needed .	Request is sent with an invalid authentication token.
540000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
54000005	Resource definition with {resType}:[h] does not exist.	Specified <code>resType</code> does not exist.
54000005	Attribute definition with {attrId}:[abc] does not exist.	Specified <code>attrId</code> does not exist.
54050005	DataSubset with {dataSubsetId}:[ds] does not exist.	Specified <code>dataSubsetId</code> does not exist.
540000E0	HTTP method [GET] is not supported for action [deleteResourceData].	Request is sent with any HTTP method other than POST.
540000E2	Application encountered an internal error .	API server encountered an unknown error.

Getting resource data information

You can obtain the following resource data information: start time, signature, relations, and scalar data. If the requested resource does not exist, then an error message is received in the response message.

Request line

```
baseUrl/dbapi.do?action=retrieveResourceData&dataset=<Datasetname>
```

Request body

```
{
  "resSignatures": [
    "",
    ""
  ]
  "resType": ""
  "attrId": ""
  "attrValue": ""
}
```

```
"snapshotTime":
}
```

Request example using GET

```
GET /dbapi.do?action=retrieveResourceData&dataset=defaultDs&resSignatures=vm#devops-
lab-vm1,h#devops-lab-host1 HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

The resource signatures in the query parameter, # must be replaced with its encode value, which is %23. So, the above request example will be:

```
GET /dbapi.do?action=retrieveResourceData&dataset=defaultDs&resSignatures=vm%23devops-
lab-vm1,h%23devops-lab-host1 HTTP /1.1
```

Request example using POST

```
POST /dbapi.do?action=retrieveResourceData&dataset=defaultDs HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "resSignatures":[
    "vm#devops-lab-vm1",
    "h#devops-lab-host1"
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	retrieveResourceData Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
resSignatures	String[]	(Optional) One or more comma-separated resource signatures.
resType	String	(Optional) Resource type. From 1 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Parameter	Type	Description
attrId	String	(Optional) Attribute ID of type scalar. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrValue	String	(Optional) Value of the specified attribute ID. From 1 to 4000 alphanumeric characters are allowed.
snapshotTime	String	(Optional) Snapshot time to get the resources data in yyyyMMdd_HH:mm:ss format. Default time zone is UTC and the default time is the current time of the Analyzer detail view server.

You must specify one of the following parameters:

- resSignatures
- resType
- attrId and attrValue

If the `resSignatures` is specified, then the `resType`, `attrId`, and `attrValue` parameters are ignored.

The parameters `resType`, `attrId`, and `attrValue` can be used in a combination or separately.

Response body

```
{
  "retrieved": [
    {
      "startTime": "",
      "signature": "",
      "name": "",
      "memory": "",
      "relations": [
        ""
      ],
      "endTime": ""
    }
  ]
}
```

```
{
  "code": "",
  "error": ""
}
```

Response example

```
{
  "retrieved": [
    {
      "startTime": "20140714_000000",
      "signature": "vm#devops-vm1",
      "name": "devops-lab-vm1",
      "memory": "204600",
      "relations": [
        "h#devops-lab-host1"
      ]
    },
    {
      "startTime": " 20140714_000000",
      "signature": "h#devops-lab-host1",
      "name": "devops-lab-host1",
      "relations": [
        "vm#devops-lab-vm1"
      ]
    }
  ]
}
```



Note: The `startTime` is when the resource appeared for the first time, in the UTC time zone. Every resource has a `startTime` associated with it. A resource can also have an `endTime` when the resource expired, also in the UTC time zone.

Response parameters

Parameter	Type	Description
retrieved	Object[]	List of resource data.
retrieved.signature	String	Resource signature.
retrieved.name	String	Value of the scalar attribute, either name or value of the signature.
retrieved.<scalar attribute>	String	Attributes data.

Parameter	Type	Description
retrieved.relations	String[]	List of related resource signatures.
retrieved.startTime	String	UTC time when the resource was created.
retrieved.endTime	String	UTC time when the resource was created.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Status code	Message	Description
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
52010002	Invalid JSON Request.	Specified JSON input is invalid.
52010009	At least one of {resSignatures}, {resType}, {attrId}, or {attrValue} parameters must be specified.	Required parameters are not specified, or the specified parameters are not sufficient.
5201000F	Invalid value specified for parameter {resSignatures}. It must be of type JSON list of string.	Specified <code>resSignatures</code> is not of type JSON list of string.
5201000F	Invalid value specified for parameter {resType}. It must be of type string.	Specified <code>resType</code> is not of type string.
5201000F	Invalid value specified for parameter {attrId}. It must be of type string.	Specified <code>attrId</code> is not of type string.
5201000F	Invalid value specified for parameter {attrValue}. It must be of type string.	Specified <code>attrValue</code> is not of type string.
5201000F	Invalid value specified for parameter {attrValue}. It must be of type string.	Specified <code>attrValue</code> is not of type string.
5201000F	Invalid value specified for parameter {snapshotTime}. It must be of type string.	Specified <code>snapshotTime</code> is not of type string.

Response code	Message	Condition
5201000F	Invalid value specified for parameter {snapshotTime}. Parameter {snapshotTime} must be in yyyyMMdd_HH:mm:ss format.	Specified <code>snapshotTime</code> is not in valid format.
5201000F	Invalid value specified for parameter {snapshotTime}	Specified <code>snapshotTime</code> is not in valid range.
5202000F	Invalid value specified for parameter {resSignatures}	Specified <code>resSignatures</code> list is empty.
5202000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.	Parameter <code>attrId</code> is specified but <code>attrValue</code> is not specified.
	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.	Parameter <code>attrValue</code> is specified but <code>attrId</code> is not specified.
5202000F	Invalid value specified for parameter {resType}. Resource {type}:[h\$%] contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified <code>resType</code> contains invalid characters.
5202000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.	Specified <code>resType</code> is not valid in terms of length.
5202000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2 to 32 characters.	Specified <code>attrId</code> is not valid in terms of length.
5202000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified <code>attrId</code> contains invalid characters.

Response code	Message	Condition
5202000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1 to 4000 characters.	Specified attrValue is not valid in terms of length.
52000005	Resource definition with {resType}:[vm] does not exist.	Resource definition does not exist for specified resType.
52000005	Attribute definition with {attrId}:[name] does not exist.	Attribute definition does not exist for specified attrId.
5200000F	Invalid value specified for parameter {attrId}. Specified Attribute definition is not of type scalar.	Specified attrId is not of type scalar.
520000E9	License is not valid .	License is not uploaded, or the uploaded license is invalid or expired .
520000E9	Authorization needed.	Request is sent with an invalid authentication token.
520000E4	Dataset with {name}:[ds1] does not exist.	Request sent for a dataset that does not exist.
520600E6	Requested resource is not found.	Requested resource is not found.
520000E4	HTTP method [PUT] is not supported for action [retrieveResourceData].	Request is sent with any HTTP method other than GET or POST.
520000E2	Application encountered an internal error.	API server encountered an unknown error.

Deactivating the resource data

You can deactivate specific resource data by specifying it in input criteria. If the requested resource does not exist, then an error message is received in a response message.

Request line

```
baseUrl/dbapi.do?action=expireResourceData&dataset=<Datasetname>
```

Request body

```
{
  "resSignatures": [
    "",
    ""
  ]
  "resType": ""
  "attrId": ""
  "attrValue": ""
  "endTime": ""
}
```

Request example using GET

```
GET /dbapi.do?action=expireResourceData&dataset=defaultDs&resSignatures=vm%23devops-
lab-vm1,h%23devops-lab-host1 HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

Request example using POST

```
Post /dbapi.do?action=expireResourceData&dataset=defaultDs&resSignatures=vm%23devops-
lab-vm1,h%23devops-lab-host1 HTTP /1.1 Authorization:Basic ZnJlZDpmcmVk
```

```
{
  "resSignatures": [
    "vm#devops-lab-vm1",
    "h#devops-lab-host1"
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	expireResourceData Specify the API function to be invoked
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
resSignatures	String[]	(Optional) One or more comma separated resource signatures.

Parameter	Type	Description
resType	String	(Optional) Resource type. From 1 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrId	String	(Optional) Attribute ID of the type scalar. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
attrValue	String	(Optional) Value for the specified attribute ID. From 1 to 4000 alphanumeric characters are allowed.
endTime	String	(Optional) End time to terminate the resources data in the yyyyMMdd_HH:mm:ss format. Default time zone is on UTC and default time is the current time of the Analyzer detail view server.

You must specify one of the following parameters:

- resSignatures
- resType
- attrId and attrValue

If the `resSignatures` parameter is specified, then the `resType`, `attrId`, and `attrValue` parameters are ignored.

The parameters `resType`, `attrId`, and `attrValue` can be used in a combination or separately.

Response body

```
{
  "expired":2
}
```


Response example

```
{  
  "expired":2  
}
```

Response parameters

Parameter	Type	Description
expired	Integer	If resource data get is terminated, then shows the number of terminated resources.
failed	Object[]	List of failed entities.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.

Status code	Message	Description
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
56010002	Invalid JSON Request.	Specified input JSON is invalid.
56010009	At least one of {resSignatures}, {resType}, {attrId}, {attrValue} parameters must be specified.	None of the required parameter are specified or specified parameters are not sufficient.
5601000F	Invalid value specified for parameter {attrValue}. It must be of type string.	Specified attrValue is not of type string.
5601000F	Invalid value specified for parameter {attrId}. It must be of type string.	Specified attrId is not of type string.
5601000F	Invalid value specified for parameter {resType}. It must be of type string.	Specified resType is not of type string.
5601000F	Invalid value specified for parameter {resSignatures}. It must be of type JSON list of string.	Specified resSignatures is not type JSON list of string.
5602000F	Invalid value specified for parameter {resSignatures}.	Specified resSignatures is an empty list.

Response code	Message	Condition
5602000F	Invalid value specified for parameter {resType}. Resource {type}:[h\$%] contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified value for <code>resType</code> contains an invalid character.
5602000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.	Specified value for <code>resType</code> has an invalid length.
5602000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.	Parameter <code>attrId</code> is specified but <code>attrValue</code> is not specified.
5602000F	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.	Parameter <code>attrValue</code> is specified but <code>attrId</code> is not specified.
5602000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2-32 characters.	Specified value for <code>attrId</code> has an invalid length.
5602000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].	Specified value for <code>attrId</code> contains an invalid character.
5602000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1 to 4000 characters.	Specified value for <code>attrValue</code> has an invalid length.
5601000F	Invalid value specified for parameter {endTime}	Invalid value specified for parameter <code>endTime</code> .
560000E9	License is not valid.	License is not uploaded or uploaded license is invalid or expired.

Response code	Message	Condition
560000E9	Authorization needed.	Request is sent with an invalid authentication token.
56000005	Dataset with {name}:[ds1] does not exist.	Request sent for a dataset that does not exist.
56060000	Not found.	No resource matched for given delete request.
560000E0	HTTP method [PUT] is not supported for action [expireResourceData].	Request is sent with any HTTP method other than GET or POST.
560000E2	Application encountered an internal error.	API server encountered an unknown error.

Chapter 6: Dataset API resources

A dataset includes the schema and data. You can obtain the dataset information or delete a specific dataset.

See [Analyzer detail view database schema \(on page 8\)](#) for more information.

Getting information about a dataset

You can obtain the following information about a dataset: number of attribute definitions, resource definitions, total resource count, and the list of data subset IDs.

Request line

```
GET baseURL/dbapi.do?action=retrieveDataset&dataset=<Datasetname>
```

Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	{retrieveDataset} Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.

Request example

```
GET /dbapi.do?action=retrieveDataset&dataset=defaultDs HTTP /1.1 Authorization:Basic  
ZnJlZDpmcmVk
```

Response body

```
{
  "name": "",
  "attributeDefCount": ,
  "resourceDefCount": ,
  "resourceCount": ,
  "dataSubset":
}
{
  "code": "",
  "error": ""
}
```

Response example

```
{
  "name": "defaultDs",
  "attributeDefCount": 100,
  "resourceDefCount": 150,
  "resourceCount": 1000,
  "dataSubset": [
    "usSubset",
    "ukSubset",
    "asSubset"
  ]
}
```

Response parameters

Parameter	Type	Description
name	String	Dataset name.
attributeDefCount	Integer	Number of attribute definitions for the given dataset.
resourceDefCount	Integer	Number of resource definitions for the given dataset.
resourceCount	Integer	Number of resources for the given dataset.
dataSubset	String	List of data subset IDs for the given dataset.
code	String	Eight-digit hexadecimal error code.

Parameter	Type	Description
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Status code	Message	Description
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
120000E4	Mandatory parameter {dataset} is not specified.	The <code>dataset</code> parameter is not specified.
120000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
120000E9	Authorization needed.	Request is sent with an invalid authentication token.
120000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a <code>dataset</code> that does not exist.
120000E0	HTTP method [POST] is not supported for action [retrieveDataset].	Request is sent with an HTTP method other than GET.
120000E2	Application encountered an internal error.	API server encountered an unknown error.

Deleting the dataset

You can delete the data of a specific dataset. You can also delete the schema from the dataset by setting the request parameter to true: `clearSchema=true`.

Request line

```
GET baseUrl/dbapi.do?action=clearDataset&dataset=<Datasetname>&clearSchema=<true | false>
```


Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	{clearDataset} Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
clearSchema	Boolean	(Optional) Indicates if the schema must be deleted. true or false Default : false

Request example

```
GET /dbapi.do?action=clearDataset&dataset=defaultDs HTTP /1.1 Authorization:Basic  
ZnJlZDpmcmVk
```

Response body

```
{  
  "cleared":""  
}  
{  
  "code":"","  
  "error":""  
}
```

Response example

```
{  
  "cleared":"defaultDs"  
}
```

Response parameters

Parameter	Type	Description
cleared	String	Cleared dataset name.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
150000E4	Mandatory parameter {dataset} is not specified.	The <i>dataset</i> parameter is not specified.
1502000F	Invalid value specified for parameter {clearSchema}.	Invalid value is specified for the <i>clearSchema</i> parameter.
150000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
150000E9	Authorization needed.	Request is sent with an invalid authentication token.
150000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
150000E0	HTTP method [POST] is not supported for action [clearDataset.]	Request is sent with an HTTP method other than GET.
150200E2	Application encountered an internal error.	API server encountered an unknown error.

Chapter 7: Request API resources

This section describes how to use REST API to get information such as the status of an asynchronous job, audit logs, and Analyzer detail view server information including, server locale, time zone, and database version.

Getting the request status

You can obtain the status of an asynchronous job. The status consists of the available request completion status and results. If the requested status does not exist, then an error message is received in a response message.

Request line

```
GET baseURL/dbapi.do?action=getStatus&dataset=<Datasetname>&requestId=<reqId>
```

Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	getStatus Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
requestId	String	Asynchronous request ID.

Request example

```
GET /dbapi.do?action=getStatus&dataset=defaultDs&requestId=12344 HTTP /1.1  
Authorization:Basic ZnJlZDpmcmVk
```

Response body

```
{
  "status":{
    "total":,
    "matched":,
    "processed":,
    "completed":,
    "aborted":,
    "hasErrors":""
  },
  "result":[
    {
      "signature": "",
      "name": "",
      "memory":{
        "type": "",
        "name": "",
        "unit": "",
        "data": ""
      },
      "cpu Usage":[
        {
          "type": "",
          "name": "",
          "unit": "",
          "interval":,
          "start": "",
          "data":[
            , , , ,
          ]
        }
      ]
    }
  ]
}
```

Response example

```
{
  "status":{
    "total":10,
    "matched":1,
    "processed":10,
    "completed":true,
    "aborted":false,
    "hasErrors":"false"
  },
  "result":[
    {
      "signature":"vm#devops1-lab-vm2",
```

```

    "name": "devops-lab-vm2",
    "memory": {
      "type": "scalar",
      "name": "Memory",
      "unit": "",
      "data": "204800"
    },
    "cpu Usage": [
      {
        "type": "timeseries",
        "name": "cpu usage",
        "unit": "",
        "interval": 60,
        "start": "20140713_230100",
        "data": [
          24.00, 33.00, 68.00, null, 4.00
        ]
      }
    ]
  }
]
}

```

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.

Status code	Message	Description
404	NOT FOUND	API server could not find a resource matching the request.
500	SERVER ERROR	API operation request failed. Check the response body for details.



Note: This API gets the status of other API requests. It can also get all the error conditions of the original API requests.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
72020008	Mandatory parameter {requestId} not specified.	Parameter <code>requestId</code> is not specified.
720600E6	Response is not present in the cache for the given <i>requestId</i> .	Response is not cached on the API server for the given request ID.
720000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
720000E9	Authorization needed.	Request is sent with an invalid authentication token.
720600E6	Response is not present in the cache for the given <i>requestId</i> .	Response is not cached on the API server for the given request ID.
720600E2	Application encountered an internal error.	API server encountered an unknown error.

Getting an audit log

You can request audit logs based on a specific search criteria. The audit log contains the following information: date, request ID, dataset, user, user agent, HTTP method, client IP, action, request length, response code, and service time.

Request line

```
POST baseUrl/dbapi.do?action=getAuditLogs&dataset=<Datasetname>
```

Request body

```
{
  "tw": "",
  "requestId": "",
  "user": "",
  "clientId": "",
  "responseCode": "",
  "serviceTime": [
    ,
  ]
}
```

Request example

```
POST /dbapi.do?action=getAuditLogs&dataset=defaultDs HTTP /1.1 Authorization: Basic
ZnJlZDpmcmVk
```

```
{
  "user": "admin",
  "responseCode": "200",
  "serviceTime": [
    10, 20
  ]
}
```

Request parameters

Parameter	Type	Description
action	String	getAuditLogs Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
tw	String	(Optional) JSON list of string where the first element represents the audit start time and the second element

Parameter	Type	Description
		represents end time for audit. Date format: yyyyMMdd_HH:mm:ss Time zone: UTC By default the last 24 hours are considered for audit.
requestId	String	Request ID of the requested audit log. Default: *
user	String	User name of the requested audit log .If you want audit logs for the specific username, then do not use the regular expression special characters in the user name. Default: *
clientIp	String	Client IP address of the requested audit log. Default: *
responseCode	String	HTTP response code of the requested audit log. Default: *
serviceTime	Integer	(Optional) Service time range of the requested audit log. List of service time where elements in the list represent the minimum and maximum service time. Default: 0 (Java integer maximum value)

Response body

```
[
  {
    "date": "",
```

```

    "requestId": "",
    "dataset": "",
    "user": "",
    "userAgent": "",
    "HTTPMethod": "",
    "clientIP": "",
    "action": "",
    "requestLength": "",
    "responseCode": ,
    "serviceTime":
  }
]

```

Response example

```

[
  {
    "date": "20150615_092909",
    "requestId": "363041533_8",
    "dataset": "defaultDs",
    "user": "admin",
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/43.0.2357.124 Safari/537.36",
    "HTTPMethod": "POST",
    "clientIP": "192.168.1.157",
    "action": "createResourceDef",
    "requestLength": "375",
    "responseCode": 200,
    "serviceTime": 14
  },
  {
    "date": "20150615_100850",
    "requestId": "924518978_17",
    "dataset": "defaultDs",
    "user": "admin",
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/43.0.2357.124 Safari/537.36",
    "HTTPMethod": "POST",
    "clientIP": "192.168.1.157",
    "action": "getAuditLogs",
    "requestLength": "12",
    "responseCode": 200,
    "serviceTime": 11
  },
  {
    "date": "20150615_124609",
    "requestId": "213068087_52",
    "dataset": "defaultDs",
    "user": "admin",
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/43.0.2357.124 Safari/537.36",

```

```

    "HTTPMethod": "POST",
    "clientIP": "192.168.1.157",
    "action": "createResourceDef",
    "requestLength": "375",
    "responseCode": 200,
    "serviceTime": 10
  }
]

```

Response parameters

Parameter	Type	Description
date	String	Time stamp of the request. Format: yyyyMMdd_HH:mm:ss Time zone: UTC
requestId	String	Request ID.
dataset	String	Dataset name.
user	String	User name that requested the request.
userAgent	String	User agent name from which the request is received.
HTTPMethod	String	HTTP method used for the request.
clientIP	String	Client IP address from where this API request is received.
action	String	Action used for this API request.
requestLength	String	Request length.
responseCode	Integer	HTTP response code.
serviceTime	Integer	Total time required to execute the request.

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
82010002	Invalid JSON Request.	Specified JSON input is invalid.
8201000C	Parameter {query, startTime, endTime} not supported Parameter {query, startTime, endTime} are not supported.	Invalid parameters.
8201000F	Invalid value specified for the parameter {tw}.	Specified value for the parameter <code>tw</code> is not of a JSON list of string.
8201000F	Invalid value specified for the parameter {tw}. The parameter [tw] must have only two elements.	Specified JSON list for the parameter <code>tw</code> does not contain 2 elements.

Response code	Message	Condition
8201000F	Invalid value specified for parameter {tw}. The parameter [tw] must contain start and end time in yyyyMMdd_HHmss format.	Specified JSON list for the parameter tw does not contain start time or end time in valid format .
8201000F	Invalid value specified for parameter {tw}. The endTime must be after the startTime.	Specified endTime is before the specified startTime.
8201000F	Invalid value specified for the parameter {requestId}.	Specified value for the parameter requestId is not of type string.
8201000F	Invalid value specified for the parameter {dataset}.	Specified value for the parameter dataset is not of type String.
8201000F	Invalid value specified for the parameter {user}.	Specified value for the parameter user is not of type string.
8201000F	Invalid value specified for the parameter {clientIP}.	Specified value for the parameter clientIP is not of type string.
8201000F	Invalid value specified for the parameter {action}.	Specified value for the parameter action is not of type string.
8201000F	Invalid value specified for the parameter {responseCode}.	Specified value for the parameter responseCode is not of type string.
8201000F	Invalid value specified for the parameter {serviceTime}.	Specified value for the parameter serviceTime is not JSON list of integer.
8201000F	Invalid value specified for the parameter {serviceTime}. The parameter [serviceTime] must have two element.	Specified JSON list for the parameter serviceTime does not contain two elements.
8201000F	Invalid value specified for parameter {serviceTime}. The maximum serviceTime must be greater than minimum.	Specified minimum service time is greater than maximum service time.

Response code	Message	Condition
8201000C	Parameter {userAgent} is not supported.	Unsupported parameter specified.
820000E2	Application encountered an internal error.	API server encountered an unknown error.

Stopping an API request

You can stop a running asynchronous job on the server. If the requested job does not exist, then an error message is received in a response message.

Request line

```
GET baseUrl/dbapi.do?action=abortRequest&dataset=<Datasetname>&requestId=<requestId of API>
```

Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	abortRequest Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
requestId	String	Asynchronous request ID.

Request example

```
GET /dbapi.do?action=abortRequest&dataset=defaultDs&requestId=316829382_14 HTTP /1.1
Authorization:Basic ZnJlZDpmcmVk
```

Response body

If the request is stopped successfully:

```
{
  "aborted": true,
  "id": "316829382_14"
}
```

The request is processed, but not stopped:

```
{
  "aborted": false,
  "id": "316829382_14"
}
```

Cannot find the request ID in the DBresponseCache:

```
{
  "code": "B5040016",
  "error": "Response is not present in cache for the given requestId '316829382_14'"
}
```

If the request for the specified requestId is already stopped:

```
{
  "code": "B5040018",
  "error": "Request for given requestId '316829382_14' is already aborted"
}
```

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.

Status code	Message	Description
404	NOT FOUND	API server could not find a resource matching the request.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
B5020008	Mandatory parameter {requestId} is not specified.	Parameter <code>requestId</code> is not specified.
B50000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
B50000E9	Authorization is required.	Request is sent with an invalid authentication token.
B5040016	Response is not present in the cache for the given request ID.	Response is not cached on the API server for the given request ID.
B5040017	Request for the given request ID is already completed.	Request for the given request ID is already completed.
B5040018	Request for the given request ID is already stopped.	Request for the given request ID is already stopped.
B50600E2	Application encountered an internal error	API server encountered an unknown error.

Getting the Analyzer detail view server information

You can obtain the following server information: server locale, time zone, database version, and so on.

Request line

```
GET baseUrl/dbapi.do?action=getServerInfo
```

Request body

Not applicable.

Request parameters

Parameter	Type	Description
action	String	getServerInfo Specify API function to be invoked .
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
includeVersion	Boolean	Specify if you want to get the version details of the database and REST API or not. True or false Default: false

Request example

```
GET /dbapi.do?action=getServerInfo&dataset=defaultDs HTTP /1.1 Authorization:Basic  
ZnJlZDpncmVk
```

Response body

```
{  
  "uuid": "",  
  "appVersion": "",  
  "dbVersion": "",  
  "apiVersion": {  
    "dbApiVersion": "",  
    "alertApiVersion": ""  
  },  
  "locale": "",  
  "timeZone": "",  
  "hostName": "",  
  "ipAddress": "",  
}
```

```
"globalDataTimeWindow":
}
```

Response example

```
{
  "uuid":"8e57094d-78fa-4a25-a839-1b5e25763971",
  "appVersion":"HDCA Server v8.0-16070112",
  "dbVersion":"2.1",
  "apiVersion":{
    "dbApiVersion":"2.0",
    "alertApiVersion":"1.2"
  },
  "locale":"en-us",
  "timeZone":"Etc/UTC",
  "hostName":"compunws58.corp.devops",
  "ipAddress":"192.168.1.117",
  "globalDataTimeWindow":null
}
```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
C20000E4	Mandatory parameter {dataset} not specified.	Parameter <code>dataset</code> is not specified.
C20000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
C20000E9	Authorization is required.	Request is sent with an invalid authentication token.
C20600E2	Application encountered an internal error.	API server encountered an unknown error.

Chapter 8: Database API resources

This section describes the database and API properties, configuration, and existing schema of the Analyzer detail view server.

Configuring database and API properties

You can configure the database and API properties in the Analyzer detail view server. The properties of the database and API are included in the different property files.

You can find these files at the following locations:

- Database properties: `/usr/local/megha/conf/sys/app.db.properties`
- API properties: `/usr/local/megha/conf/sys/app.db.api.properties`

If these files do not exist, then you must create them. Any changes in the configuration properties take effect only after you restart the Analyzer detail view server.

The megha user must have read and write permissions for these two files.

Editing database properties

You can change the database properties by editing the `app.db.properties` file.

The database properties file is located at the following location: `/usr/local/megha/conf/sys/app.db.properties`. You can edit the following properties:

- The number of query processing threads: The API server divides a query into subtasks that can be run concurrently using a thread pool. By default, the query thread pool is configured to have two active threads.

Change the number of query processing threads by defining the property

`QueryProcess.threadpool.size`.

Recommendation: Set the number of query processing threads equal to the number of CPUs on the host..

- Memory bound for a resource relations cache: The resource relations cache maintains the relations mapping between the resources in memory, for fast access. By default, there is no memory bound on the resource relations cache, but you can put a memory bound by defining the following property:

`app.db.memory.bound.for.relation.caches.in.mb`

The bound is specified in MBps. If this bound is exceeded, then the cache overflows to disk.

If property value is set as 0, it is unbounded.



Note: This property should be changed only if the server is running out of memory. Putting a memory bound can result in a performance problem. If the size of a resource relations cache goes beyond the configured value, then performance is degraded as disk I/O increases.

The following table shows the sample case study (when you do not set memory bound and when you set memory bound) performed on datasets of various sizes.

Dataset number	Resource count	Resource relations cache size (MB)	Optimum: 100%		90%		75%		50%	
			Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)
1	65815	64	64	1.64	57.6	2.24	48	3.46	32	4.16
2	115404	110	110	1.50	99	2.90	82.5	2.18	55	3.42
3	338818	380	380	4.24	342	5.21	285	7.84	190	10.67
4	500000	625	625	112	562.5	95.00	468.75	126.00	312.5	166.00
5	740000	930	930	115	837	127.30	697.5	166.00	465	208.40

Dataset number	Resource count	Resource relations cache size (MB)	Optimum: 100%		90%		75%		50%	
			Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)	Mem. bound (MB)	Time (sec)
6	100000	1350	1350	164	1215	176.00	1012.5	271.00	675	344.70

The following describes the terms used in the sample case study (when you do not set memory bound and when you set memory bound) performed on datasets of various sizes:

- **Resource count:** Number of resources already in the Analyzer detail view database.
- **Cache size:** Memory used by the resource relations cache for resource count mentioned in the Resource Count column.
- **Optimum - 100%:** The memory usage and timings when there is no overflow to disk, and the whole relations cache is in-memory.
- **90%:** 90% of cache is in-memory, 10% is on disk.
- **75%:** 75% of cache is in-memory, 25% is on disk.
- **50%:** 50% of cache is in-memory, 50% is on disk.

The test operations are based on the following use case:

- **For datasets 1 to 3:** Update ~5,000 resources
- **For datasets 4 to 6:** Update 20,000 resources



Note: When configuring the bound for your dataset, keep a margin of 15 to 20% above the values specified in the table. For example, for 500,000 resources, the appropriate memory bound to keep the entire cache in-memory is 625 MB, but the recommended value for your dataset is 700 MB.

Summary of tuning properties

All tuning properties are summarized in the following table.

Parameter	Property name	Default value	Minimum value	Maximum value
Number of query processing threads	QueryProcess.threadpool.size	2	1	Should be the number of CPUs on host

Parameter	Property name	Default value	Minimum value	Maximum value
Memory bound for Resource Relations cache	app.db.memory.bound.for.relation.caches.in.mb	0	0	Java Integer max value, i.e., 2,147,483,647

Editing API properties

You can change the API properties by editing the `app.db.api.properties` file.

The API properties file is located at the following location: `/usr/local/megha/conf/sys/app.db.api.properties`.

The `createOrUpdateResourceData` API requests are categorized as small or big based on the length of the request body, and the rest are considered as small requests. A request that exceeds the limit defined in the property file

`app.db.api.small.request.max.content.length` is considered a big request. The API server saves all small requests in memory and saves big requests into a file before processing. The API server maintains two thread pools to process the big and small requests simultaneously.

The `createOrUpdateResourceData` API requests can contain multiple resources to be updated. For quick response time, the API server splits `createOrUpdateResourceData` requests into multiple batches of resources, and these batches are processed concurrently.

- **Maximum request length:** The API server rejects requests that are greater than 536,870,912 bytes (512 MB).

Change the maximum request length by defining following two properties:

`app.db.api.request.max.content.length` and
`app.db.api.request.max.content.length.factor`

The maximum memory available for JVM is divided by `app.db.api.request.max.content.length.factor`. This value is compared with `app.db.api.request.max.content.length`, and the smallest value of these two values is considered. Any request length greater than the smallest value is rejected.

- **Small request maximum content length:** By default, all requests with lengths up to 15,728,640 bytes (15 MB) are considered a small request. Requests greater than this value are considered big requests.

Change the small request maximum content length by defining the following property:

`app.db.api.small.request.max.content.length`

A large value for this property might result in holding the larger requests in memory, which can cause the server to run out of memory..

A small value for this property might result in increased file I/O operations and can impact performance.

- **Small request file write threshold:** By default, if a small request exceeds 10,485,760 bytes (10 MB), the API server writes the request in a temporary file.

Change the small request file write threshold value by defining the following property:

`app.db.api.small.request.file.write.threshold`

- **Small request executor thread pool size:** By default, the small request executor thread pool size is set to 4. So, four small requests can be executed in parallel.

Change the small request executor thread pool size by defining the following property:

`app.db.api.small.request.executor.threadpool.size`

- **Request split batch size:** By default, the API server puts 100 resources in one batch.

Change the request split batch size by defining the following property:

`app.db.api.request.split.batch.size`

This parameter must be set depending on the number of cores present on the server.

- **Big request executor thread pool size:** By default, big request executor thread pool size is set to 4. Four big requests can be executed in parallel.

Change the big request executor thread pool size value by defining the following property:

`app.db.api.big.request.executor.threadpool.size`

This parameter must be set depending on the number of cores present on the server.

- **Request processor queue load factor:** The API server parses requests and puts them in the executor queue. By default, the executor queue size is three times the number of threads in the pool. For example, if the thread pool has 4 threads, a maximum of $3 \times 4 = 12$ requests can be in the executor queue.

This executor queue load factor can be changed by defining the following property:

`app.db.api.request.processor.q.load.factor`

A large value for this property might result in holding the larger requests in the executor queue (in-memory) and the server might run out of memory.

- **Scheduler queue size:** The API server controls the number of small requests, which remain in memory. By default, if the number of small requests in the queue is greater than 200, further requests are written to temporary files. While processing, the API server reads the requests back and processes them.

Change the scheduler queue size by defining the following property:

`app.db.api.sched.q.size`

If the value of this property is small, then more requests are written and read. This increases the I/O of the file and might impact the performance.

A large value for this property might result in holding the small requests in the executor queue (in-memory) and the server might run out of memory.

- **Response retention time:** The API server allows the request to be processed asynchronously and stores the corresponding response in an in-memory cache. By default, the response is retained in the cache for 3,600 seconds (1 hour).

Change the retention time by defining the following property:

`app.db.api.response.cache.time.limit.in.secs`

A large value for this parameter might result in holding responses in memory for a longer duration.

- **Wait time:** A time interval that API request processor waits before sending the `Job Accepted` response. The default value for this parameter is 10 seconds. Override the default value by sending the `x-waitTime` parameter in the HTTP request header. Only a non-negative `x-waitTime` is overridden. See [API request headers \(on page 12\)](#) for more details on the `x-waitTime` parameter.

Change the default server side value by defining the following property:

`app.db.api.default.wait.time.for.request.completion.in.secs`

- **Compactor threshold resource count:** The `deleteResourceData` API does not delete resources from the disk, it just marks them as deleted. When the number of deleted resources exceeds a threshold value, the compactor runs and removes all resources from the disk that are marked as deleted. By default, the threshold value is set to 10,000.

Change the threshold value by defining the following property:

`app.db.api.compactor.threshold.resource.count`

- **Wait time for run compactor:** If the operations are not executed on the API server for 300,000 milliseconds (5 minutes) and the threshold values have already been exceeded, then the API server runs the compactor.

Change the wait time to run the compactor by defining the following property:

`app.gtc.idle.time.to.run.blocking.process.in.ms`

- **Compactor thread pool size:** The `deleteResourceData` API does not delete resources from the disk, it marks them as deleted. When the number of deleted resources exceeds a threshold value, the API server runs the compactor and removes all resources from the disk that are marked as deleted. By default, four threads are being used by the compactor.

Change the compactor thread pool size for run compactor by defining the following

property: `app.db.api.compactor.threadpool.size`

Summary of API properties

Parameter	Property name	Default value	Minimum value	Maximum value
Small request maximum content length	<code>app.db.api.small.request.max.content.length</code>	1048576	1	Java Integer max value: 2,147,483,647
Request maximum content length	<code>app.db.api.request.max.content.length</code>	2147483648	1	Java long max value: $2^{63} - 1$
Request maximum content length factor	<code>app.db.api.request.max.content.length.factor</code>	4	1	Java Integer max value: 2,147,483,647
Request split batch size	<code>app.db.api.request.split.batch.size</code>	100	1	Java Integer max value: 2,147,483,647

Parameter	Property name	Default value	Minimum value	Maximum value
Processor queue load factor	app.db.api.request.processor.queue.load.factor	3	1	Java Integer max value: 2,147,483,647
Scheduler queue size	app.db.api.scheduler.q.size	200	1	Java Integer max value: 2,147,483,647
Small request maximum content length	app.db.api.small.request.max.content.length	15728640	1	Java Integer max value: 2,147,483,647
Small request file write threshold	app.db.api.small.request.file.write.threshold	10485760	1	Java Integer max value: 2,147,483,647
Small request executor thread pool size	app.db.api.small.request.executor.threadpool.size	4	1	Java Integer max value: 2,147,483,647
Big request executor thread pool size	app.db.api.big.request.executor.threadpool.size	4	1	Java Integer max value: 2,147,483,647
Response retention time	app.db.api.response.cache.time.limit.in.secs	3600	5	3600
Wait time	app.db.api.default.wait.time.for.request.completion.in.secs	10	0	Java Integer max value: 2,147,483,647 [2]
Compactor threshold resource count	app.db.api.compactor.threshold.resource.count	10000	Java Integer min value i.e. -2,147,483,648	Java Integer max value: 2,147,483,647
Wait time for run compactor	app.gtc.idle.time.to.run.blocking.process.in.ms	300000	Java Integer min value i.e. -2,147,483,648	Java Integer max value: 2,147,483,647
Compactor thread pool size	app.db.api.compactor.threadpool.size	4	Java Integer min value i.e. -2,147,483,648	Java Integer max value: 2,147,483,647

The API server does not perform the validation on the minimum and maximum values of the configuration parameters. In case of any violation, the API server the uses default value and logs the appropriate message.

Configuring the number of worker threads for the application web server (Jetty)

The Analyzer detail view REST API interface is powered by the Jetty web server and is responsible for accepting incoming HTTP requests, parsing them, and passing them to the Analyzer detail view CRUD interface. By default, the Jetty web server is configured to have 200 worker threads. This default number of worker threads is enough for a normal workload. To run a large number of requests in parallel, increase the number of Jetty worker threads by changing the property `jetty.threadPool.maxThreads` in the file `/usr/local/megha/jetty/start.d/server.ini`.

If you increase the default number of Jetty worker threads, ensure that sufficient cores are available on the Analyzer detail view server host machine.

Increasing the maximum number of open file handles

If you expect a large number of concurrent requests in the Analyzer detail view server, increase the maximum number of file handles that can be opened by changing the system configuration. Perform the following steps on the VM or individual machine on which the Analyzer detail view server is installed.

Procedure

1. Log in to the VM or individual machine on which the Analyzer detail view server is installed.
2. Open the `sysctl.conf` file using the command `vi /etc/sysctl.conf`.
3. Add the following line at the end of file and set its value to a large number. In the example, the value is set to 65535. Refer to your OS documentation for the maximum allowed value.

```
fs.file-max = 65535
```

4. Save the file and exit.
5. Open the `limits.conf` file using the command `vi /etc/security/limits.conf`.
6. Add the following lines to the `limits.conf` file:

```
* soft nfile 65535
* hard nfile 65535
```

7. Save the file and exit.
8. Run the `ulimit -a` command to check the max open file ulimit:

```
[root@localhost ~]# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority     (-e) 0
```

```

file size          (blocks, -f) unlimited
pending signals    (-i) 63833
max locked memory  (kbytes, -l) 32
max memory size    (kbytes, -m) unlimited
open files         (-n) 65535
pipe size          (512 bytes, -p) 8

```

9. Reboot the VM or individual machine on which the Analyzer detail view server is installed.

Viewing the existing schema using REST APIs

To view the existing schema in the Analyzer detail view server, use the following APIs.

API	URL	Result
Retrieve dataset	<p>HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do?action=retrieveDataset&dataset=defaultDs</p> <p>Example:</p> <p>HTTPS://192.168.20.35:8443/dbapi.do?action=retrieveDataset&dataset=defaultDs</p>	Returns the number of attribute definitions, resource definitions, total resource count, and list of data subset IDs.
Retrieve specific data subset	<p>HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do?action=retrieveDataSubset&dataset=defaultDs&dataSubsetId=<Data Subset Name></p> <p>Example:</p> <p>HTTPS://192.168.20.35:8443/dbapi.do?action=retrieveDataSubset&dataset=defaultDs&dataSubsetId=97542</p>	Retrieves all attribute definitions with their details, such as ID, attribute type, attribute name, and unit of measure.
Retrieve all attribute definitions	<p>HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do?action=retrieveAttributeDef&dataset=defaultDs</p>	Retrieves all attribute definitions with their details, such as ID, attribute type, attribute name, and unit of measure.

API	URL	Result
	Example: HTTPS:// 192.168.20.35:8443/ dbapi.do? action=retrieveAttributeDef& dataset=defaultDs	
Retrieve specific attribute definition	HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do? action=retrieveAttributeDef& dataset=defaultDs&attrIds=<Attribute definition name> Example: HTTPS:// 192.168.20.35:8443/ dbapi.do? action=retrieveAttributeDef& dataset=defaultDs&attrIds=p volSerialNumber	Retrieves information of the specified attribute definition.
Retrieve all resource definitions	HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do? action=retrieveResourceDef &dataset=defaultDs Example: HTTPS:// 192.168.20.35:8443/ dbapi.do? action=retrieveResourceDef &dataset=defaultDs	Retrieves all the resource definitions with details, such as type, name, attribute names, and relations.
Retrieve specific resource definition	HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do? action=retrieveResourceDef &dataset=defaultDs&types= <Resource Definition Name> Example:	Retrieves information of the specified resource definition.

API	URL	Result
	HTTPS:// 192.168.20.35:8443/ dbapi.do? action=retrieveResourceDef &dataset=defaultDs&types=r aidMPB	
Retrieve resource data corresponding to resType/ attrId/ attrValue/ resSignatures	HTTPS://<Analyzer detail view server IP Address>:8443/dbapi.do? action=retrieveResourceDat a&dataset=defaultDs&resTy pe=<Resource Type> Example: HTTPS:// 192.168.20.35:8443/ dbapi.do? action=retrieveResourceDat a&dataset=defaultDs&resTy pe=raidLdev	Retrieves all the resource data corresponding to resource type or attribute ID, attribute values, or resource signature.

Chapter 9: Query API resources

You can run the Analyzer detail view query language (MQL) to get the resources data from the database.

Running the MQL query

The MQL query can get both time series and scalar data. The query time window must be specified in the Analyzer detail view server time zone.

Request line

```
POST baseUrl/dbapi.do?action=query&dataset=<Datasetname>&processSync=<true | false>
```

Request body

```
{
  "query": "",
  "startTime": "",
  "endTime": ""
}
```

Request example

```
POST /dbapi.do?action=query&dataset=defaultDs HTTP /1.1 Authorization:Basic
ZnJlZDpncmVh
```

```
{
  "query": " vm[~memory]",
  "startTime": "20140713_230100",
  "endTime": "20140714_000000"
}
```

Request parameters

Parameter	Type	Description
action	String	MQL query. Specify the API function to be invoked.
dataset	String	Dataset name. From 2 to 32 alphanumeric, underscore (_), and hyphen (-) characters are allowed.
query	String	Specify the MQL query.
processSync	Boolean	(Optional) Specify if the query should return incremental or complete results. False: Query returns the incremental result. True: Query returns the complete result. The default value is false.
startTime	String	Start time of the query. Date format: yyyyMMdd_HH:mm:ss. Time zone: Analyzer detail view server time zone.
endTime	String	End time of the query. Date format: yyyyMMdd_HH:mm:ss. Time zone: Analyzer detail view server time zone.
maxResults	Integer	(Optional) Specifies the maximum number of top-level resources on which the query can return in the response. This parameter is ignored if the <code>processSync</code> parameter is set to true. Positive integers. If not specified then the default value is the value of the <code>pending.result.limit.to.proceed.external</code> property from the <code>/usr/local/megha/conf/sys/query.properties</code> file.

Parameter	Type	Description
splitResponse	Boolean	<p>(Optional) If the value of this parameter is set to true, then the server splits the query response based on the maxResponseSize specified by the user or the default maxResponseSize.</p> <p>True or false</p> <p>Default: false</p>
useRealTime	Boolean	<p>(Optional) Helps to define whether real-time data should be included for performance reporting or analysis. If the value of this parameter is set to true, real-time data is included while reporting.</p> <p>True or false</p> <p>Default: false</p>
utcOffset	String	<p>(Optional) Offset from the coordinated universal time. The UTC offset is applicable for API request and response.</p> <p>UTC+/-HH:MM</p> <p>Example: UTC+09:00</p> <p>Default: Non-DST offset of Analyzer detail view server time zone is used.</p> <p>Note: When a time zone is within daylight saving time, the changes in offset according to daylight saving time are not applicable for this API. Only the Analyzer detail view server time zone offset is applicable.</p>

Response body

```
{
  "status":{
    "total":,
    "matched":,
    "processed":,
    "completed":,
    "aborted":,
    "hasErrors":
  },
}
```

```

"result":[
  {
    "signature":"",
    "name":"",
    "memory":{
      "type":"",
      "name":"",
      "unit":"",
      "data":""
    },
    "cpu Usage":[
      {
        "type":"",
        "name":"",
        "unit":"",
        "interval":,
        "start":"",
        "data":[
          , , , ,
        ]
      }
    ]
  }
]
}

```

Response example

```

{
  "status":{
    "total":100,
    "matched":1,
    "processed":10,
    "completed":false,
    "aborted":false,
    "hasErrors":false
  },
  "result":[
    {
      "signature":"vm#devops1-lab-vm2",
      "name":"devops-lab-vm2",
      "memory":{
        "type":"scalar",
        "name":"Memory",
        "unit":"",
        "data":"204800"
      },
      "cpu Usage":[
        {
          "type":"timeseries",
          "name":"cpu usage",

```

```

    "unit": "",
    "interval": 60,
    "start": "20140713_230100",
    "data": [
      24.00, 33.00, 68.00, null, 4.00
    ]
  }
]
}
]
}

```

Request parameters

Parameter	Type	Description
status	Object	Object that holds meta information about the progress of the query operation.
status.total	Number	Total number of top-level resources that exist in the database for the given time range.
status.matched	Number	Total number of top-level resources matched after applying all filters.
status.processed	Number	Total number of top-level resources processed.
status.completed	Boolean	Indicates that the query processing is finished.
status.aborted	Boolean	Indicates that the query processing is stopped by the user.
status.hasErrors	Boolean	Indicates that the query processing has encountered a runtime error. Results are processed until error conditions are returned.
result	Object[]	List of results.
result.signature	String	Unique identifier of the resource data.

Parameter	Type	Description
result.name	String	Value of the scalar attribute <i>name</i> , or in its absence the value of the signature.
result.<scalar attribute>	Object	If the scalar filter is present.
result.<scalar attribute>.type	String	Scalar (fixed value of the scalar data object).
result.<scalar attribute>.name	String	Display the name of the scalar attribute.
result.<scalar attribute>.unit	String	Unit of the scalar attribute.
result.<scalar attribute>.data	String	String value of the scalar attribute.
result.<timeseries attribute>	Object[]	If the timeseries filter is present.
result.<timeseries attribute>.type	String	The fixed value of the timeseries data object.
result.<timeseries attribute>.name	String	Name of the timeseries attribute.
result.<timeseries attribute>.unit	String	Unit of the timeseries attribute.
result.<timeseries attribute>.interval	Number	Integer interval of the data, in seconds.
result.<timeseries attribute>.start	String	Begin time of the data in the API server time zone, in the yyyyMMdd_HHmmss format.
result.<timeseries attribute>.data	Double[]	Timeseries (fixed value for timeseries data object).
result.related	String[]	Shows if the current path has the subpath.
result.scalarChanges	Object	Shows if the scalar data change filter is present.
result.scalarChanges.attrId	String	ID of the scalar attribute.
result.scalarChanges.data.name	String	Name of the scalar attribute.
result.scalarChanges.data.oldValue	String	Previous value of the scalar attribute.

Parameter	Type	Description
result.scalarChanges.data.newValue	String	New value of the scalar attribute .
result.scalarChanges.data.time	String	Time when the change is observed in the Analyzer detail view server time zone, in yyyyMMdd_HH:mm:ss format.
result.relationChanges	Object	Shows if the relation change filter is present.
result.relationChanges.signature	String	Signature of the resource data.
result.relationChanges.name	String	Name of the resource data.
result.relationChanges.connected	Boolean	Indicates if the resource is connected or not.
result.relationChanges.time	String	Time when the change in the relation status is observed, in the API server time zone in yyyyMMdd_HH:mm:ss format.
code	String	Eight-digit hexadecimal error code.
error	String	Error message.

Request examples

Query to get timeseries and scalar data:

```
{
  "query": "vm[=memory rx .*][@cpuUsage rx b .*]",
  "startTime": "20140713_230000",
  "endTime": "20140713_230500"
}
```

Query to get Tuning Manager historical timeseries (performance) data:

To query the migrated historical timeseries (performance) data, you must specify the following parameters in the request body:

- **db=cdb1:** Obtains only the Tuning Manager historical timeseries (performance) data.
- **inputInterval=supported-values:** The migrated historical timeseries (performance) data collection intervals. The supported values for `inputInterval` are: h (hour), d (day), w (week), m (month), and y (year). The query output is based on the specified `inputInterval` (data collection interval).

```
{
  "query": "(db=cdb1)raidLdev[=name rx 421358-00:00:09]&[@readIOPS rx b .*]
  {inputInterval=h}",
  "startTime": "20190404_113000",
  "endTime": "20190405_113000"
}
```

Notes:

- You can query Tuning Manager historical timeseries (performance) data that was processed until v11.0.0-00 of the Analyzer detail view server.
- You cannot set the advanced MQL aggregations, roll up and script filters, and synthetic attributes on the Tuning Manager historical timeseries (performance) data.
- Only `. *` is allowed as regex for timeseries attributes (double or string regex is not allowed).

Query to get changes in the scalar data:

This query gets the resources for which the scalar data of the specified attribute has been changed within the query time window.

```
{
  "query": " vm[~memory]",
  "startTime": "20140713_230100",
  "endTime": "20140714_000000"
}
```

Query to get resource relation changes:

This query gets the resources for which relation of the specified resource type has been changed within the query time window.

```
{
  "query": " vm[+h]",
  "startTime": "20140713_230100",
  "endTime": "20140714_000000"
}
```

Get the complete query result at the same time with `utcOffset` specified:

```
{
  "query": "vm[=memory rx .*][@cpuUsage rx b .*]",
  "startTime": "20140713_230000",
  "endTime": "20140713_230500",
  "utcOffset": "UTC-04:00"
}
```

When `splitResponse` and `maxResponseSize` are specified:

```
{
  "query": "vm[=memory rx .*][@cpuUsage rx b .*]",
  "startTime": "20140713_230000",
  "endTime": "20140713_230500",
  "splitResponse": true,
  "maxResponseSize": 0.05
}
```

If `useRealTime` is not specified in the API URL but specified in the query API body, the value defined in the query API request body is considered.

```
{
  "query": "(useRealTime=true)raidPort[@totalIOPS rx b .*]",
  "startTime": "20170124_000000",
  "endTime": "20170124_150000"
}
```

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.

Status code	Message	Description
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
62010008	Mandatory parameter {query} is not specified.	Parameter <code>query</code> is not specified.
6201000F	Invalid value specified for parameter {query}. Query : vm[memory rx .*].	Parameter <code>query</code> contains an invalid MQL query.
62010008	Mandatory parameter {startTime} is not specified.	Parameter <code>startTime</code> is not specified.
6201000F	Invalid value specified for parameter {startTime}.	Parameter <code>startTime</code> is not specified in the correct format (yyyyMMdd_HH:mm:ss).
6202000F	Invalid value specified for parameter {startTime}.	Specified <code>startTime</code> parameter is not in the valid range.
62010008	Mandatory parameter {endTime} is not specified.	Parameter <code>endTime</code> is not specified.

Response code	Message	Condition
6201000F	Invalid value specified for parameter {endTime}.	Parameter <code>endTime</code> is not specified in proper format (yyyyMMdd_HH:mm:ss).
6202000F	Invalid value specified for parameter {endTime}.	Specified <code>endTime</code> is not in the valid range.
6202000F	Invalid value specified for parameter {startTime, endTime} the endTime must be after startTime.	Specified <code>endTime</code> parameter is before or equal to the <code>startTime</code> parameter.
62010002	Invalid JSON Request.	Specified input JSON is invalid.
6202001B	Input query length is beyond the permissible limit of 8192 characters.	Input query length is beyond the permissible limit.
620000E9	License is not valid.	License is not uploaded, or the uploaded license is invalid or expired.
620000E9	Authorization needed.	Request is sent with an invalid authentication token.
620000E4	Dataset with {name}:[ds1] does not exist.	Request is sent for a dataset that does not exist.
620000E0	HTTP method [GET] is not supported for action [query].	Request is sent with any HTTP method other than POST.
620000E2	Application encountered an internal error.	API server encountered an unknown error.

Best practices for query APIs

Consider the following best practices when using the query API.

Avoid inordinate resource usage on the Analyzer detail view server

Compose your queries by considering the system performance monitor resource usage.

Aborting requests to collect results

If HTTP status code 206 is returned from the query API, use the get request status API to obtain the results until status code 200 is returned. To stop an API request while it is still processing, use the abort request API. If you do not use the abort request API, then the query result is saved in memory until the response is obtained (default: one hour), after the query processing is complete.

Split resource-intensive queries

We do not recommend running a query that might involve a large number of resources. If you must run such a query, use the REST API to check the CPU and memory usage. For best results, divide the request into multiple queries.

For example, the following query involves a large number of resources at a time:

```
raidMPB[=name rx XXXXXX.*/raidLdev[=volumeAttribute rx V-VOL]&[=name rx .*/
raidPool[=name rx .*/raidLdev[=volumeAttribute rx POOL]&[=name rx .*/raidPG[=name
rx .*]
```

The best practice is to split the query as shown in the following example.

Query1:

```
raidMPB[=name rx XXXXXX.*/raidLdev[=volumeAttribute rx V-VOL]&[=name rx .*/
raidPool[=name rx .*]
```

Query 2:

```
raidPool[=name rx Pool-names-retrieved-by-query-1]/raidLdev[=volumeAttribute rx
POOL]&[=name rx .*/raidPG[=name rx .*]
```

Avoid too many filtering conditions

Using a large number of filter conditions can significantly increase the system load. When specifying filter conditions in MQL, use the REST API to monitor the resource usage, and reduce the number of filter conditions accordingly.

Separate the query conditions

For example, the following query involves too many conditions at a time:

```
raidLdev[=name rx
XXXXXX-00:00:00|XXXXXX-00:00:01|Too-many-conditions|XXXXXX-00:03:FF]&[@totalIOPS
rx b .*][@responseTime rx b .*]
```

The best practice is to split the conditions as shown in the following example.

Query 1:

```
raidLdev[=name rx
XXXXXX-00:00:00|XXXXXX-00:00:01|Conditions|XXXXXX-00:01:FF]&[@totalIOPS rx b
.*][@responseTime rx b .*]
```

Query 2:

```
raidLdev[=name rx
XXXXXX-00:02:00|XXXXXX-00:02:01|Conditions|XXXXXX-00:03:FF]&[@totalIOPS rx b
.*][@respo
```

Use a regular expression

For example, the following query involves too many conditions at a time:

```
raidLdev[=name rx
XXXXXX-00:00:00|XXXXXX-00:00:01|Too-many-conditions|XXXXXX-00:03:FF]&[@totalIOPS
rx b .*][@responseTime rx b .*]
```

The best practice is to use the regular expression as follows:

```
raidLdev[=name rx
XXXXXX-00:0[0-3]:.*]&[@totalIOPS rx b .*][@responseTime rx b
.*]
```

Chapter 10: Alert API resources

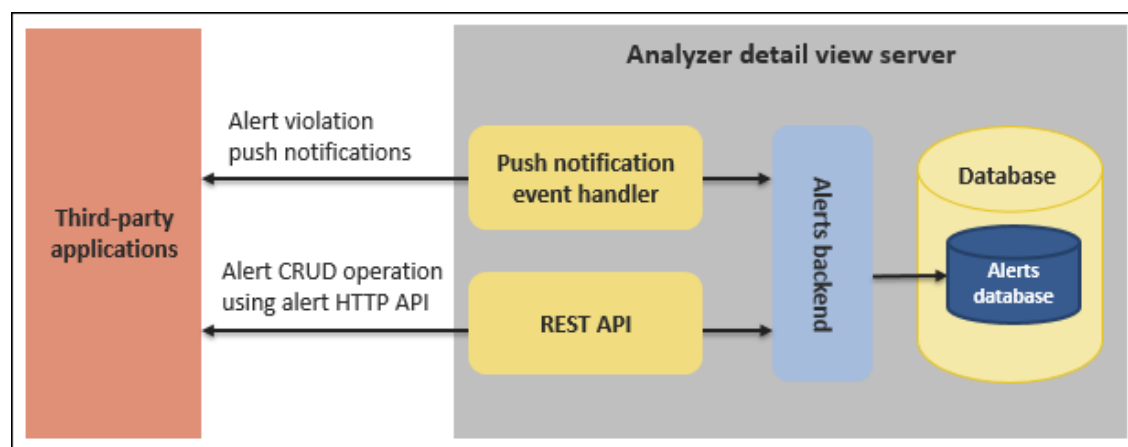
Using the alert APIs, you can create and obtain the alert definitions, you can set up the notification profiles to receive alert notifications.

The following are the features of the alert APIs:

- Allows you to create, obtain, update, delete, subscribe, unsubscribe, pause, and restart the paused alerts in the Analyzer detail view over HTTP.
- Provides the API access for authorized users only. The authorization is based on Analyzer detail view users.
- Allows you to obtain or acknowledge violations.
- Allows the alert definition owner to add or remove resources, and update thresholds of existing resources.
- Alerts are shared between all the users. However, only owners (owner is a user who creates an alert) and built-in admin user can perform administrative tasks on them.
- The Analyzer detail view can send push notifications for alerts. You can subscribe for push notifications using the alert API.

The following diagram illustrates the alert API flow.

You can access all alert definitions (created using UI or API) that are available in the alert database.



Alert API design

The alert APIs follow the same design concepts that are used in the database API.

Following are the high-level design concepts:

- The request URL of the database API and alert API have the common endpoint, however, these URLs are identified by the functions. For example, the general URL format is `https://localhost:8443/alertapi.do?action=XXX` where XXX is the API function.
- API functionality is supported only through two HTTP methods: GET and POST. The DELETE, PUT, and other HTTP methods are not considered.
- The API platform is reused. The `getStatus` and `getAuditLogs` APIs are similar to that of Database APIs.

Supported alert API functions

Function	Description
<code>createAlertDef</code>	Create the alert definition.
<code>retrieveAlertDef</code>	Retrieve information about the alert definition.
<code>updateAlertDef</code>	Add or remove resources and threshold.
<code>getAlerts</code>	Get alert violations.
<code>subscribeAlertDef</code>	Subscribe to alert definition.
<code>unsubscribeAlertDef</code>	Unsubscribe the subscribed alert definition.
<code>disableAlertDef</code>	Pause alert evaluation.
<code>enableAlertDef</code>	Restart alert evaluation.
<code>getAuditLogs</code>	Get audit logs for API calls.
<code>getStatus</code>	Verify the request that is processing the status of the jobs those are being run asynchronously. This is a helper function available for all the above functions.

Access rights for alert API functions

Function	User
<code>createAlertDef</code>	All
<code>retrieveAlertDef</code>	All
<code>updateAlertDef</code>	Creator and Admin role user
<code>getAlerts</code>	All

Function	User
subscribeAlertDef	All
unsubscribeAlertDef	All
disableAlertDef	Creator and Admin role user
enableAlertDef	Creator and Admin role user
deleteAlertDef	Creator and Admin role user
getAuditLogs	All

Creating an alert definition

Analyzer detail view classifies metrics into two types: configuration metrics and performance metrics. Configuration metrics do not change frequently such as resources Status, Capacity, and so on. Performance metrics change frequently such as IOPS, response time, and so on. You can create an alert definition for critical parameters of the configuration and performance metrics.

Refer to, *Hitachi Ops Center Analyzer Detail View Metrics Reference Guide* for a list of the configuration metrics for which you can create an alert definition.

You can create multiple alert definitions through a single request.



Note: After creating a new alert definition, it is attached to the associated resource when the data is imported. However, the actual evaluation of the alert definition does not begin until the next data is imported.

If the owner of an alert definition is deleted, then the alert definition ownership is transferred to the built-in administrator user. (The user name for the built-in administrator account is `admin`.)

Request line

```
POST baseUrl/alertapi.do?action=createAlertDef
```

Request payload is an array of alert definition JSON objects.

Request body

```
[
  {
    "name": "",
    "metric": "",
    "resource_mql": "",
    "resource_type": "",
    "resources": [
```

```

{
  "signature": "",
  "threshold": [

  ]
},
{
  "signature": ""
}
],
"severity": "",
"condition": "",
"default_threshold": [

],
"criteria": {
  "type": "",
  "m": ,
  "n": ,
  "n_unit": ""
},
"monitoring_window": {
  "from": "",
  "to": ""
},
"subscription": {
  "email": {
    "id": [
      "",
      ""
    ],
    "damping": ""
  },
  "syslog": {
    "server": [
      "",
      ""
    ]
  },
  "snmp": {
    "manager": [
      "hostname",
      "IP"
    ]
  }
}
}
]

```

Request example

```
POST /alertapi.do?action=createAlertDef
```

```
[
  {
    "name": "Production VMs High CPU Usage",
    "metric": "cpuUsage",
    "resource_mql": "ldev[=type rx virtual]",
    "resource_type": "ldev",
    "resources": [
      {
        "signature": "XXXXXX",
        "threshold": [
          90
        ]
      },
      {
        "signature": "XXXXXX"
      }
    ],
    "severity": "critical",
    "condition": "gt",
    "default_threshold": [
      80
    ],
    "criteria": {
      "type": "m/n",
      "m": 3,
      "n": 5,
      "n_unit": "minutes"
    },
    "monitoring_window": {
      "from": "08:00",
      "to": "17:59"
    },
    "subscription": {
      "email": {
        "id": [
          "xxx",
          "yyy"
        ],
        "damping": "no_damping"
      },
      "syslog": {
        "server": [
          "hostname:portNo:TCP",
          "IP:portNo:UDP"
        ]
      }
    }
  },
]
```



```

    "snmp": {
      "manager": [
        "hostname",
        "IP"
      ]
    }
  }
}
]

```

Request example of the configuration metrics with string value (having condition)

```
POST /alertapi.do?action=createAlertDef
```

```

[
  {
    "name": "test_string_10",
    "severity": "critical",
    "resource_type": "raidPool",
    "metric": "dataSaving",
    "condition": "having",
    "default_threshold": [
      "Compression"
    ],
    "resources": [
      {
        "signature": "raidPool#421358-14"
      }
    ]
  }
]

```

Response JSON:

```

{
  "created": [
    {
      "name": "test_string_10",
      "id": 57
    }
  ]
}

```

Request parameters



Note: In the following table, if a parameter is specified in the `a:b` format, then it means field `b` of JSON object `a`.

Parameter	Type	Description
name	String	<p>Name of the alert. Alphanumeric, underscore (<code>_</code>), space characters are allowed</p> <p>Name must be unique across all alert definitions with maximum length of 100 characters.</p>
metric	String	<p>Attribute ID on which the alert is created.</p> <p>Valid attribute ID, must be present in <code>AttributeDef</code> and associated with resources for which the alert is to be created.</p> <p>Dynamic attribute is not supported.</p>
condition	String	<p>Alert condition: <code>gt</code> for greater than, <code>lt</code> for less than, and <code>bt</code> for between.</p> <p>Alert condition for string metric: If the metric type is string, then you must provide the condition as <code>having</code>.</p>
default_threshold	Float[]	<p>Threshold for which condition is to be evaluated.</p> <p>This value can be overridden by the value in the resource object.</p> <p>For the <code>bt</code> condition, two values must be specified in the array. The first value in the array is lower bound and the second one is upper bound. Both these values are inclusive in violation computation.</p>

Parameter	Type	Description
		Float: <ul style="list-style-type: none"> Maximum value = Java Float.Max value i.e., +3.4028235 E 38. Minimum value = Java Float.Min value i.e., - 3.4028235 E 38.
default_threshold (for having condition)	String	The threshold field for having condition supports alphanumeric, hyphen (-), underscore (_), comma (,), colon (:), at (@), slash (/), dot (.), and space.
criteria	Object	Criteria to determine the alert condition violation. Note: The <code>criteria</code> parameter is not required while setting the alert definition for configuration metric.
criteria:type	String	Type of criteria. Default value is <code>m/n</code> . Only supported option is <code>m/n</code> <code>m</code> = number of samples <code>n</code> = duration
criteria:m	Integer	Number of samples (threshold violations). Value of <code>M</code> in <code>m/n</code> criteria. $0 < m \leq \text{samples in } N \text{ time window}$.
criteria:n	Integer	Length of a time-window. Value of <code>N</code> in <code>m/n</code> criteria.
criteria:n_unit	String	(Optional) Unit of time-window <code>N</code> . $0 < n \leq 60 \text{ minutes}$

Parameter	Type	Description
resource_mql	String	(Optional) MQL for resource selection. It is useful for setting alerts on a tree node for all nodes of a certain type. You do not have to maintain a list of resources because the Analyzer detail view transparently manages the additions and deletions. The resource MQL is run every time a configuration import occurs, and alert definitions are attached to the newly added resources matching MQL only on configuration import.
resource_type	String	(Optional) A valid resource type value . This is useful if you want to apply an alert on all resources.
resources	Object[]	(Optional) Array of resource objects on which alert is to be defined. All resources should be of the following type: <code>resource_type</code> .
resources:signature	String	Signature of a resource.
resources:threshold	Float[]	(Optional) Resource specific threshold of the alert. This overrides the <code>default_threshold</code> . Float: <ul style="list-style-type: none"> ▪ Max value = Java Float.Max value i.e., +3.4028235 E 38. ▪ Min value = Java Float.Min value i.e., -3.4028235 E 38.

Parameter	Type	Description
<code>resources:threshold</code> (for having condition)	String	<p>(Optional) Resource specific threshold of the alert. This overrides the <code>default_threshold</code>.</p> <p>The threshold field for having condition supports alphanumeric, hyphen (-), underscore (_), comma (,), colon (:), at (@), slash (/), dot (.), and space.</p>
<code>severity</code>	String	<p>(Optional) Severity of the alert: warning or critical</p> <p>Default: critical</p>
<code>monitoring_window</code>	Object	<p>(Optional) Restrict alert evaluation to certain hours of the day. This can be specified in half hour increments. Both <code>from</code> and <code>to</code> values are inclusive.</p> <p>Alert monitoring window is based on UTC timezone. Be sure to select the appropriate time window. If applicable, also consider the Daylight Saving Time. For example, if the Analyzer detail view UI timezone is set to Pacific Time (GMT-08:00) and you want to add alert for time window 9:00 AM to 7:00 PM, then set alert</p> <pre>monitoring_window:from and monitoring_window:to as 05:00 PM to 03:00 AM, respectively.</pre> <p>Note: The <code>monitoring_window</code> parameter is not required while setting the alert definition for configuration metric.</p>

Parameter	Type	Description
monitoring_window:from	String	(Optional) Start time of the monitoring window. The time specification must be in HH:MM format. Valid values for HH range from 00 to 23, and valid values for MM are 00 and 30.
monitoring_window:to	String	(Optional) End time of the monitoring window. The time specification must be in HH:MM format. Valid values for HH range from 00 to 23, and valid values for MM are 29 and 59.
subscription	Object[]	(Optional) Details of the alert subscription, that describes how you want to subscribe the alert notification. For example, email, snmp, or syslog.
subscription:email:id	String[]	(Optional) Array of email IDs.
subscription:email:damping	String	(Optional) Specify if you want to send individual violation or in a group. (no_damping, state_change, hourly_gist, daily_gist)
subscription:snmp:manager	String	IP addresses or host names of SNMP Managers. <host>: Either host name or IP address of the SNMP Manager.
syslog:server	String[]	(Optional) Array of IP address or host name of the Syslog server.

Parameter	Type	Description
		<p>Array of strings, each representing a server in “<host>:<port number>:<protocol>” format.</p> <ul style="list-style-type: none"> ▪ <host>: Either hostname or IP address of the syslog server. ▪ <port number>: Valid port number. ▪ <protocol>: TCP or UDP.

Response body

On successful alert creation, ID is generated and is returned in response. This ID acts as a primary key for further API calls.

```
{
  "created": [
    {
      "id": 1,
      "name": "Sample AlertDef One"
    }
  ]
}
```

Response example with an error

```
{
  "created": [
    {
      "id": 1,
      "name": "Sample AlertDef One"
    }
  ],
  "failed": [
    {
      "reason": "Attribute definition with id:[unknownAttribute] does not exist.",
      "name": "Sample AlertDef Two",
      "code": "A1020008"
    }
  ]
}
```

Status codes

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
A1020003	Mandatory parameter {name} is not specified.	Alert name is not specified.
A1020003	Mandatory parameter {metric} is not specified .	Metric is not specified.
A1020008	Attribute definition with id: [unknownAttribute] does not exist.	Specified metric does not exist.
A1020004	At least one of {resource_type, resource_mql} parameters must be specified.	Parameters <code>resource_type</code> and <code>resource_mql</code> are not specified.
A1020009	Resource definition with type: [unknownResourceType] does not exist.	Specified <code>resource_type</code> does not exist.
A102000A	Resource definition with type [vm] does not contain [diskMaxTotalLatency] metric.	Specified resource type does not contain the given metric.
A1020003	Mandatory parameter {condition} is not specified.	Condition is not specified.

Response code	Message	Condition
A1020003	Mandatory parameter {criteria} not specified.	Criteria is not specified.
A1020006	Invalid {criteria} is specified.	Criteria:m, Criteria:n are not specified.
A1010003	Mandatory parameter {default_threshold} is not specified.	Parameter default_threshold is not specified
A1020006	Invalid {condition} is specified.	Invalid condition is specified.
A1020006	Invalid {resource_mql} is specified.	Invalid resource_mql is specified.
A1010007	Invalid {threshold} specified for resource [vm#vm1].	Invalid threshold is specified for any resource.
A104000C	Failed to add resource [vm#vm10] as resource does not exist.	Resource does not exist for the specified resource signature.
A1010006	Invalid {monitoring_window} is specified.	Invalid value is specified for monitoring window.
A1010005	Invalid {emailIds} specified [yyy, abc@@gmail.com].	Invalid value is specified for email IDs.
A1010006	Invalid {damping} specified.	Invalid value is specified for email damping.
A1010005	Invalid {sysLogServer} specified [hostName].	Invalid Syslog server details.
A1020018	Resource [ldev#sig2] is not present.	Resource on which you want to apply the alert definition and alert definition does not exist.
A1020019	Found unknown properties [resource, unknownKey].	Unknown key is specified.
A1020021	Invalid attribute specified [attribute_id].	Specified attribute is of the scalar type and cannot generate an alert on the specified attribute ID.
A1020022	Invalid condition specified [condition].	Invalid condition is specified.

Getting information about alert definitions

You can obtain information for alert definitions through a single request.

Request line

```
baseUrl/alertapi.do?action=retrieveAlertDef
```

Request body

```
{
  "defStatus":""
}
```

Request example using GET

```
GET /alertapi.do?action=retrieveAlertDef&defStatus=subscribed
```

Request example using POST

```
POST /alertapi.do?action=retrieveAlertDef
```

```
{
  "defStatus":"subscribed"
}
```

Request parameters



Note: If you do not specify the parameter value in the request, then all definitions present in the system are returned.

Parameters	Values	Description
defStatus	String	(Optional) Filter by definition status: created, subscribed, unsubscribed, and disabled.

Parameters	Values	Description
owner	String	<p>(Optional) Filter by definition owner (user who created the alert). A valid user on the Analyzer detail view.</p> <p>Note: Any user can read alerts created by other users including admin.</p> <p>For the Common Services user, use the following format:</p> <pre>user- name@CommonService</pre> <p>For example: sysadmin@CommonService</p>
severity	String	(Optional) Filter by alert severity: critical or warning.
from	String	(Optional) Retrieve definitions created in a certain time window. This is the start of the time window in yyyyMMdd_HH:mm:ss. By default, this timestamp is in the UTC time zone unless the <code>utcOffset</code> parameter is specified.
to	String	(Optional) Retrieve definitions created in a certain time window. This is end of the time window in yyyyMMdd_HH:mm:ss. By default, this timestamp is assumed to be in the UTC time zone unless the <code>utcOffset</code> parameter is specified.
utcOffset	String	(Optional) UTC offset for <code>from</code> and <code>to</code> timestamps. The valid UTC offset string example: UTC+05:30, UTC-08:00

Parameters	Values	Description
namePattern	String	(Optional) Filter by alert name. Regex matching the name of the definition.
resources	String[]	<p>(Optional)</p> <p>List of resource signatures.</p> <p>Retrieve alerts configured for resources provided in this list.</p> <p>Array of strings: Valid resource signatures for POST.</p> <p>This parameter cannot be specified using HTTP GET.</p>
alertIds	Integer[]	<p>(Optional) Retrieve for specific list of alert IDs. If this list is provided, all the other parameters are ignored.</p> <p>For HTTP GET, a comma-separated list of alert IDs.</p> <p>For HTTP POST, an array of alert IDs</p>
includeSubscription	Boolean	<p>(Optional) Include subscription settings in output.</p> <p>True or False</p> <p>Default: False</p>
includeResourceList	Boolean	<p>(Optional) Include resources in output.</p> <p>True or False</p> <p>Default: False</p>

Parameters	Values	Description
resInstanceAlertsOnly	Boolean	<p>(Optional) Retrieve definition of specific resource signatures.</p> <p>True or False</p> <p>If the value is true, then the retrieved list contains only those alert definitions whose signature matches the resources list.</p> <p>Default: False</p>



Note: The retrieve alert definition output displays `advanced_properties` used for creating alerts from the Analyzer detail view server UI. They contain information about the resource tree and are displayed when an alert is added through UI. (Otherwise, the values are displayed as `null`.)

Response body

```
{
  "retrieved": [
    {
      "alertId":,
      "name": "",
      "owner": "",
      "defStatus": "",
      "metric": "",
      "condition": "",
      "default_threshold": [
        ,
      ],
      "severity": "critical",
      "criteria": {
        "type": "",
        "m":,
        "n":,
        "n_unit": ""
      },
      "resource_type": "",
      "resources": [
        {
          "signature": "",
          "threshold": [
            ,
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
],
"monitoring_window":{
  "to":"",
  "from":""
},
"subscriptions":[
  {
    "subscription":{
      "syslog":{
        "server":[ ]
      },
      "email":{
        "id":[
          ""
        ],
        "damping":""
      }
    },
    "user":""
  }
],
"advanced_properties":{
  "category":null,
  "breadcrumb":null,
  "probe_type":null,
  "probe_ids":null,
  "sub_category":null,
  "category_resource_type":null,
  "category_tree_filter":null,
  "from_tree":false,
  "resource_filter":null,
  "parent_resource_key":null,
  "dynamic_resource_key":null,
  "resource_pattern":null,
  "resource_only":true
}
},
]
}

```

Response example

```

{
  "retrieved":[
    {
      "alertId":2,
      "name":"Sample AlertDef 2",
      "owner":"admin",
      "defStatus":"subscribed",

```

```

"metric": "cpuUsage",
"condition": "bt",
"default_threshold": [
  30.0, 60.0
],
"severity": "critical",
"criteria": {
  "type": "m/n",
  "m": 3,
  "n": 5,
  "n_unit": "minutes"
},
"resource_type": "vm",
"resources": [
  {
    "signature": "vm#vm2",
    "threshold": [
      20.0, 90.0
    ]
  }
],
"monitoring_window": {
  "to": "23:59",
  "from": "10:00"
},
"subscriptions": [
  {
    "subscription": {
      "syslog": {
        "server": [ ]
      },
      "email": {
        "id": [
          "rah.shrma@culus.com"
        ],
        "damping": "no_damping"
      }
    },
    "user": "admin"
  }
],
"advanced_properties": {
  "category": null,
  "breadcrumb": null,
  "probe_type": null,
  "probe_ids": null,
  "sub_category": null,
  "category_resource_type": null,
  "category_tree_filter": null,
  "from_tree": false,
  "resource_filter": null,

```

```

        "parent_resource_key":null,
        "dynamic_resource_key":null,
        "resource_pattern":null,
        "resource_only":true
    }
},
{
    "alertId":1,
    "name":"Sample AlertDef 1",
    "owner":"admin",
    "defStatus":"subscribed",
    "metric":"cpuUsage",
    "condition":"bt",
    "default_threshold":[
        10.0, 100.0
    ],
    "severity":"critical",
    "criteria":{
        "type":"m/n",
        "m":3,
        "n":5,
        "n_unit":"minutes"
    },
    "resource_type":"vm",
    "resources":[
        {
            "signature":"vm#vm1",
            "threshold":[
                10.0, 100.0
            ]
        },
        {
            "signature":"vm#vm2",
            "threshold":[
                10.0, 100.0
            ]
        }
    ],
    "monitoring_window":{
        "to":"23:59",
        "from":"10:00"
    },
    "subscriptions":[
        {
            "subscription":{
                "syslog":{
                    "server":[ ]
                },
                "email":{
                    "id":[
                        "mj.w@cumls.com"
                    ]
                }
            }
        }
    ]
}

```



```

    ]
    "damping": "no_damping"
  }
},
"user": "admin"
}
]
}
]
}

```



Note: The input parameters `includeSubscription` and `includeResourceList` control verbosity of the response.

Response example for the configuration metrics:

```

{
  "retrieved": [
    {
      "alertId": 57,
      "name": "test_string_10",
      "owner": "admin",
      "defStatus": "unsubscribed",
      "metric": "dataSaving",
      "condition": "having",
      "default_threshold": [
        "Compression"
      ],
      "severity": "critical",
      "resource_type": "raidPool",
      "isSystemLevelAlert": false
    }
  ]
}

```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.

Response codes

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
A2020006	Invalid {from} is specified.	Invalid <code>from</code> timestamp.
A2020006	Invalid {to} is specified.	Invalid <code>to</code> timestamp.
A2020003	Mandatory parameter {from} is not specified.	If <code>to</code> timestamp is specified but <code>from</code> is not specified.
A2020003	Mandatory parameter {to} is not specified.	If <code>from</code> timestamp is specified but <code>to</code> is not specified.
A2020006	Invalid {defStatus} is specified.	Invalid <code>defStatus</code> .
A2020006	Invalid {severity} is specified.	Invalid <code>severity</code> .
A2020006	Invalid {namePattern} is specified.	Invalid regex is specified in the <code>namePattern</code> .
A2020015	Time window is not specified for the <code>utcOffset</code> .	Time window is not specified for <code>utcOffset</code> .
A2020006	Invalid {utcOffset} is specified.	Invalid <code>utcOffset</code> .
A2000013	Application encountered internal error.	API server encountered an internal error.
A2020019	Found invalid properties [resource, unknownKey].	Invalid key is specified.

Getting information about alert condition violations

You can obtain information about alert condition violations in a single request.

When you create an alert definition, the time (in GMT format) is recorded on the Analyzer detail view and is attached to the matching resources. The alert definition is then attached to the matching resources. If you create an alert definition when there is a delay in the data import, then the violation computation occurs only after the import time reaches the alert creation time.

The following table shows the behavior of `getAlerts` API function in different scenarios.

Alert definition creation time	Data import time	Time window	Violation computation	getAlerts API output	Remark
8:00 AM	6:00:00 AM (delay in the data import)	6 AM – 7:59 AM	No	No	Alert definition does not exist in a given time window.
		8 AM – 10 AM	Yes	Yes	Returns violation from 8:00 AM to 10:00 AM.
		7 AM – 9 AM	Yes (8 AM - 9 AM)	Yes (8 AM - 9 AM)	Alert definition does not exist for the time window 7:00 AM to 8:00 AM. Returns violation from 8:00 AM to 9:00 AM.

This same behavior is applicable while updating the alert violation threshold.

Request line

```
baseUrl/alertapi.do?action=getAlerts&from=<from>&to=<to>
```

Request parameters

Parameter	Type	Description
from	String	Start of the specified time period in yyyyMMdd_HH:mm:ss. The default time period is in the UTC time zone, unless the <code>utcOffset</code> parameter is specified.

Parameter	Type	Description
to	String	End of the specified time period in yyyyMMdd_HH:mm:ss. The default time period is in the UTC time zone, unless the <code>utcOffset</code> parameter is specified.
utcOffset	String	(Optional) UTC offset for the from and to time period. The valid UTC offset string example UTC+05:30, UTC-08:00 If requested, the time period in the response message uses the UTC time zone format.
alertIds	Integer[]	(Optional) Gets violations for the specific alerts. For HTTP-GET, list of comma separated alertIds. For HTTP-POST, array of alertId strings. If not requested, the API returns alert violation data for alertIds present in the specified time window.
resources	String[]	(Optional) Gets violations for specified resources. For HTTP-POST, array of resource signature strings. This parameter cannot be specified using the HTTP-GET request.
severity	String	(Optional) Severity of alert definition: critical or warning.

Request examples

The following request returns all violations generated in a given time period:

```
GET /alertapi.do?action=getAlerts&from=20151201_203000&to=20151201_223000
```

```
POST /alertapi.do?action=getAlerts
{
  "from": "20151201_203000",
  "to": "20151201_223000"
}
```

The following request returns all violations for a particular ID generated in a given time period:

```
GET /alertapi.do?
action=getAlerts&from=20151201_203000&to=20151201_223000&alertIds=123231
```

```
POST /alertapi.do?action=getAlerts
{
  "from": "20151201_203000",
  "to": "20151201_223000"
  "alertIds": [123231]
}
```

The following request returns all violations for a particular severity generated in a given time period:

```
GET /alertapi.do?
action=getAlerts&from=20151201_203000&to=20151201_223000&severity=critical
```

```
POST /alertapi.do?action=getAlerts
{
  "from": "20151201_203000",
  "to": "20151201_223000",
  "severity": "critical"
}
```

Response body

```
[
  {
    "alertId": "",
    "name": "",
    "severity": "",
    "violations": {
      "XXX": [
        "",
```

```

        "",
        ""
    ],
    "YYY": [
        "",
        "",
        ""
    ]
}
}
]

```



Note: The violations object in the response message indicates when the violation occurred. The Key in this object is the resource signature, and the value is an array of time period in YYYYMMDD_HHMMSS format. The default time is in the UTC time zone. However, if the `utcOffset` parameter is specified as input then the time is shown is the same time zone and `utcOffset` key specifies the time zone in the output.

Response example

```

[
  {
    "alertId": "123213",
    "name": "Production VMs High CPU Usage",
    "severity": "critical",
    "violations": {
      "XXX": [
        "20151201_203000",
        "20151201_203300",
        "20151201_203400"
      ],
      "YYY": [
        "20151201_203200",
        "20151201_203300",
        "20151201_203400"
      ]
    }
  }
]

```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.

For a list of alert API response codes, see [Alert API response error codes and messages \(on page 269\)](#).

Subscribing to push notifications

You can subscribe to push notifications to receive alerts for the violations when they occur.

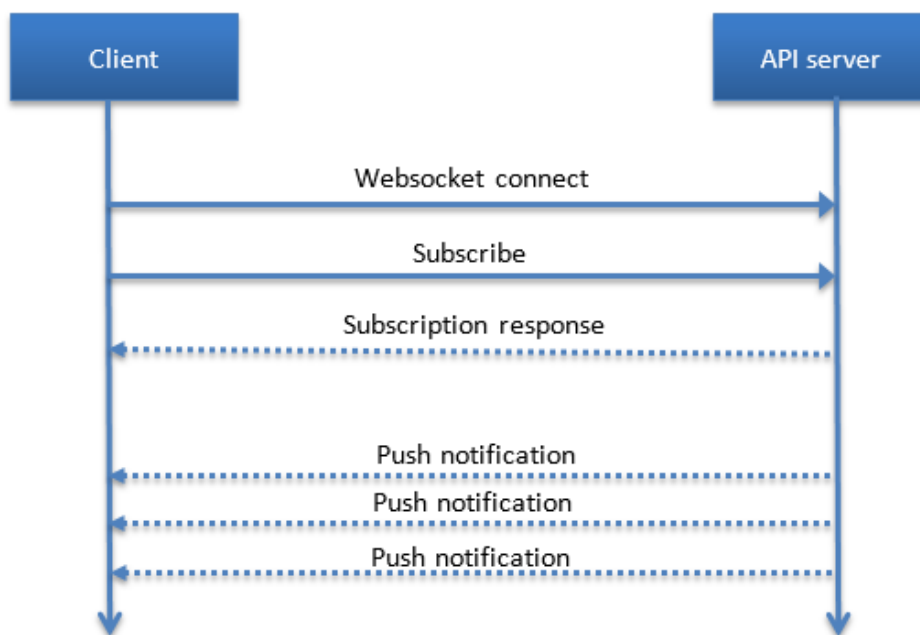
For push notifications, you must open the websocket connection between the client (the client used to invoke the REST API calls) and the Analyzer detail view, and send the list of alert IDs for which you want to receive notifications. The Analyzer detail view then starts sending the alerts for the violations over this connection until the connection is closed or terminated.

To subscribe to the push notifications, you must send the list of alert IDs to the Analyzer detail view (this is a one-time activity). This list of alert IDs cannot be changed later.

To change the subscription, you must stop the connection and reconnect using the updated list of alert IDs.

Push notifications are designed to send alerts for current violations over a websocket connection. If there is a connectivity issue between the websocket and the Analyzer detail view server, the Analyzer detail view server does not resume from where it left off. Instead, it starts sending a current set of alerts. You can get the missing alerts using the `getAlerts` call.

Push notification API is only available over HTTPS in sync with the rest of the API.



Request line

The client opens the websocket connection with the given URL and authorization header.

```
wss://<hostname>:8443/pusher/alerts
Authorization:Basic XXXXXXXXX
```

Request example

```
{
  "function":"subscribe",
  "alertIds":[
    11232,
    31323,
    43134
  ]
}
```


Response examples

The following push message format in JSON is sent over the websocket connection. This JSON contains alert violation data for performance metrics.

```
[
  {
    "id": "1",
    "name": "Production VMs High CPU Usage",
    "severity": "warning",
    "from": "20160401_235300",
    "to": "20160401_235700",
    "violations": {
      "vm#vm1234": [
        "20160401_235500",
        "20160401_235600",
        "20160401_235700"
      ]
    }
  }
]
```

The following push message format in JSON is sent over the websocket connection. This JSON contains alert violation data for configuration metrics.

```
[
  {
    "id": "31",
    "name": "vmware_test",
    "severity": "critical",
    "violations": { "vm#Test_192.168.144.140_2": ["20190314_143100"] },
    "Value": ["POWERED_ON"]
  }
]
```

The response message where alert ID does not exist:

```
{
  "meta": {
    "code": "277",
    "failed": [
      {
        "id": 3,
        "code": "FFFFFFFF",
        "message": "alert with id does not exists"
      }
    ]
  }
}
```

The response message where the alert ID is already subscribed:

```
{
  "meta":{
    "code":"277",
    "failed":[
      {
        "id":1,
        "code":"FFFFFFFE",
        "message":"current alert is already subscribed"
      }
    ]
  }
}
```

Further subscription calls are ignored after this stage.

Websocket

Websocket enables bidirectional, message-oriented streaming of text and binary data between client and server. It is the set of standards, out of which API that is supported by browsers is defined by W3C and underlying protocol is maintained by the HyBi Working Group (IETF).

Websocket has been widely adopted in recent times and various client libraries have been developed in many programming languages, including Java that offer emulated websocket API.

For more information, see:

Websocket API: <http://www.w3.org/TR/websockets/>

Websocket protocol : <https://tools.ietf.org/html/rfc6455>

Overview of websockets: <http://chimera.labs.oreilly.com/books/1230000000545/ch17.html>

Subscribing to alert notifications

You can subscribe to alert notifications by configuring the SMTP, Syslog server, and SNMP Manager.



Note: You must configure the SMTP server to receive email alert notifications. The SMTP, Syslog server, and SNMP Manager configuration is a one-time activity that should be performed from the Analyzer detail view UI.

Request line

```
POST baseUrl/alertapi.do?action=subscribeAlertDef
```

The request payload is an array of subscription JSON objects.

Request body

```
[
  {
    "alertId":,
    "subscription":{
      "email":{
        "id":[
          "",
          ""
        ],
        "damping":""
      },
      "syslog":{
        "server":[
          "",
          ""
        ]
      },
      "snmp":{
        "manager":[
          "",
          ""
        ]
      }
    }
  }
]
```

Request example

As part of the subscription, you must specify the notification settings in the notification object.

```
POST /alertapi.do?action=subscribeAlertDef
```

```
[
  {
    "alertId":122131,
    "subscription":{
      "email":{
        "id":[
          "xxx",
          "yyy"
        ],
        "damping":"no_damping"
      },
      "syslog":{
        "server":[
          "hostname:portNo:TCP",

```

```

        "IP:portNo:UDP"
    ]
},
"snmp":{
    "manager": [
        "hostname",
        "IP"
    ]
}
}
}
]

```

You cannot modify an existing subscription, but you can replace it by sending another request which overwrites the original subscription.

Request parameters

Parameter	Type	Description
alertId	Integer	Alert definition ID to be subscribed.
subscription	Object	Subscription details.

Response structure

```

{
  "subscribed":{
    "alertId":[
      1
    ]
  }
}
{
  "failed":[
    {
      "alertId:",
      "error":"",
      "code":""
    }
  ]
}

```

Response examples**Success**

```
{
  "subscribed":{
    "alertId":[
      1
    ]
  }
}
```

Error

```
{
  "failed":[
    {
      "alertId":12,
      "error":"Alert definition with id:[12] not found.",
      "code":"A5040010"
    }
  ]
}
```

SNMP examples

```
{
  "failed":[
    {
      "alertId":10,
      "error":"SNMP Manager(s) [192.168.20.10] does not exist.",
      "code":"A5040010"
    }
  ]
}
```

```
{
  "failed":[
    {
      "alertId":-10,
      "error":"SNMP Manager cannot be subscribed for system level alerts.",
      "code":"A5040010"
    }
  ]
}
{
  "failed":[
    {
      "alertId":-10,
      "error":"SNMP Manager cannot be subscribed for system level alerts.",
```

```

    "code": "A5040010"
  }
]
}

```



Note: This error message is received when you try to subscribe the SNMP notifications for system level alert definitions, the system level alerts do not support SNMP notifications.

Status codes

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
A5010005	Invalid {emailIds} specified [yyy, abc@@gmail.com].	Invalid email IDs are specified.
A5010006	Invalid {damping} specified.	Invalid damping.
A5010006	Invalid {alertId} specified.	Invalid alert ID.
A5010005	Invalid {sysLogServer} specified [hostName].	Invalid Syslog server.
A5040010	Alert definition with id:[12] not found.	Alert definition ID not found.
A5020019	Found invalid properties [resources, unknownKey].	Invalid key is specified.

Unsubscribing an alerts notification

You can unsubscribe existing alert notifications.

Request line

```
POST baseUrl/alertapi.do?action=unsubscribeAlertDef
```

The request payload is a JSON object.

Request body

```
{
  "alertId": [
    ,
  ]
}
```

Request example

```
POST /alertapi.do?action=unsubscribeAlertDef
```

```
{
  "alertId": [
    23423, 542324
  ]
}
```

Request parameters

Parameter	Type	Description
alertId	Integer[]	List of alert definition IDs to unsubscribe.

Response structure

```
{
  "unsubscribed": {
    "alertId": [

    ]
  }
}
{
  "failed": [
```

```
{
  "alertId":,
  "error": "",
  "code": ""
}
]
```

Response example

Success

```
{
  "unsubscribed": {
    "alertId": [
      1
    ]
  }
}
```

Error

```
{
  "failed": [
    {
      "alertId": 12,
      "error": "Alert definition with id:[12] not found.",
      "code": "A6040010"
    }
  ]
}
```

Status codes

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
A6040010	Alert definition with id:[12] is not found.	Invalid alert ID.
A604000D	User [admin] is already unsubscribed from alert definition [1].	User is trying to the unsubscribe alertDef.
A6020019	Found invalid properties [resources, unknownKey].	Invalid key is specified.

Pausing an alert definition evaluation

You can pause the alert definition to stop the evaluation and notification.

Request line

```
POST baseUrl/alertapi.do?action=disableAlertDef
```

The request payload is a JSON object.

Request parameters

Parameter	Type	Description
alertId	Integer[]	List of alert definition IDs to be paused.

Request example

```
POST /alertapi.do?action=disableAlertDef
```

```
{
  "alertId": [
    23423, 542324
  ]
}
```

Response body

```
{
  "disabled":{
    "alertId":[

    ]
  }
}
{
  "failed":[
    {
      "alertId:",
      "error":"",
      "code":""
    }
  ]
}
```

Response examples**Success**

```
{
  "disabled":{
    "alertId":[
      23423
    ]
  }
}
```

Error

```
{
  "failed":[
    {
      "alertId":12,
      "error":"Alert definition with id:[12] not found.",
      "code":"A8040010"
    }
  ]
}
```

For a list of alert API error codes, see [Alert API response error codes and messages \(on page 269\)](#).

Restarting an alert definition evaluation

You can restart a paused alert definition for an alert based on its ID.

Request line

```
POST baseUrl/alertapi.do?action=enableAlertDef
```

The request payload is a JSON object.

Request body

```
{
  "alertId": [
    ,
  ]
}
```

Request example

```
POST /alertapi.do?action=enableAlertDef
```

```
{
  "alertId": [
    23423,
    542324
  ]
}
```

Request parameters

Parameter	Type	Description
alertId	Integer[]	List of alert definition IDs to be restarted.

Response body

```
{
  "enabled": {
    "alertId": [

    ]
  }
}
{
  "failed": [
    {
      "alertId": ,
      "error": ""
    }
  ]
}
```

```

    "code":""
  }
]
}

```

Response examples

Success

```

{
  "enabled":{
    "alertId":[
      23423
    ]
  }
}

```

Error

```

{
  "failed":[
    {
      "alertId":23423,
      "error":"Alert definition with id:[12] not found.",
      "code":"A7040010"
    }
  ]
}

```

For a list of alert API error codes, see [Alert API response error codes and messages \(on page 269\)](#).

Deleting an alert definition

You can delete one or more alert definitions and all of the associated information.

Request line

```
POST baseUrl/alertapi.do?action=deleteAlertDef
```

The request payload is a JSON object.

Request body

```

{
  "alertId":[
    ,

```

```
]
}
```

Request example

```
POST /alertapi.do?action=deleteAlertDef
```

```
{
  "alertId":[
    23423, 542324
  ]
}
```

Request parameters

Parameter	Type	Description
alertId	Integer[]	List of alert definition IDs, to be deleted.

Response body

```
{
  "deleted":{
    "alertId":[

    ]
  }
}
{
  "failed":[
    {
      "alertId:",
      "error:"",
      "code":""
    }
  ]
}
```

Response examples

Success

```
{
  "deleted":{
    "alertId":[
      23423
    ]
  }
}
```

```
}
}
```

Error

```
{
  "failed": [
    {
      "alertId": 23423,
      "error": "Alert definition with id:[3] not found.",
      "code": "A4040010"
    }
  ]
}
```

For a list of alert API error codes, see [Alert API response error codes and messages \(on page 269\)](#).

Updating an alert threshold

You can update the threshold value for the resources and add or remove the resources from the alert definition.

For example: suppose you have created an alert on the MPB (raidMPB#12345-MPB-2MA) and you want to update this value with 40. See the following request:

```
update": [
  {
    "signature": "raidMPB#12345-MPB-2MA",
    "threshold": [
      40
    ]
  }
]
```

After updating, the raidMPB#12345-MPB-2MA resource, an alert will be generated every time it exceeds the updated threshold value.

Request line

```
POST baseUrl/alertapi.do?action=updateAlertDef
```

Request payload is an array of JSON objects.

Request body

```
[
  {
```

```

    "alertId":1,
    "update":[
      {
        "signature":"",
        "threshold":[

        ]

      }
    ],
    "remove":[
      ""
    ]
  }
]

```

Request example

```
POST /alertapi.do?action=updateAlertDef
```

```

[
  {
    "alertId":1,
    "update":[
      {
        "signature":"vm#vm1",
        "threshold":[
          40
        ]
      }
    ],
    "remove":[
      "vm#vm2"
    ]
  }
]

```

Request parameters

Parameter	Type	Description
alertId	Integer	ID of the alert definition.
update	Object[]	Collection of resources to be added or updated to the alert definition.
update:signature	String	Signature of the resource.

Parameter	Type	Description
update:threshold	Float[]	(Optional) Threshold value for a resource. Float: <ul style="list-style-type: none"> Maximum value = Java Float.Max value i.e., +3.4028235 E 38. Minimum value = Java Float.Min value i.e., - 3.4028235 E 38.
update:threshold (for having condition)	String	(Optional) Threshold value for having condition. The threshold field for having condition supports alphanumeric, hyphen (-), underscore (_), comma (,), colon (:), at (@), slash (/), dot (.), and space.
remove	String[]	Collection of resources to be removed from the alert definition. Note: If you have created an alert definition using <code>resource_mql</code> , then you cannot remove the resources from the alert definition.



Note: A parameter specified as `a:b` indicates the `b` field of JSON object `a`.

Response examples

Success

```
[
  {
    "alertId":23423,
    "status":"updated"
  }
]
```


Error

```
[
  {
    "alertId":1,
    "status":"partially updated",
    "failed":[
      {
        "reason":"Failed to remove resource [vm#vm2]. It doesnot exist in alert
definition with id:[1].",
        "code":"A304000B",
        "signature":"vm#vm2"
      }
    ]
  }
]
```

Return codes

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
500	SERVER ERROR	API operation request failed. Check the response body for details.

Response codes

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
A304000C	Failed to add resource [vm#vm16], resource does not exist.	An attempt to add a resource that is not available in the cache.
A304000B	Failed to remove resource [vm#vm5], it does not exist in the alert definition with id:[2].	An attempt to remove a resource that has not been defined for the alert.
A3040010	Alert definition with id:[12] is not found.	The alert definition is not be available for specified ID.

Response code	Message	Condition
A3020019	Found invalid properties [resources, unknownKey].	Invalid key is specified.
A3020023	Failed to edit alert [173] using alert API as it is created from UI	The UI alerts added in the Analyzer detail view v10.5.0-01 onwards cannot be edited through REST API.
A3040013	Application encountered an internal error.	API server encountered an unknown error.

Getting the status of an alert request

You can obtain the status of a previously generated alert request. The status indicates whether the alert request has completed and provides any result details. If any of your requests fail due to a timeout or for any other reason, you will receive a request ID which you can use to obtain further information concerning the request. If the requested status does not exist, an error message is generated in a response message.

Request line

```
GET baseURL/alertapi.do?action=getStatus
```

Request parameters

Parameter	Type	Description
requestId	String	Asynchronous request ID.

Request example

```
GET /alertapi.do?action=getStatus&requestId=12345
```

Getting an alert audit log

You can obtain audit logs based on the following criteria: date, request ID, user, user agent, HTTP method, client IP, API function, request length, response code, and service time based on the specified search criteria.

Request line

```
baseUrl/alertapi.do?action=getAuditLogs
```

The request payload is a JSON object in the case of an HTTP POST request.

Request body

```
{
  "function": [
    "",
    ""
  ],
  "tw": [
    "",
    ""
  ]
}
```

Request examples

Audit logs for all request IDs in the range of 12355 to 12365 in the last 24 hours:

```
GET /alertapi.do?action=getAuditLogs&requestId=123[5-6]5
```

Audit logs for API functions `deleteAlertDef` and `disableAlertDef` for a specific timeline:

```
POST /alertapi.do?action=getAuditLogs
```

```
{
  "function": [
    "deleteAlertDef",
    "disableAlertDef"
  ],
  "tw": [
    "20151102_200000",
    "20151103_000000"
  ]
}
```

Request parameters

Parameter	Values	Description
tw	String[]	(Optional) JSON list of string where the first element represents the start time and

Parameter	Values	Description
		<p>the second element represents the end time for the audit.</p> <p>Date format: yyyyMMdd_HH:mm:ss</p> <p>Time zone: UTC</p> <p>Default: Application considers last 24 hours for audit.</p>
requestId ^r	String	<p>(Optional) Request ID whose audit log is required.</p> <p>Supports regular expressions</p> <p>Default: .*</p>
function	String	<p>(Optional) List of strings, each value representing the API function name to be searched.</p>
user ^r	String	<p>(Optional)</p> <p>User name whose audit log is required.</p> <p>If you want to give audit logs a specific username, then regular expression special characters in the user name should be escaped.</p> <p>Supports regular expressions</p> <p>Default: .*</p>
clientIp ^r	String	<p>(Optional) Client IP address for which audit log is required.</p> <p>Supports regular expressions</p> <p>Default: .*</p>

Parameter	Values	Description
responseCode ^r	String	(Optional) HTTP response code for which audit log is required. Supports regular expressions Default: .*



Note: ^r represents parameters which support regular expressions where their default value is .*

Response body

```
[
  {
    "date": "",
    "requestId": "",
    "user": "",
    "userAgent": "",
    "clientIp": "",
    "action": "",
    "requestLength": ,
    "httpMethod": "",
    "responseCode": ,
    "serviceTime":
  }
]
```

Response example

```
[
  {
    "date": "20160314_182254",
    "requestId": "1466911767_1",
    "user": "admin",
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.116 Safari/537.36",
    "clientIp": "0:0:0:0:0:0:0:1",
    "action": "createAlertDef",
    "requestLength": 756,
    "httpMethod": "POST",
    "responseCode": 200, "serviceTime": 673
  },
  {
    "date": "20160314_182303",
    "requestId": "1261810925_2",
```

```

    "user": "admin",
    "userAgent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
    Gecko) Chrome/48.0.2564.116 Safari/537.36",
    "clientId": "0:0:0:0:0:0:1",
    "action": "createAlertDef",
    "requestLength": 754,
    "httpMethod": "POST",
    "responseCode": 200,
    "serviceTime": 13
  }
]

```

Status codes

Status code	Message	Description
200	OK	Request was successful, and the response body contains the requested log details.
206	PARTIAL CONTENT	Request is in progress, and the incremental content is available in the response body.
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.
409	CONFLICT	Resource already exists.

Status code	Message	Description
413	REQUEST ENTITY TOO LARGE	Server cannot process the request because the request entity is too large for processing.
500	SERVER ERROR	API operation request failed. Check the response body for details.
501	NOT IMPLEMENTED	API server does not support the specified action.
503	SERVICE UNAVAILABLE	API server cannot process the request, possibly because the API server instance is not running.

Evaluating alert violations

This section describes how to select the data interval for alert evaluation, edit the threshold value, and how the data hole is considered while evaluation.

Selecting the data interval

You can select the data interval when the multiple data intervals are collected for a time-series attribute. In this scenario the alert evaluation is performed only on one qualified interval that is received first in a day. The interval is considered as qualified if it meets the m/n criteria. The data importer picks the different qualified interval on the next day or when the day is changed.

For example: Alert definition criteria = 1/5 (1 violation in 5 minutes)

If the data is received in the following intervals: 600-second, 300-second, and 60-second, the following three scenarios are possible:

- Case 1: If the importer first receives the data interval of 600-second:

- $m = 1$
- $n = 5$
- Number of samples in n minutes = $(n/\text{data interval in minutes}) = (5/10) = 0$

Therefore, zero data samples occur in n minutes (5 minutes).

In this scenario, the 600-second interval does not qualify for alert evaluation.



Note: In this scenario, the qualified interval for a violation computation is either 300-second or 60-second, whichever is received first after 600-second. This qualified interval for alert violation computation is only for current day data. It changes after the data is received the next day. On the next day, the system performs the same calculation to determine the qualified interval.

- Case 2: If the importer first receives the data interval of 300-second:

- $m = 1$
- $n = 5$
- Number of samples in n minutes = $(n/\text{data interval in minutes}) = (5/5) = 1$

Therefore, one data sample occurs in n minutes (5 minutes).

In this scenario, the 300-second interval qualifies for alert evaluation.

Once the importer determines the qualified data interval, it skips all other incoming intervals for that day irrespective of whether they qualify or not. Here, since the 300-second interval is qualified first, a 60-second interval is ignored even if it qualifies.

- Case 3: If the importer first receives the data interval of 60-second:

- $m = 1$
- $n = 5$
- Number of samples in n minutes = $(n/\text{data interval in minutes}) = (5/1) = 5$

Therefore, 5 data samples occurs in n minutes (5 minutes).

In this scenario, the 60-second interval qualifies for alert evaluation.

Data hole consideration in the alert violation computation

If a data hole is detected while importing the time-series data into the Analyzer detail view database through the Database API or Data Importer, the importer considers the data hole as a valid nonviolated (green) value when evaluating the time-series data.

Example:

- Alert definition criteria = 2/5 (2 violations in 5 minutes)
- Alert definition threshold = greater than 10
- Inserted time-series data = 15, 20, null, null, null, null, 40, 50
- Data interval = 1 minute
- Data start time = 10:10 (HH:MM)

In the above scenario, violations are reported at time stamp = 10:11, 10:12, 10:13, 10:14, 10:17.

Even though 10:12, 10:13, 10:14, and 10:15 are a data hole, they are considered during alert evaluation as nonviolating values.

State changes are reported at = 10:11 (Green --> Red), 10:15 (Red --> Green), and 10:17 (Green --> Red).

Editing the threshold value

You can edit the threshold value of a specific resource. When the threshold value is edited, the updated value is immediately applied to the time-series data that is imported after the update.

For example, see the following scenarios.

Alert definition criteria = 1/1 (1 violation in 1 minute)

Alert definition threshold = less than 20.

Alert definition is attached to resources R1, R2, and R3, and data importer is importing the time-series data of resources R1, R2, and R3.

- **Scenario one:** Before the threshold is updated.

Series of R1, R2, and R3:

(R1) t1----- 20-----t2

(R2) t1----- 20-----t2----- 20-----t3

(R3) t1----- 20-----t2----- 20-----t3

Where t1, t2, and t3 are the data time stamps.

In the above series, the threshold value is mentioned between two time stamps.

Here, threshold 20 is used during time window t1 to t2 for resources R1, R2, and R3.

The same threshold value 20 is also used during time window t2 to t3 for resources R2 and R3.

If the threshold is updated to 40, then this updated value is applied to the data that comes after the threshold is updated.

- **Scenario two:** After the threshold is updated.

Series of R1, R2, and R3:-

(R1) t1----- --20-----t2----- --40-----t3-----40-----t4

(R2) t1----- --20-----t2-----20-----t3--- --40-----t4

(R3) t1----- 20-----t2-----20-----t3-----40-----t4

In the above series, the updated threshold value is used immediately for alert evaluation after the update.

Here, threshold 40 is used during time window t2 to t4 for resource R1.

The same threshold value 40 is used during time window t3 to t4 for resources R2 and R3.

Chapter 11: User, license, and product URL API resources

This section describes the following API resources:

- **User management API:** Create, update, and delete a user on the Analyzer detail view. Once the Ops Center Analyzer application (third-party application) registers its users with the Analyzer detail view, the Ops Center Analyzer users can access the Analyzer detail view seamlessly. This API handles the bootstrap handshake between the Analyzer detail view server and the third-party application.
- **License API:** Register the third-party application on the Analyzer detail view server.
- **Product (Analyzer detail view) launch URL API:** Launch the Analyzer detail view from the third-party application. This API authenticates the user, creates a session on the server, and returns a URL with session details to the third-party application. The Analyzer detail view is launched seamlessly (without a login screen).

Access rights for API functions

Alert definitions are shared with users on the Analyzer detail view. Access to some functions is restricted to the alert definition owner.

The following table describes the function available based on the user type.

Function	Admin user	Normal user
createUser	Yes	No
updateUser	Yes (for all users)	Yes (self)
changePassword	Yes (for all users)	Yes (self)
deleteUser	Yes (for all users)	Yes (self)
registerLicense	Yes	No
getAppUrl	Yes	Yes

API authentication

The database and alert APIs are authenticated based on the users. The API user must be a user on the Analyzer detail view. The user must submit the API request with a user authentication token available from the Analyzer detail view UI.

When creating a user or registering a license, the user might not be available on the Analyzer detail view. A bootstrap handshake is handled when there are no users on the Analyzer detail view.

License key authentication

A special *license authentication key* is distributed separately from the regular license.

You can send the *license authentication key* string as value of the `x-authorization` header in a request:

```
x-authorization: <License Authentication Key>
```

User authentication token

You can use a user-specific authentication token from the Analyzer detail view UI.

To authorize the request, you can send the authentication token as a value of `Authorization` header in a request:

```
Authorization: Basic <space><Authentication Token>
```

Refer to "Using the authentication token" section in the *Hitachi Ops Center Analyzer Analyzer detail view User Guide*.

Authentication details for API functions

The following table maps the API functions and their authentication.

Function	License authentication	User authentication
createUser	Yes	No
updateUser	No	Yes
changePassword	No	Yes
deleteUser	Yes	No
registerLicense	Yes	No
getAppUrl	No	Yes

API request headers

The following are the request headers for all the API requests.

Request Header	Description
Authorization	Authorize a request using the user authentication token.
x-authorization	Authorize a request using the license authentication key.



Note: Any one of the above request headers is mandatory. See [Authentication Details For API Functions \(on page 228\)](#) to pick the appropriate function along with the request header.

User management API design

The user management APIs follow the same high-level design concepts used in the database API:

- The request URLs of the database API and user API have a common endpoint. However, these URLs are identified by the functions. For example, the general URL format is `https://localhost:8443/userapi.do?action=XXXX`, where XXX is the API function.
- API functionality is supported only through two HTTP methods: GET and POST. DELETE, PUT, or other HTTP methods are not considered.
- The GET method is not supported for functions causing side effects on the server.
- The API platform is reused. Therefore, the API reuses the partial completion and status API.
- The API also uses HTTP basic authentication.
- Response codes and error object structure are same as the database API.
- The API is supported only over HTTPS.

Creating a user

You can create multiple users in a single request on the Analyzer detail view.

Request line

```
POST baseUrl/userapi.do?action=createUser
```

Request payload is an array of user objects.

Request body

```
[
  {
    "username": "",
    "password": "",
    "firstname": "",
    "lastname": "",
    "email": "",
    "type": "",
    "role": "",
    "locale": ""
  }
]
```

Request examples

Create local user:

```
POST /userapi.do?action=createUser x-authorization:<license authentication key>
```

```
[
  {
    "username": "john.s",
    "password": "axCd2!43mn",
    "firstname": "John",
    "lastname": "Smith",
    "email": "john@example.com",
    "type": "local",
    "role": "admin",
    "locale": "en-us"
  }
]
```

Create Active Directory user:

```
POST /userapi.do?action=createUser x-authorization:<license authentication key>
```

```
[
  {
    "username": "john.s",
    "email": "john@xxx.com",
    "type": "ad",
    "role": "admin",
    "locale": "en-us"
  }
]
```

Request parameters

Parameter	Values	Description	Required	
			Local user	Active Directory user
username	String	User name for a Local or Active Directory user.	Yes Up to 20 characters except Greater than and Less than signs(< >), Square brackets ([]), spaces, double quotation mark ("), colon (:).	Yes
password	String	Password for the user name. <ul style="list-style-type: none"> ▪ Password must be a minimum of six characters, by default. ▪ You cannot reuse the previous six passwords when resetting the password. ▪ Password must not contain the user name string in it. 		

Parameter	Values	Description	Required	
			Local user	Active Directory user
		<ul style="list-style-type: none"> ▪ Password must contain one uppercase letter and one lowercase letter. ▪ Password must contain one digit and one special character. The following special characters are supported: !, ~, ` , @, #, \$, %, ^, &, *, (,), -, _ , +, =. <p>Note: The password minimum requirements are not applicable for Active Directory users.</p>		
firstname	String	First name of a normal user. Up to 30 characters except <,>,[,]	Yes	N/A
lastname	String	Last name of a normal user. Up to 30 characters except <,>,[,]	Yes	N/A

Parameter	Values	Description	Required	
			Local user	Active Directory user
email	String	Email of a normal or Active Directory user. Valid email address up to 80 characters.	Yes	Yes
type	String	User type as Active Directory user or local user.	No Default: local	Yes
role	String	Role of a user: Administrator or Normal	Yes	Yes
locale	String	Locale of a user: en-us or ja-jp	No Default: en-us	No Default: en-us

Response body

```
{
  "created": {
    "sandp.b": "<auth_token>"
  }
}
{
  "failed": [
    {
      "username": "",
      "code": "",
      "reason": ""
    }
  ]
}
```

Response examples

Success message:

```
{
  "created": {
    "john.s": "<auth_token>"
  }
}
```

```
}
}
```

Error message:

```
{
  "failed":[
    {
      "username":"john.s",
      "code":"C1040005",
      "reason":"User with {username} 'john.s' already exists on server"
    }
  ]
}
```

Status codes

Status code	Message	Description
277	PARTIAL OK	Request completed successfully with some failures. Check the response body for details.
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
C10100E0	HTTP method [GET] is not supported for action [createUser].	HTTP method is not POST.
C1010001	Authorization failed.	Request is sent with an invalid authentication token.
C10400E9	Invalid Authentication	User does not have admin rights.
C1010002	Invalid JSON Request	Specified JSON input is invalid.
C1010003	Mandatory parameter {username} is not specified	Username is not specified in the parameter {username}.
C1010004	Invalid value specified for parameter {username}.	Specified username is invalid.
C1010003	Mandatory parameter {password} is not specified.	Password is not specified in the parameter {password}.
C1010004	Invalid value specified for parameter {password}.	Specified password is invalid.
C1010003	Mandatory parameter {firstname} is not specified	First name is not specified in the parameter {firstname}.
C1010004	Invalid value specified for parameter {firstname}.	Specified firstname is invalid.
C1010003	Mandatory parameter {lastname} is not specified.	Last name is not specified in the parameter {lastname}.
C1010004	Invalid value specified for parameter {lastname}.	Specified lastname is invalid.
C1010003	Mandatory parameter {email} is not specified.	Email address is not specified in the parameter {email}.
C1010004	Invalid value specified for parameter {email}.	Specified email address is invalid.
C1010004	Invalid value specified for parameter {type}.	Specified user type is invalid.
C1010003	Mandatory parameter {role} is not specified.	User role is not specified in the parameter {role}.
C1010004	Invalid value specified for parameter {role} specified.	Specified user role is invalid.

Response code	Message	Condition
C1010004	Invalid value specified for parameter {locale}.	Specified locale is invalid.
C1040005	User with {username} (admin role) already exists on the server.	User already exists with the same username.

Updating specific user information

You can update the email and locale of a user. You cannot update the first name and last name of a user.

Request line

```
POST baseURL/userapi.do?action=updateUser
```

Request payload is a JSON user object.

Request body

```
{
  "username": "",
  "email": "",
  "locale": ""
}
```

Request example

```
POST /userapi.do?action=updateUser Authorization: Basic <user authentication token>
```

```
{
  "username": "john.s",
  "email": "johns@example.com",
  "locale": "ja-jp"
}
```

Request parameters

Parameter	Type	Description
username	String	Up to 20 characters except <, >, [,]

Parameter	Type	Description
password	String	<p>Password for the user name.</p> <ul style="list-style-type: none"> ▪ Password must be a minimum of six characters, by default. ▪ You cannot reuse the previous six passwords when resetting the password. ▪ Password must not contain the user name string in it. ▪ Password must contain one uppercase letter and one lowercase letter. ▪ Password must contain one digit and one special character. The following special characters are supported: !, ~, `, @, #, \$, %, ^, &, *, (,), -, _, +, =. <p>Note: The password minimum requirements are not applicable for Active Directory users.</p>

Response body

```
{
  "updated":""
}
```

Response example

```
{
  "updated":"john.s"
}
```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
C20100E0	Http method [GET] is not supported for action [updateUser].	HTTP method is not POST.
C2010001	Authorization failed.	Request is sent with an invalid authentication token.
C2040001	Invalid Authentication.	User does not have admin rights.
C2010002	Invalid JSON Request.	Specified JSON input is invalid.
C2010003	Mandatory parameter {username} is not specified	Username is not specified in the parameter {username}.
C2010004	Invalid value specified for parameter {username}.	Specified username is invalid.
C2010004	Invalid value specified for parameter {email}.	Specified email address is invalid.

Response code	Message	Condition
C2010004	Invalid value specified for parameter {locale}.	Specified locale is invalid.
C2040006	User name that is specified in the parameter {username} does not exist.	Specified username does not exist.

Changing a password

You can change the password of the local user. You cannot change the password of the Active Directory users.

HTTP request syntax (URI)

```
POST baseURL/userapi.do?action=changePassword
```

Request payload is a JSON user object.

Request body

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

Request example

```
POST /userapi.do?action=changePassword
Authorization: Basic <user authentication token>
```

```
{
  "username": "john.s",
  "password": "axCd2!43mn"
}
```

Request parameters

Parameter	Type	Description
username	String	User name of the normal user. Up to 20 characters except <, >, [,]

Parameter	Type	Description
password	String	<p>Password for the user name.</p> <ul style="list-style-type: none"> ▪ Password must be a minimum of six characters, by default. ▪ You cannot reuse the previous six passwords when resetting the password. ▪ Password must not contain the user name string in it. ▪ Password must contain one uppercase letter and one lowercase letter. ▪ Password must contain one digit and one special character. The following special characters are supported: !, ~, ` , @, #, \$, %, ^, &, *, (,), -, _ , +, =. <p>Note: The password minimum requirements are not applicable for Active Directory users.</p>

Response body

```
{
  "updated": {
    "john.s": "<auth_token>"
  }
}
```

Response example

```
{
  "updated": {
    "john.s": "<auth_token>"
  }
}
```


Return codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
C30100E0	HTTP method [GET] is not supported for action [changePassword]	HTTP method is not POST.
C3010001	Authorization failed.	Request is sent with an invalid authentication token.
C3040001	Invalid Authentication.	User does not have admin rights.
C3010002	Invalid JSON Request.	Specified JSON input is invalid.
C3010003	Mandatory parameter {username} is not specified	Username is not specified in the parameter {username}.
C3010004	Invalid value specified for parameter {username}.	Specified username is invalid.
C3010003	Mandatory parameter {password} is not specified.	Password is not specified in the parameter {password}.

Response code	Message	Condition
C3010004	Invalid value specified for parameter {password}.	Specified password is invalid.
C3040007	Operation {changePassword} is not allowed for Active Directory users.	Active directory user is specified in the parameter {changePassword}.
C3040009	Password did not meet the criteria.	Specified password did not meet the criteria.
C3040006	User name that is specified in the parameter {username} does not exist.	Specified username does not exist.

Deleting a specific user from the Analyzer detail view

You can delete a user from the Analyzer detail view. You cannot delete the default administrator user (admin user).

Request line

```
POST baseURL/userapi.do?action=deleteUser
```

Request payload is a JSON object.

Request body

```
{
  "username":""
}
```

Request example

```
POST /userapi.do?action=deleteUser x-authorization:<license authentication key>
```

```
{
  "username":"john.s"
}
```

Request parameters

Parameter	Type	Description
username	String	User name of the user to be deleted. Up to 20 characters except <, >, [,]

Response body

```
{
  "deleted": ""
}
```

Response example

```
{
  "deleted": "john.s"
}
```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
404	NOT FOUND	API server could not find a resource matching the request.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
C40100E0	HTTP method [GET] is not supported for action [deleteUser]	HTTP method is not POST.
C4010001	Authorization failed.	Request is sent with an invalid authentication token.
C4010002	Invalid JSON Request.	Specified JSON input is invalid.
C4010003	Mandatory parameter {username} is not specified.	Username is not specified in the parameter {username}.
C4010004	Invalid value specified for parameter {username}.	Specified username is invalid.
C4040006	User name that is specified in the parameter {username} does not exist.	Specified username does not exist.

Registering a license

You can register a license on the Analyzer detail view. If the license already exists, the new license overwrites the old license. To register the license, you can either upload a license file or use the JSON with the license keys.

Request line

```
POST baseUrl/licenseapi.do?action=registerLicense
```

Request payload is a JSON array with the license keys or license file.

Request body

JSON array with the license keys or license file :

```
[
  XXXX-XXXX-XXXX-XXXX,
  YYYY-YYYY-YYYY-YYYY
]
```

Request examples

Upload the license file:

```
POST /licenseapi.do?action=registerLicense&type=file
x-authorization:<license authentication key>
Content-Type:text/plain
```

Send the license keys:

```
POST /licenseapi.do?action=registerLicense
x-authorization:<license authentication key>
```

```
[
  XXXX-XXXX-XXXX-XXXX,
  YYYY-YYYY-YYYY-YYYY
]
```

Request parameters

Parameter	Type	Description
type	String	<p>The type is file.</p> <p>To upload the license, set the parameter value to <code>file</code> and the header <code>Content-Type</code> to <code>text/plain</code>.</p> <p>Required if payload is a license file.</p> <p>Optional if payload is a JSON with license keys.</p>

Response body

```
{
  "status": ""
}
{
  "code": "",
  "error": ""
}
{
  "failed":[
    "xxxx-xxxx-xxxx-xxxx-xxxx"
  ],
  "registered":[
    "xxxx-xxxx-xxxx-xxxx-xxxx"
  ]
}
```

```
]
}
```

Response examples

Response message when the correct license file is provided:

```
{
  "status": "License registered successfully"
}
```

Response message when the incorrect license file is provided:

```
{
  "code": "D1040002",
  "error": "Specified license file is invalid"
}
```

Response message when one of the provided license keys are invalid:

```
{
  "failed": [
    "xxxx-xxxx-xxxx-xxxx-xxxx"
  ],
  "registered": [
    "xxxx-xxxx-xxxx-xxxx-xxxx"
  ]
}
```

Status codes

Status code	Message	Description
400	BAD REQUEST	Request URL or request body validation failed. Check the response body for details.
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message	Condition
D10100E0	HTTP method [GET] is not supported for action [registerLicense].	HTTP method is not POST.
D1010001	Authorization failed.	Request is sent with an invalid authentication token.
D1040003	Specified license type is invalid.	License type is incorrect.
D1040004	Specified JSON input is invalid.	Specified JSON input is invalid.

Getting the URL of the Analyzer detail view

You can get the URL in order to launch the Analyzer detail view. The API authenticates a user and creates a user session, to launch the Analyzer detail view in a logged-in state for the specified user.

Request line

```
GET baseUrl/app.do?action=getAppUrl&landing=home
```

Request payload is a query parameter.

Request parameters

Parameter	Type	Description
landing	String	(Optional) A parameter to send additional information to the Analyzer detail view, so that the landing page can be changed as per your requirements. The option is: default

Request example

```
GET /app.do?action=getAppUrl&landing=default
Authorization: Basic <user authentication token>
```

Response body

```
{
  "url":""
}
```

Response example

The application URL with the embedded session ID is returned. The session is created for a user based on the authentication token that is used to validate the API request.

```
{
  "url":" https://hdca.company-systems.com/launchApp.htm?
launchId=14nurcaogcq89mvnaqy19dk3j"
}
```

Return codes

Status code	Message	Description
401	UNAUTHORIZED	Supplied authentication token is invalid or does not have the appropriate credentials to access the resource.
405	METHOD NOT ALLOWED	Request HTTP method is not allowed for the operation.

To view the list of common error code see [Alert API response error codes and messages](#)

Response codes

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Detailed Error Message	Condition
C60100E0	HTTP method [POST] is not supported for action [getAppUrl]	HTTP method is not GET.

Response code	Detailed Error Message	Condition
C60100E9	Authorization failed.	Request is sent with an invalid authentication token.

Appendix A: API server response error codes and messages

This appendix lists the common and alert API server response codes and messages.

API server response error codes and messages

Common API response error codes and messages

The following table lists the common API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message
120000E4	Mandatory parameter {dataset} not specified.
120000E9	License is not valid.
120000E9	Authorization needed.
120000E4	Dataset with {name}:[ds1] does not exist.
120000E0	Http method [POST] is not supported for action [retrieveDataset].
120000E2	Application encountered an internal error.
150000E4	Mandatory parameter {dataset} not specified.
1502000F	Invalid value specified for parameter {clearSchema}.
150000E9	License is not valid.
150000E9	Authorization needed.
150000E4	Dataset with {name}:[ds1] does not exist.
150000E0	Http method [POST] is not supported for action [clearDataset].

Response code	Message
150200E2	Application encountered an internal error.
210000E4	Mandatory parameter {dataset} not specified.
21010002	Invalid JSON Request.
21010008	Mandatory parameter {dataSubsetId} not specified.
2101000F	Invalid value specified for parameter {dataSubsetId}. It must be of type string.
2102000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} length must be between 2-32 characters.
2102000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} contains invalid characters. Valid characters are [alphanumeric, - and _].
21010008	Mandatory parameter {dataSubsetType} not specified.
2101000F	Invalid value specified for parameter {dataSubsetType}. It must be of type string.
2102000F	Invalid value specified for parameter {dataSubsetType}. {dataSubsetType} length must be between 2-32 characters.
2102000F	Invalid value specified for parameter {dataSubsetType}. {dataSubsetType} contains invalid characters. Valid characters are [alphanumeric, - and _].
2101000C	Parameter {dataset} not supported.
210000E9	License is not valid.
210000E9	Authorization needed.
210000E4	Dataset with {name} [ds1] does not exist.
210000E0	Http method [GET] is not supported for action [createDataSubset].
21050004	DataSubset with {dataSubsetId} [US-NorthAmerica] already exists.
210100E2	Application encountered an internal error.

Response code	Message
220000E4	Mandatory parameter {dataset} not specified.
22040008	Mandatory parameter {dataSubsetId} not specified.
2202000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} length must be between 2-32 characters.
2202000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} contains invalid characters. Valid characters are [alphanumeric, - and _].
220000E9	License is not valid.
220000E9	Authorization needed.
220000E4	Dataset with {name}:{ds1} does not exist.
22050005	DataSubset with {dataSubsetId} [US-NorthAmerica] does not exist.
220000E0	Http method [POST] is not supported for action [retrieveDataSubset].
220400E2	Application encountered an internal error.
41050004	Attribute definition with {id}:[cpuUsage] already exists.
4102000F	Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are [alphanumeric, - and _].
4102000F	Invalid value specified for parameter {id}. Dynamic attributeId must start with [^].
4102000F	Invalid value specified for parameter {id}. Specified regular expression is invalid.
41050004	Attribute definition with {id}:[cpuUsage] already exists.
41050004	Attribute definition with {id}:[^byteFlow_(\\d{1,5})_(\\d{1,5})_([A-Za-z]+[0-9]+)\$] already exists.
4102000F	Invalid value specified for parameter {id}. {id} length must be between 2-32 characters.

Response code	Message
41020008	Mandatory parameter {type} not specified.
4102000F	Invalid value specified for parameter {type}. Valid values: scalar and timeseries.
41020007	No regular expression is specified.
4102000F	Invalid value specified for parameter {bandFactor}. Band factor must be one of [1.0, 0.1, 0.01, 0.001, 0.0001].
4102000F	Invalid value specified for parameter {unit}. {unit} length must be between 2-32 characters.
4102000F	Invalid value specified for parameter {unit}. {unit} contains invalid characters. Valid characters are [alphanumeric, - and _].
4102000F	Invalid value specified for parameter {name}. {name} length must be between 2-128 characters.
4102000C	Parameter {bandFactor} not supported. Parameter {bandFactor} is only supported for timeseries attributes.
4102000F	Invalid value specified for parameter {id}. Dynamic attributeId must start with [^].
4102000C	Parameter {dynamic} not supported. Parameter {dynamic} is only supported for timeseries attributes.
41040007	No Attribute definition is specified.
41010002	Invalid JSON Request.
41050008	Mandatory parameter {id} not specified.
410000E9	License is not valid.
410000E9	Authorization needed.
410000E4	Dataset with {name}:{ds1} does not exist.
410000E0	Http method [PUT] is not supported for action [createAttributeDef].
410200E2	Application encountered an internal error.

Response code	Message
42050005	Attribute definition with {id}:[cpuUsage] does not exist.
4202000F	Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are alphanumeric, -.
4202000F	Invalid value specified for parameter {id}. {id} length must be between 2-32 characters.
42020007	No regular expression is specified.
42010002	Invalid JSON Request.
420000E9	License is not valid.
420000E9	Authorization needed.
420000E4	Dataset with {name}:[ds1] does not exist.
420000E0	Http method [PUT] is not supported for action [retrieveAttributeDef].
420100E2	Application encountered an internal error.
43050005	Attribute definition with {id}:[cpuUsage] does not exist.
4302000F	Invalid value specified for parameter {unit}. {unit} contains invalid characters. Valid characters are [alphanumeric, - and _].
4302000F	Invalid value specified for parameter {unit}. {unit} length must be between 2-32 characters.
4302000F	Invalid value specified for parameter {name}. {name} length must be between 2-128 characters.
4302000F	Invalid value specified for parameter {id}. {id} length must be between 2-32 characters.
43050014	Cannot update read-only 'Attribute definition'.
43040007	No Attribute definition is specified.
43010002	Invalid JSON Request.
43050008	Mandatory parameter {id} not specified.

Response code	Message
420000E4	Mandatory parameter {dataset} not specified.
430000E9	License is not valid.
430000E9	Authorization needed.
430000E4	Dataset with {name}:[ds1] does not exist.
430000E0	Http method [GET] is not supported for action [updateAttributeDef].
430500E2	Application encountered an internal error.
44050005	Attribute definition with {id}:[cpuUsage] does not exist.
4402000F	Invalid value specified for parameter {id}. {id} contains invalid characters. Valid characters are alphanumeric, -.
4402000F	Invalid value specified for parameter {id}. {id} length must be between 2 to 32 characters.
44050011	Attribute definition deletion failed due to dependency on one or more Resource definition.
44050014	Cannot deleted read-only 'Attribute definition'.
44010002	Invalid JSON Request.
44040007	No Attribute definition is specified.
440000E9	License is not valid.
440000E9	Authorization needed.
440000E4	Dataset with {name}:[ds1] does not exist.
440000E0	Http method [PUT] is not supported for action [deleteAttributeDef].
440400E2	Application encountered an internal error.
31050004	Resource definition with {type}:[vm] already exists.
31010008	Mandatory parameter {type} not specified.
3101000F	Invalid value specified for parameter {type}. It must be of type string.

Response code	Message
3101000F	Invalid value specified for parameter {type}. Resource {type}:[] length must be between 1-32 characters.
3101000F	Invalid value specified for parameter {type}. {<?<} contains invalid characters. Valid characters are [alphanumeric, - and _].
3101000F	Invalid value specified for parameter {name}. It must be of type string.
3101000F	Invalid value specified for parameter {name}. {name} length must be between 1 to 32 characters.
3101000F	Invalid value specified for parameter {attributes}. It must be of type JSON list of string.
3101000F	createResourceDef failed for {type}:[host]. Create attribute definition for [cpuUsage].
3101000F	Invalid value specified for parameter {relations}. It must be of type JSON list of string.
3101000F	Invalid value specified for parameter {relations}. {<>?} contains invalid characters. Valid characters are [alphanumeric, - and _].
3101000F	Invalid value specified for parameter {relations}. Resource {type}:[] length must be between 1-32 characters.
3101000F	Invalid value specified for parameter {additionalProperties}. It must be of type JSON object.
3101000F	Invalid value specified for parameter {additionalProperties}. Maximum 128 additional properties can be provided in resource definition.
3101000F	Invalid value specified for parameter {additionalProperties}. Additional properties key can not be one of [type, configAttributes, displayName, directRelations]

Response code	Message
3101000F	Invalid value specified for parameter {additionalProperties}. {@@} contains invalid characters. Valid characters are [alphanumeric, - and _].
3101000F	Invalid value specified for parameter {additionalProperties}. {} length must be between 1-32 characters.
3101000F	Invalid value specified for parameter {additionalProperties values}. It must be of type string.
3101000F	Invalid value specified for parameter {additionalProperties}. {key} length must be between 1-32 characters.
3101000F	Invalid value specified for parameter {additionalProperties}. {>>} contains invalid characters. Valid characters are [alphanumeric, - and _].
31020012	createResourceDef failed for {type}:[host]. Remove self relation from resource definition [host].
31020012	createResourceDef failed for {type}:[host]. Resource type [host] already has relation with [vm].
31010002	Invalid JSON Request.
310000E9	License is not valid.
310000E9	Authorization needed.
310000E4	Dataset with {name}:[ds1] does not exist.
310000E0	Http method [PUT] is not supported for action [createResourceDef].
310000E2	Application encountered an internal error.
32040005	Resource definition with {type}:[host] does not exist.
3202000F	Invalid value specified for parameter {type}. Resource {type}:[!] contains invalid characters. Valid characters are [alphanumeric, - and _].

Response code	Message
32040008	Mandatory parameter {type} not specified.
3202000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.
32010002	Invalid JSON Request.
320000E9	License is not valid.
320000E9	Authorization needed.
320000E4	Dataset with {name}:[ds1] does not exist.
320000E0	Http method [PUT] is not supported for action [retrieveResourceDef]
320000E2	Application encountered an internal error.
33010008	Mandatory parameter {type} not specified.
3301000F	Invalid value specified for parameter {name}. It must be of type string.
3301000E	[add1] is not a valid key in parameter {{attributes}}
3301000F	Invalid value specified for parameter {attributes}. Atleast one correct key must be specified in {attributes}.
3301000F	Invalid value specified for parameter {attributes}. It must be of type JSON object.
3301000F	Invalid value specified for parameter {add}. It must be of type JSON array.
3301000F	Invalid value specified for parameter {completeList}. It must be of type JSON array.
3301000F	Invalid value specified for parameter {remove}. It must be of type JSON array.
3301000E	[add1] is not a valid key in parameter {{relations}}.
3301000F	Invalid value specified for parameter {relations}. Atleast one correct key must be specified in relations.

Response code	Message
3301000F	Invalid value specified for parameter {relations}. It must be of type JSON object.
3301000F	Invalid value specified for parameter {add}. It must be of type JSON array.
3301000F	Invalid value specified for parameter {completeList}. It must be of type JSON array.
3301000F	Invalid value specified for parameter {remove}. It must be of type JSON array.
3301000F	Invalid value specified for parameter {additionalProperties}. It must be of type JSON object.
3301000A	Nothing specified for update.
3302000F	Invalid value specified for parameter {type}. Resource {type}:[!] contains invalid characters. Valid characters are [alphanumeric, - and _].
3302000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.
3302000F	Invalid value specified for parameter {name}. {name} length must be between 2-32 characters.
3302000F	Invalid value specified for parameter {type}. Resource {type}:[] length must be between 1-32 characters.
3302000F	Invalid value specified for parameter {type}. Resource {type}:[!@] contains invalid characters. Valid characters are [alphanumeric, - and _].
33020010	Resource definition updation failed due to self relation. Remove self relation from resource definition [host].
33020010	Resource definition updation failed due to already existing relation. Resource type [host] already has relation with [vm].

Response code	Message
33020012	updateResourceDef failed for {type}:[h]. Number of additional properties can not be more than 128 in a resource definition.
33020012	updateResourceDef failed for {type}:[h]. Additional properties key cannot be one of type,configAttributes,displayName or directRelations
33020012	updateResourceDef failed for {type}:[h]. Invalid keys for additional properties. Keys : [!@]
33020012	updateResourceDef failed for {type}:[h]. Invalid keys for additional properties. Keys : [{}]
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]
33020012	updateResourceDef failed for {type}:[h]. Value(s) for following additional properties are invalid. :[{12}]
33050014	Cannot update read-only 'Resource definition' .
33050005	Resource definition with {type}:[host] does not exist .
33010002	Invalid JSON Request.
330000E9	License is not valid.
330000E9	Authorization needed.
330000E4	Dataset with {name}:[ds1] does not exist.
330000E0	Http method [GET] is not supported for action [updateResourceDef] .
330000E2	Application encountered an internal error.

Response code	Message
3402000F	Invalid value specified for parameter {type}. Resource {type}:[!] contains invalid characters. Valid characters are [alphanumeric, - and _].
34050008	Mandatory parameter {type} not specified.
3402000F	Invalid value specified for parameter {type}. Resource {type}: [IAmSuchABigStringWhichIsNotAllowedAsResourceTypeInMARSBM] length must be between 1-32 characters.
34050005	Resource definition with {type}:[vm] does not exist.
34050014	Cannot delete read-only 'Resource definition'.
34050011	Resource definition deletion failed due to dependency on one or more other resources.
34030007	No Resource definition is specified.
34010002	Invalid JSON Request.
340000E9	License is not valid.
340000E9	Authorization needed.
34050005	Dataset with {name}:[ds1] does not exist.
340500E0	Http method [PUT] is not supported for action [deleteResourceDef].
340000E2	Application encountered an internal error.
51010002	Invalid JSON Request.
51010008	Mandatory parameter {meta} not specified.
5101000F	Invalid value specified for parameter {meta}.
5101000F	Invalid value specified for parameter {ts}.
5101000F	Invalid value specified for parameter {ts} .
5101000F	Invalid value specified for parameter {dataSubsetId}.
51010005	DataSubset with {id}:[ds11] does not exist.

Response code	Message
5101000F	Invalid value specified for parameter {fullSubsetData}.
5101000F	Invalid value specified for parameter {fullSubsetFor}.
51010008	Mandatory parameter {resources} not specified.
5101000F	Invalid value specified for parameter {resources}
5101000F	Invalid value specified for parameter {fullSubsetData}. If {fullSubsetFor} is specified, then {fullSubsetData} must be true.
5101000F	Invalid value specified for parameter {fullSubsetFor}. If {fullSubsetFor} is specified, then it must not be empty.
51010006	Resource definition [vm] does not exist.
5101000F	Invalid value specified for parameter {signature}.
510000E9	License is not valid.
510000E9	Authorization needed.
510000E4	Dataset with {name}:[ds1] does not exist.
510000E0	Http method [GET] is not supported for action [createOrUpdateResourceData].
510000E2	Application encountered an internal error.
52010002	Invalid JSON Request.
52010009	At least one of {resSignatures}, {resType}, {attrId}, {attrValue} parameters must be specified.
5201000F	Invalid value specified for parameter {resSignatures}. It must be of type JSON list of string.
5201000F	Invalid value specified for parameter {resType}. It must be of type string.
5201000F	Invalid value specified for parameter {attrId}. It must be of type string.

Response code	Message
5201000F	Invalid value specified for parameter {attrValue}. It must be of type string.
5201000F	Invalid value specified for parameter {attrValue}. It must be of type string.
5201000F	Invalid value specified for parameter {snapshotTime}. It must be of type string.
5201000F	Invalid value specified for parameter {snapshotTime}. Parameter {snapshotTime} must be in yyyyMMdd_HHmmss format.
5201000F	Invalid value specified for parameter {snapshotTime}
5202000F	Invalid value specified for parameter {resSignatures}
5202000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.
5202000F	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.
5202000F	Invalid value specified for parameter {resType}. Resource {type}:[h\$%] contains invalid characters. Valid characters are [alphanumeric, - and _].
5202000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.
5202000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2-32 characters.
5202000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].
5202000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1-4000 characters.
52000005	Resource definition with {resType}:[vm] does not exist

Response code	Message
52000005	Attribute definition with {attrId}:{name} does not exist
5200000F	Invalid value specified for parameter {attrId}. Specified Attribute definition is not of type scalar
520000E9	License is not valid
520000E9	Authorization needed
520000E4	Dataset with {name}:{ds1} does not exist
520600E6	Requested resource not found
520000E4	Http method [PUT] is not supported for action [retrieveResourceData]
520000E2	Application encountered an internal error
54010002	Invalid JSON Request
54020009	At least one of {resType}, {attrId}, {attrValue}, {dataSubsetId} parameters must be specified
5402000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.
5402000F	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.
5402000F	Invalid value specified for parameter {resType}
5402000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.
5402000F	Invalid value specified for parameter {resType}. Resource {type}:[#] contains invalid characters. Valid characters are [alphanumeric, - and _].
5402000F	Invalid value specified for parameter {attrId}
5402000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2-32 characters.

Response code	Message
5402000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].
5402000F	Invalid value specified for parameter {attrValue}
5402000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1-4000 characters.
5402000F	Invalid value specified for parameter {dataSubsetId}
5402000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} length must be between 2-32 characters.
5402000F	Invalid value specified for parameter {dataSubsetId}. {dataSubsetId} contains invalid characters. Valid characters are [alphanumeric, - and _].
54000005	Resource definition with {resType}:[vm] does not exist
54000005	Attribute definition with {attrId}:[name] does not exist
5400000F	Invalid value specified for parameter {attrId}. Specified Attribute definition is not of type scalar
54050005	DataSubset with {dataSubsetId}:[ds1] does not exist
540000E9	License is not valid
540000E9	Authorization needed
540000E4	Dataset with {name}:[ds1] does not exist
54000005	Resource definition with {resType}:[h] does not exist
54000005	Attribute definition with {attrId}:[abc] does not exist
54050005	DataSubset with {dataSubsetId}:[ds] does not exist

Response code	Message
540000E0	Http method [GET] is not supported for action [deleteResourceData]
540000E2	Application encountered an internal error
56010002	Invalid JSON Request
56010009	At least one of {resSignatures}, {resType}, {attrId}, {attrValue} parameters must be specified
5601000F	Invalid value specified for parameter {attrValue}. It must be of type string.
5601000F	Invalid value specified for parameter {attrId}. It must be of type string.
5601000F	Invalid value specified for parameter {resType}. It must be of type string.
5601000F	Invalid value specified for parameter {resSignatures}. It must be of type JSON list of string.
5602000F	Invalid value specified for parameter {resSignatures}
5602000F	Invalid value specified for parameter {resType}. Resource {type}:[h\$%] contains invalid characters. Valid characters are [alphanumeric, - and _].
5602000F	Invalid value specified for parameter {resType}. Resource {type}:[] length must be between 1-32 characters.
5602000F	Invalid value specified for parameter {attrValue}. If {attrId} is specified, then {attrValue} must be specified.
5602000F	Invalid value specified for parameter {attrId}. If {attrValue} is specified, then {attrId} must be specified.
5602000F	Invalid value specified for parameter {attrId}. {attrId} length must be between 2-32 characters.
5602000F	Invalid value specified for parameter {attrId}. {attrId} contains invalid characters. Valid characters are [alphanumeric, - and _].

Response code	Message
5602000F	Invalid value specified for parameter {attrValue}. {attrValue} length must be between 1-4000 characters.
5601000F	Invalid value specified for parameter {endTime}
560000E9	License is not valid
560000E9	Authorization needed
56000005	Dataset with {name}:[ds1] does not exist
56060000	Not found
560000E0	Http method [PUT] is not supported for action [expireResourceData]
560000E2	Application encountered an internal error
62010008	Mandatory parameter {query} not specified
6201000F	Invalid value specified for parameter {query}. Query : vm[memory rx .*]
62010008	Mandatory parameter {startTime} not specified
6201000F	Invalid value specified for parameter {startTime}
6201000F	Invalid value specified for parameter {startTime}
62010008	Mandatory parameter {endTime} not specified
6201000F	Invalid value specified for parameter {endTime}
6201000F	Invalid value specified for parameter {endTime}
6202000F	Invalid value specified for parameter {startTime, endTime}. endTime must be after startTime.
62010002	Invalid JSON Request
620000E9	License is not valid
620000E9	Authorization needed

Response code	Message
620000E4	Dataset with {name}:[ds1] does not exist
620000E0	Http method [GET] is not supported for action [query]
620000E2	Application encountered an internal error
72020008	Mandatory parameter {requestId} not specified
720600E6	Response is not present in cache for the given requestId
720000E9	License is not valid
720000E9	Authorization needed
720600E6	Response is not present in cache for the given requestId
720600E2	Application encountered an internal error
82010002	Invalid JSON Request
8201000C	Parameter {query, startTime, endTime} not supportedParameter {query, startTime, endTime} not supported
8201000F	Invalid value specified for parameter {tw}
8201000F	Invalid value specified for parameter {tw}. Parameter [tw] must have only two elements.
8201000F	Invalid value specified for parameter {tw}. Parameter [tw] must contain start and end time in yyyyMMdd_HHmmss format.
8201000F	Invalid value specified for parameter {tw}. endTime must be after startTime.
8201000F	Invalid value specified for parameter {requestId}
8201000F	Invalid value specified for parameter {dataset}
8201000F	Invalid value specified for parameter {user}
8201000F	Invalid value specified for parameter {clientIP}
8201000F	Invalid value specified for parameter {action}

Response code	Message
8201000F	Invalid value specified for parameter {responseCode}
8201000F	Invalid value specified for parameter {serviceTime}
8201000F	Invalid value specified for parameter {serviceTime}. Parameter [serviceTime] must have only two element.
8201000F	Invalid value specified for parameter {serviceTime}. Maximum serviceTime must be greater than minimum.
8201000C	Parameter {userAgent} not supported
820000E2	Application encountered an internal error

Alert API response error codes and messages

The following table lists the alert API response codes that might be generated through the standard API. If an error response is not listed, examine the HTTP status codes to determine the best method for addressing the issue.

Response code	Message
A1020003	Mandatory parameter {name} not specified.
A1020003	Mandatory parameter {metric} not specified.
A1020008	Attribute definition with id:[unknownAttribute] does not exist.
A1020004	At least one of {resource_type, resource_mql} parameters must be specified.
A1020009	Resource definition with type: [unknownResourceType] does not exist.
A102000A	Resource definition with type [vm] doesn't contain [diskMaxTotalLatency] metric.
A1020003	Mandatory parameter {condition} not specified.
A1020003	Mandatory parameter {criteria} not specified.

Response code	Message
A1020006	Invalid {criteria} specified.
A1010003	Mandatory parameter {default_threshold} not specified.
A1020006	Invalid {condition} specified.
A1020006	Invalid {resource_mql} specified.
A1010007	Invalid {threshold} specified for resource [vm#vm1].
A104000C	Failed to add resource [vm#vm10] as resource does not exist.
A1010006	Invalid {monitoring_window} specified.
A1010005	Invalid {emailIds} specified [yyy, abc@@gmail.com].
A1010006	Invalid {damping} specified.
A1010005	Invalid {sysLogServer} specified [hostName].
A1020018	Resource [ldev#sig2] not present.
A1020019	Found unknown properties [resources, unknownKey].
A2020006	Invalid {from} specified.
A2020006	Invalid {to} specified.
A2020003	Mandatory parameter {from} not specified
A2020003	Mandatory parameter {to} not specified.
A2020006	Invalid {defStatus} specified.
A2020006	Invalid {severity} specified.
A2020006	Invalid {namePattern} specified.
A2020015	Timewindow not specified for the utcOffset.
A2020006	Invalid {utcOffset} specified.
A2020006	Invalid {from} specified.
A2020006	Invalid {to} specified.
A2020003	Mandatory parameter {from} not specified.

Response code	Message
A2020003	Mandatory parameter {to} not specified.
A2020006	Invalid {severity} specified.
A2020015	Timewindow not specified for the utcOffset.
A2020006	Invalid {utcOffset} specified.
A2000013	Application encountered internal error.
A2020019	Found unknown properties [resources, unknownKey].
A5010005	Invalid {emailIds} specified [yyy, abc@@gmail.com].
A5010006	Invalid {damping} specified.
A5010006	Invalid {alertId} specified.
A5010005	Invalid {sysLogServer} specified [host Name].
A5040010	Alert definition with id:[12] not found.
A5020019	Found unknown properties [resources, unknown Key].
A6040010	Alert definition with id:[12] not found.
A604000D	User [admin] is already unsubscribed from alert definition [1].
A6020019	Found unknown properties [resources, unknownKey].
A304000C	Failed to add resource [vm#vm16] as resource does not exist.
A304000B	Failed to remove resource [vm#vm5]. It does not exist in alert definition with id:[2].
A3040010	Alert definition with id:[12] not found.
A3020019	Found unknown properties [resources, unknownKey].

Appendix B: References

- IANA timezone database
<http://www.iana.org/time-zones>
- Oracle timezone updater tool
<http://www.oracle.com/technetwork/java/javase/tzupdater-readme-136440.html>
- General rounding rules: double rounded to nearest long
[http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#round\(double\)](http://docs.oracle.com/javase/7/docs/api/java/lang/Math.html#round(double))
- JSON specification – RFC 4627
[\[http://www.ietf.org/rfc/rfc4627.txt\]](http://www.ietf.org/rfc/rfc4627.txt)
- Simple date format
<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>
- Data type of data value is double so max value for input is Double.MAX_VALUE. After applying bandfactor this number is then converted in to long. So upper limit for normalized value would be Math.round(new Double(Double.MAX_VALUE).longValue).
- Data type of data value is double so there could be chance of precision loss after some ranges. For more details refer the following document
http://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA



HitachiVantara.com/contact