Red Hat Insights 2023

System Comparison API Documentation

Using the REST API for the Insights for Red Hat Enterprise Linux drift service
Using the REST API for the Insights for Red Hat Enterprise Linux drift service

Red Hat Customer Content Services
Abstract

This document provides web services available to integrate the System Comparison/drift service 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. Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
Table of Contents

CHAPTER 1. USING APIs FOR QUERIES ................................................................. 3
  1.1. SCHEMA .................................................................................. 3
  1.2. BASIC AUTHENTICATION ....................................................... 3
  1.3. SPECIFICATION .................................................................. 3
        1.3.1. REST API entry point .................................................. 3
        1.3.2. HTTP methods ............................................................ 3
        1.3.3. Data types ................................................................. 4
        1.3.4. HTTP status codes .................................................. 4

CHAPTER 2. COMPARISON REPORT API ......................................................... 5
  2.1. SYSTEMS–ONLY REPORT QUERIES ........................................... 6
  2.2. SYSTEMS AND BASELINE REPORT QUERIES .......................... 6
  2.3. SYSTEM AND HISTORICAL SYSTEM PROFILE REPORT QUERIES .................................................................................. 7
  2.4. SYSTEM, BASELINE, AND HISTORICAL SYSTEM PROFILE REPORT QUERIES ....................................................... 7
  2.5. EXPORTING THE COMPARISON REPORT API .......................... 7
  2.6. REFERENCE QUERIES .............................................................. 7

CHAPTER 3. BASELINES API .................................................................. 9
  3.1. CREATING A NEW BASELINE FROM SCRATCH ...................... 9
  3.2. COPYING AN EXISTING SYSTEM TO CREATE A NEW BASELINE .................................................................................. 10
  3.3. COPYING AN EXISTING BASELINE TO CREATE A NEW BASELINE .................................................................................. 11
  3.4. COPYING A HISTORICAL SYSTEM PROFILE TO CREATE A NEW BASELINE .................................................................................. 11
  3.5. SORTING BASELINES .............................................................. 12
  3.6. EDITING A BASELINE ............................................................ 12
  3.7. EXPORTING BASELINES ........................................................ 12
  3.8. DELETING A BASELINE ........................................................ 13

CHAPTER 4. HISTORICAL SYSTEM PROFILES API .................................... 14
  4.1. SYSTEMS METHOD ............................................................... 14
  4.2. PROFILES METHOD .............................................................. 15

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .......................... 17
CHAPTER 1. USING APIS FOR QUERIES

The drift service enables you to compare the system configuration of one system to that of other systems and baselines registered in your Insights for Red Hat Enterprise Linux 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 Insights for Red Hat Enterprise Linux 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.

1.1. SCHEMA

The API access is over HTTPS, and accessed from the root URL of https://access.redhat.com/documentation/en-us/red_hat_insights/2023/html/system_comparison_api_documentation/ [System Comparison API Documentation]. All data is sent and received as JSON. The JSON files include:

- drift-openapi.json
- historical-system-profiles-openapi.json
- system-baseline-openapi.json

1.2. 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

1.3. SPECIFICATION

1.3.1. REST API entry point

The REST API is available by using the /api URL prefix. Accessed it on the server as follows:

"/api/drift/v1"

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

### 1.3.3. 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 <strong>items</strong> 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>

### 1.3.4. HTTP status codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Text</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
<td>Success.</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>
CHAPTER 2. COMPARISON REPORT API

The 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 simply need to fetch the information for a system and do not require comparison information, you can use the inventory API directly.

**NOTE**

Currently, there is 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. Such requests may hit the timeout limit set by the API infrastructure.

**Example**

The following example uses the comparison_report API to interact with registered systems in inventory to compare system configurations using the GET method. This request fetches the most current system state, using data from the inventory service for the given system IDs, and generates a comparison report.

**Request:**

```
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'

Example

The following example uses the comparison_report API to interact with registered systems in inventory
to compare system configurations by providing a 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

An example of JSON file content follows.

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

2.1. SYSTEMS-ONLY REPORT QUERIES

You can query systems-only reports by using API:

GET
api/drift/v1/comparison_report?system_ids[]=55e47a1d-05d7-4abb-963-a-ba513a0a57ad&system_ids[]=....

2.2. SYSTEMS AND BASELINE REPORT QUERIES

You can query systems and baseline reports by using API:
2.3. SYSTEM AND HISTORICAL SYSTEM PROFILE REPORT QUERIES

You can query systems and historical system profile reports by using API:

```text
GET
api/drift/v1/comparison_report?system_ids[]=55e47a1d-05d7-4abb-963
a-ba513a0a57ad&baseline_ids[]=79032c7b-8284-44d0-b820-ebbb6d25c5e2
```

2.4. SYSTEM, BASELINE, AND HISTORICAL SYSTEM PROFILE REPORT QUERIES

You can query systems, baselines, and historical system profile reports by using API:

```text
GET
api/drift/v1/comparison_report?system_ids[]=55e47a1d-05d7-4abb-963
a-ba513a0a57ad&baseline_ids[]=79032c7b-8284-44d0-b820-ebbb6d25c5e2
&historical_system_profile_ids[]=5d7dd2e9-18e5-412e-8f25-efa8a96a4289
```

2.5. EXPORTING THE COMPARISON REPORT API

You can use the comparison report API to export the entire comparison report output for the given systems, baselines, and historical system profile 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'
```

2.6. REFERENCE QUERIES

You can set a reference in the query when using the API:

```text
GET
api/drift/v1/comparison_report?reference_id=UUID1&system_ids[]=55e47a1d-05d7-4abb-963
a-ba513a0a57ad&baseline_ids[]=79032c7b-8284-44d0-b820-ebbb6d25c5e2
&historical_system_profile_ids[]=5d7dd2e9-18e5-412e-8f25-efa8a96a4289
```
NOTE

The `reference_id` parameter is optional, and you can use only one reference ID in a query. You can set it to any used parameter in the comparison query: system ID, baseline ID, or historical system profile ID. When you use the reference ID, the Comparison service compares all systems, baselines, and historical system profiles in the list against it.
CHAPTER 3. BASELINES API

Baselines are a set of facts that can be created and managed by using the API.

The following command-line examples demonstrate how to use the baselines API to create and manage baselines. You can create baselines from scratch, or by copying from a system inventory ID, from a baseline ID, and from an historical system profile ID.

3.1. CREATING A NEW BASELINE FROM SCRATCH

You can create a new baseline from scratch.

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"
}
```

The following examples show a CURL command using POST with the JSON data method for creating a new baseline from scratch.

Request:

```
$ 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": [
```
3.2. COPYING AN EXISTING SYSTEM TO CREATE A NEW BASELINE

You can copy an existing system to create a new baseline by using the API.

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

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

**Request:**

```
$ 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.
3.3. COPYING AN EXISTING BASELINE TO CREATE A NEW BASELINE

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

1. 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 the POST method. 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
   ```

3.4. COPYING A HISTORICAL SYSTEM PROFILE TO CREATE A NEW BASELINE

You can copy a historical system profile to create a new baseline via API.

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

```json
{"hsp_uuid": "<UUID>",
;display_name": "from-hsp1"}
```

Request:

```bash
$ curl -k -X POST -d @create-from-hsp.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
{
}
3.5. SORTING BASELINES

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

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

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
```

3.6. EDITING A BASELINE

You can edit a baseline by using the API, by using the `curl` command.

```
$ curl -u username:password https://cloud.redhat.com/api/system-baseline/v1$/baselines/[uuid]
```

3.7. EXPORTING BASELINES

You can export baselines as a CSV file by using the API GET method.
3.8. DELETING A BASELINE

You can delete a baseline by using the API.

Use the following curl command to delete a baseline via the API.

```bash
```
CHAPTER 4. HISTORICAL SYSTEM PROFILES API

Historical system profiles capture points in time for system profiles in Insights for Red Hat Enterprise Linux inventory. The profiles return lists of system profiles for a particular system.

The historical system profile has two main methods:

- Systems method
- Profiles method

4.1. SYSTEMS METHOD

The systems method returns a list of all historical system profiles for the system you specify in the uuid parameter.

Use the following curl command to get all historical system profiles for a system uuid.

```
$ curl -u username:password https://cloud.redhat.com/api/historical-system-profiles/v1/system/[ID]
```

For each profile, the service returns:

- **captured_date**— the date indicating when the system profile was captured
- an **id** — which is the profile uuid.
- **system_id**— the system uuid, which is the same as what was specified in the request.

The following is a partial sample of the Systems method output you can expect to see:

```
{
    "data": [
    {
    "profiles": [

    {
    "captured_date": "2020-10-28T06:23:25+00:00",
    "id": "35054dcf-da10-489e-b383-8580511c8c10",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
    },

    {
    "captured_date": "2020-10-27T18:43:28+00:00",
    "id": "472ad292-e2b1-40ac-b514-502df11df765",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
    },

    {
    "captured_date": "2020-10-26T07:34:17+00:00",
    "id": "89235117-e9f5-4a2c-86d4-c3e908a392a7",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
    }
    ]
    }
```
4.2. PROFILES METHOD

The profiles endpoint returns system profile information for each profile ID you specify in the `uuid` parameter. The API takes one or more comma-separated historical system profile ids and returns system profile information for each id.

Use the following curl command to get historical system profiles passed as comma-separated parameters:

```
$ curl -u username:password https://cloud.redhat.com/api/historical-system-profiles/v1/profiles/[ID1],[ID2]
```

The following is a partial sample of the Profiles method output you can expect to see:

```json
{
  "data": [
    {
      "account": "ACCOUNTID",
      "captured_date": "2020-10-27T18:43:28+00:00",
      "created": "2020-10-27T18:43:48.340185+00:00",
      "display_name": "dsmith-rhel82kvm",
      "id": "472ad292-e2b1-40ac-b514-502df11df765",
      "inventory_id": "563f782e-6266-41f0-a761-0e0ab29463b"
    },
    {
      "captured_date": "2020-10-25T06:40:18+00:00",
      "id": "b66e8bb8-916c-409b-96d8-93119944e71d",
      "system_id": "563f782e-6266-41f0-a761-0e0ab29463b"
    },
    {
      "captured_date": "2020-10-24T06:23:19+00:00",
      "id": "b7dadf95-90ec-4289-b50a-6c8d19124ccf",
      "system_id": "563f782e-6266-41f0-a761-0e0ab29463b"
    },
    {
      "captured_date": "2020-10-23T06:34:14+00:00",
      "id": "3c9ce38d-22da-4d06-876d-35133eb5959e",
      "system_id": "563f782e-6266-41f0-a761-0e0ab29463b"
    },
    {
      "captured_date": "2020-10-22T05:59:09+00:00",
      "id": "18cd803d-3f1f-4ce7-9bbf-6a6585dc9c03",
      "system_id": "563f782e-6266-41f0-a761-0e0ab29463b"
    }
  ]
}
```
"system_profile": {
"arch": "x86_64",
"bios_release_date": "04/01/2014",
"bios_vendor": "SeaBIOS",
"bios_version": "1.12.0-2.fc30",
"captured_date": "2020-10-27T18:43:28+00:00",
"cores_per_socket": 2,

"cpu_flags": [],
"display_name": "dsmith-rhel82kvm",

"dnf_modules": [],

"enabled_services": [],
"fqdn": "dsmith-rhel82kvm",
"id": "472ad292-e2b1-40ac-b514-502df11df765",
"infrastructure_type": "virtual",
"infrastructure_vendor": "kvm",
"insights_client_version": "3.0.12-0",
"insights_egg_version": "3.0.162-1",

"installed_packages": [],

"installed_products": [],

"installed_services": [],

"kernel_modules": [],
"last_boot_time": "2020-05-01T02:11:28",

"network_interfaces": [],
"number_of_cpus": 8,
"number_of_sockets": 4,
"os_kernel_version": "4.18.0",
"os_release": "8.2",

"running_processes": [],
"satellite_managed": false,
"selinux_config_file": "enforcing",
"selinux_current_mode": "enforcing",
"system_memory_bytes": 1914204160,

"tags": [],
"tuned_profile": "virtual-guest",

"yum_repos": []
}
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, highlight text in a document and add comments.

Prerequisites

- You are logged in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, the document is in the Multi-page HTML viewing format.

Procedure

To provide your feedback, perform the following steps:

1. Click the Feedback button in the top-right corner of the document to see existing feedback.

   **NOTE**
   
   The feedback feature is enabled only in the Multi-page HTML format.

2. Highlight the section of the document where you want to provide feedback.

3. Click the Add Feedback pop-up that appears near the highlighted text.
   A text box appears in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.
   A documentation issue is created.

5. To view the issue, click the issue link in the feedback view.