Using the System Comparison REST API

Red Hat Customer Content Services
Abstract

This document provides web services available to integrate System Comparison with external applications. It describes the specification of the System Comparison RESTful API, which is implemented as standard REST HTTP requests and responses of content type JSON. Providing Feedback: If you have a suggestion to improve this document or find an error, submit a Bugzilla report at http://bugzilla.redhat.com against Cloud Software Services (cloud.redhat.com) for the System Comparison component.
# Table of Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OVERVIEW</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>SCHEMA</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>BASIC AUTHENTICATION</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>SPECIFICATION</td>
<td>6</td>
</tr>
<tr>
<td>4.1</td>
<td>REST API ENTRY POINT</td>
<td>6</td>
</tr>
<tr>
<td>4.2</td>
<td>METHODS</td>
<td>6</td>
</tr>
<tr>
<td>4.3</td>
<td>DATA TYPES</td>
<td>6</td>
</tr>
<tr>
<td>4.4</td>
<td>HTTP STATUS CODE</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>COMPARISON_REPORT API</td>
<td>7</td>
</tr>
<tr>
<td>5.1</td>
<td>EXPORTING COMPARISON_REPORT API</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>BASELINES API</td>
<td>9</td>
</tr>
<tr>
<td>6.1</td>
<td>CREATING A NEW BASELINE</td>
<td>9</td>
</tr>
<tr>
<td>6.2</td>
<td>COPYING AN EXISTING SYSTEM TO CREATE A NEW BASELINE</td>
<td>10</td>
</tr>
<tr>
<td>6.3</td>
<td>COPYING AN EXISTING BASELINE TO CREATE A NEW BASELINE</td>
<td>11</td>
</tr>
<tr>
<td>6.4</td>
<td>SORTING BASELINES</td>
<td>11</td>
</tr>
<tr>
<td>6.5</td>
<td>EXPORTING BASELINES</td>
<td>11</td>
</tr>
<tr>
<td>6.6</td>
<td>DELETING A BASELINE</td>
<td>11</td>
</tr>
</tbody>
</table>
CHAPTER 1. OVERVIEW

System Comparison enables you to compare system configuration of one system to other systems and baselines registered in your cloud management services inventory. It allows you to query system configurations using a REST API, create and manage baselines, and returns fact values as well as comparison states. You can also generate CSV output of the systems and baselines you are comparing.

NOTE

- For additional Cloud Management Services for Red Hat Enterprise Linux API documentation, see https://cloud.redhat.com/docs/api-docs
- For the latest OpenAPI Specification document, see OpenAPI Specification.
CHAPTER 2. SCHEMA

The API access is over HTTPS, and accessed from the root URL of https://cloud.redhat.com/api/drift. All data is sent and received as JSON.
CHAPTER 3. BASIC AUTHENTICATION

The user and password credentials are passed in with each HTTP request.

Request

$ curl --user username:password -i https://cloud.redhat.com/api/drift/v1/comparison_report?
   system_ids[]=<UUID>

or

$ curl --user username:password -i https://cloud.redhat.com/api/system-baseline/v1/baselines
CHAPTER 4. SPECIFICATION

4.1. REST API ENTRY POINT

The REST API is available via the /api URL prefix. It is accessed on the server as follows:

/api/drift/v1

4.2. METHODS

The HTTP methods currently supported for API requests are GET and POST.

4.3. DATA TYPES

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer</td>
<td>Integer value</td>
</tr>
<tr>
<td>String</td>
<td>JSON string</td>
</tr>
<tr>
<td>Array</td>
<td>The items keyword is required in arrays. The value of items is a schema that describes the type and format of array items. Arrays can be nested.</td>
</tr>
<tr>
<td>Boolean</td>
<td>Value resulting in True or False</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Timestamp in ISO8601 format</td>
</tr>
</tbody>
</table>

4.4. HTTP STATUS CODE

<table>
<thead>
<tr>
<th>Code</th>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>Success. A comparison report for the given system IDs.</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
<td>The request could not be understood by the server due to incorrect syntax.</td>
</tr>
<tr>
<td>500</td>
<td>Internal Server Error</td>
<td>The server encountered an unexpected condition which prevented it from fulfilling the request.</td>
</tr>
</tbody>
</table>
The main purpose of the comparison report is to fetch system profiles for the systems requested, pull their facts from the inventory service, and then transform those facts into a comparison report. If you are simply trying to fetch the information for a system and do not require comparison information, you can use the inventory API directly.

**NOTE**

There is currently a 45 system limit when comparing systems via either the web user interface or API. Requests to compare more than 45 systems at a time may result in a 400 error. Also, requests may also hit the timeout limit set by the API infrastructure.

**Example using the GET method:**

The following is an example of using the comparison_report API to interact with registered systems in inventory to compare system configurations using the GET method. This will fetch the most current system state using data from inventory service for given system IDs and generate a comparison report.

**Request:**

```plaintext
GET /api/drift/v1/comparison_report?system_ids=<UUID>&system_ids=<UUID>
```

```json
openapi: 3.0.1
info:
  version: "1.0"
  title: Drift Backend Service
  description: Service that returns differences between systems.
  license:
    name: Apache 2.0
    url: https://www.apache.org/licenses/LICENSE-2.0.html

servers:
  - url: /api/drift/v1

paths:
  /comparison_report:
    parameters:
      - $ref: '#/components/parameters/rhIdentityHeader'
    get:
      summary: fetch comparison report
      description: "Fetch a comparison report for given system IDs. This will only fetch the most current system state using data from inventory service."
      operationId: drift.views.v1.comparison_report
      parameters:
        - name: system_ids[]
          in: query
          schema:
            type: array
            items:
              type: string
              format: uuid
          required: true
          description: list of system IDs to generate report for
```
Responses:

'200':
  description: a comparison report for the given system IDs
  content:
    application/json:
      schema:
        $ref: "#components/schemas/ComparisonReport"
      examples:
        jsonObject:
          summary: A sample comparison report
          externalValue: "https://git.io/fhQ00"

'400':
  $ref: '#/components/responses/BadRequest'

'500':
  $ref: '#/components/responses/InternalServerError'

Command-line example with curl using the POST method:

The following is an example of using the comparison_report API to interact with registered systems in inventory to compare system configurations by providing JSON file as input using the POST method.

NOTE
There is no system limit when comparing systems using the POST method via API.

Request:

$ curl -k -u username:password -X POST --header "Content-Type: application/json" https://cloud.redhat.com/api/drift/v1/comparison_report -d @/directory/example-uuid-file.json

Example .json file content:

{"system_ids": ["UUID1", "UUID2", "UUID3"]}

5.1. EXPORTING COMPARISON_REPORT API

You can use the comparison_report API to export the entire comparison report output for the given systems and baselines IDs without any filters. The output includes all facts including categories (parent facts) and sub facts, and their values in a comma-separated values (CSV) format.

Request:

$ curl -X GET 'https://username:password@cloud.redhat.com/api/drift/v1/comparison_report?baseline_ids[]=UUID1&system_ids[]=UUID2&system_ids[]=UUID3' -H 'accept: text/csv'
CHAPTER 6. BASELINES API

Baselines are set of facts that can be created and managed using the API. You can compare system configurations to one or more baselines to identify discrepancies in your environment, export the output in CSV format, and perform drift analysis.

The following command-line examples demonstrate how to use the baselines API to interact with registered systems in the inventory to create and manage baselines by providing JSON file as input.

6.1. CREATING A NEW BASELINE

An example of a create.json file follows:

```json
{
    "baseline_facts": [
        {
            "name": "fact name 1",
            "value": "fact value 1"
        },
        {
            "name": "fact name 2",
            "value": "fact value 2"
        },
        {
            "name": "category 1",
            "values": [
                {
                    "name": "sub fact name 1",
                    "value": "sub fact value 1"
                }
            ]
        }
    ],
    "display_name": "demo baseline"
}
```

Request:

```bash
$ curl -u username:password
https://cloud.redhat.com/api/system-baseline/v1/baselines -X
POST -d @create.json -H "content-type: application/json"
```

Response:

```json
{
    "account": "[account number]",
    "baseline_facts": [
        {
            "name": "category 1",
            "values": [
                {
                    "name": "sub fact name 1",
                    "value": "sub fact value 1"
                }
            ]
        }
    ]
}
```
6.2. COPYING AN EXISTING SYSTEM TO CREATE A NEW BASELINE

You can copy an existing system to create a new baseline via API.

An example `create-from-inv.json` file follows:

```json
{"inventory_uuid": "<UUID>",
 "display_name": "from-inv1"}
```

Request:

```bash
$ curl -k -X POST -d @create-from-inv.json
https://username:password@cloud.redhat.com/api/system-baseline/v1/baselines
-H 'content-type: application/json'
```

Response:

Note that some of the facts have been removed from the response to improve readability.

```json
{
    
    "Name": "number_of_sockets",
    "Value": "1"
},
{
    "Name": "cores_per_socket",
    "Value": "1"
},
{
    "Name": "number_of_cpus",
    "Value": "None"
},
{
    "Name": "system_memory",
    "Value": "488.35 MiB"
}```
6.3. COPYING AN EXISTING BASELINE TO CREATE A NEW BASELINE

You can copy an existing baseline to create a new baseline via API using the GET and POST methods.

1. First, obtain the list of existing baseline IDs via GET:

   ```
   $ curl -X GET https://username:password@cloud.redhat.com/api/system-baseline/v1/baselines -H 'content-type: application/json'
   ```

2. Select the baseline ID you want to copy from to create a new baseline.

3. Create a new baseline using the selected baseline ID via POST. You must provide a new display name, otherwise it will result in a 400 error.

   ```
   $ curl -X POST https://username:password@cloud.redhat.com/api/system-baseline/v1/baselines/<ID>?display_name=copied-from-baseline-name
   ```

6.4. SORTING BASELINES

You can sort baselines via API using the GET method by appending order_by and order_how parameters.

- order_by parameter accepts the following values: display_name, created_on
- order_how parameter accepts the following values for sorting the results: ASC (ascending) and DESC (descending)

Note that by default results are sorted by alphabetical order on display name.

Request:

```
$ curl -X GET https://username:password@cloud.redhat.com/api/system-baseline/v1/baselines?order_by=created_on&order_how=DESC
``` 

6.5. EXPORTING BASELINES

You can export baselines via API using the GET method.

```
$ curl -X GET https://username:password@cloud.redhat.com/api/system-baseline/v1/baselines -H 'accept: text/csv'
``` 

6.6. DELETING A BASELINE
Use the following curl command to delete a baseline via API.

```
```