



Red Hat Insights 2023

系统比较 API 文档

将 REST API 用于 Red Hat Enterprise Linux 偏移服务

Red Hat Insights 2023 系统比较 API 文档

将 REST API 用于 Red Hat Enterprise Linux 偏移服务

Red Hat Customer Content Services

法律通告

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档提供了 Web 服务，可用于将系统过期/drift 服务与外部应用集成。它描述了系统相似 RESTful API 的规范，该 API 作为标准的 REST HTTP 请求和内容类型 JSON 响应实施。红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 信息。

目录

第 1 章 使用 API 进行查询	3
1.1. 模式	3
1.2. 基本身份验证	3
1.3. 规格	3
第 2 章 比较报告 API	5
2.1. 仅系统查询报告查询	6
2.2. 系统和基线报告查询	6
2.3. 系统和系统配置文件报告查询	6
2.4. 系统、基线和系统配置文件报告查询	7
2.5. 导出相关的报告 API	7
2.6. 参考查询	7
第 3 章 BASELINES API	8
3.1. 从 SCRATCH 创建新基线	8
3.2. 复制现有系统以创建新基线	9
3.3. 将现有基线复制到创建新基线	10
3.4. 复制系统配置文件以创建新基线	10
3.5. 排序基本行	11
3.6. 编辑基本行	11
3.7. 导出基本行	11
3.8. 删除基本行	11
第 4 章 历史系统配置集 API	13
4.1. 系统方法	13
4.2. 配置集方法	14
对红帽文档提供反馈	16

第 1 章 使用 API 进行查询

偏移服务允许您将一个系统的系统配置与在 Insights for Red Hat Enterprise Linux 清单中注册的其他系统和基准进行比较。它允许您使用 REST API 查询系统配置，创建和管理基准，并返回事实值以及比较状态。您还可以生成您要比较的系统和基准的 CSV 输出。



注意

- 有关 Red Hat Enterprise Linux API 的 Red Hat Enterprise Linux 服务的其他 Insights 文档，请参阅 <https://cloud.redhat.com/docs/api-docs>。
- 有关最新的 OpenAPI 规格文档，请参阅 [OpenAPI 规格](#)。

1.1. 模式

API 访问通过 HTTPS，并从 https://access.redhat.com/documentation/zh-cn/red_hat_insights/2023/html/system_comparison_api_documentation/ 的 root URL 访问 [[系统比较 API 文档](#)]。所有数据都以 JSON 的形式发送和接收。JSON 文件包括：

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

1.2. 基本身份验证

用户和密码凭据通过每个 HTTP 请求传递。

Request (请求)

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

或

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

1.3. 规格

1.3.1. REST API 条目点

REST API 可通过 /api URL 前缀获得。它在服务器上被访问，如下所示：

```
"/api/drift/v1"
```

1.3.2. HTTP 方法

目前支持 API 请求的 HTTP 方法是 GET 和 POST。

1.3.3. 数据类型

数据类型	描述
整数	整数值
字符串	JSON 字符串
数组	数组中需要 items 关键字。items 的值是描述数组项目的类型和格式的 schema。数组可以嵌套。
布尔值	导致 True 或 False 的值
Timestamp	ISO8601 格式的时间戳

1.3.4. HTTP 状态代码

代码	文本	描述
200	确定	成功。
400	错误请求	由于语法不正确，服务器无法理解请求。
500	内部服务器错误	服务器遇到意外条件，阻止它满足请求。

第 2 章 比较报告 API

比较报告的目的是为请求的系统获取系统配置文件，从库存服务中提取其事实，然后将这些事实转换为比较报告。如果您只是需要获取系统的信息且不需要比较信息，您可以直接使用清单 API。



注意

目前，通过 Web 用户界面或 API 比较系统时，有 45 个系统限制。在一个时间点上比较超过 45 个系统的请求可能会导致 400 错误。此类请求可能会达到 API 基础架构设置的超时限制。

示例

以下示例使用 compare_report API 与清单中注册的系统交互，以使用 GET 方法比较系统配置。此请求获取最新的系统状态，使用来自给定系统 ID 的清单服务中的数据，并生成比较报告。

Request:

GET

api/drift/v1/comparison_report?system_ids[]=<UUID>&system_ids[]=<UUID>

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

响应：

```

'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'

```

示例

以下示例使用 `compare_report` API 与清单中注册的系统交互，通过使用 POST 方法提供 JSON 文件作为输入来比较系统配置。



注意

在通过 API 使用 POST 方法比较系统时，没有系统限制。

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
```

以下是 JSON 文件内容的示例。

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

2.1. 仅系统查询报告查询

您只能使用 API 查询系统报告：

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

2.2. 系统和基线报告查询

您可以使用 API 查询系统和基准报告：

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

2.3. 系统和系统配置文件报告查询

您可以使用 API 查询系统和历史系统配置集报告：

```
GET
api/drift/v1/comparison_report?system_ids[]=55e47a1d-05d7-4abb-963
a-ba513a0a57ad&historical_system_profile_ids[]=5d7dd2e9-18e5-412e-8f25-efa8a96a4289
```

2.4. 系统、基线和系统配置文件报告查询

您可以使用 API 查询系统、基准和历史系统配置集报告：

```
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-efa8a96a4
289
```

2.5. 导出相关的报告 API

您可以使用比较报告 API 在没有过滤器的情况下导出给定系统、基准和历史系统配置文件 ID 的整个比较报告输出。输出中包括包括类别（父事实）和子事实在内的所有事实，以及它们的值以逗号分隔的值 (CSV) 格式。

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. 参考查询

在使用 API 时，您可以在查询中设置引用：

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



注意

reference_id 参数是可选的，您只能在查询中使用一个引用 ID。您可以在比较查询中将它设置为任何使用的参数：系统 ID、基准 ID 或历史系统配置文件 ID。当您使用引用 ID 时，相关的服务会将列表中的所有系统、基准和历史系统配置集与其进行比较。

第 3 章 BASELINES API

基准是一组事实，可使用 API 创建和管理。

以下命令行示例演示了如何使用基准 API 创建和管理基准。您可以从头开始创建基准，或者从系统清单 ID、从基准 ID 以及历史系统配置文件 ID 复制基准。

3.1. 从 SCRATCH 创建新基线

您可以从头开始创建新基准。

一个 `create.json` 文件示例如下：

```
{
  "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"
}
```

以下示例演示了使用 POST 和 JSON 数据方法从头开始创建新基准的 CURL 命令。

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:

```
{
  "account": "[account number]",
  "baseline_facts": [
    {
      "name": "category 1",
      "values": [
        {
          "name": "sub fact name 1",
```

```

        "value": " sub fact value 1"
      }
    ]
  },
  { :context: {parent-context}
    "name": "fact name 1",
    "value": "fact value 1"
  },
  {
    "name": "fact name 2"
    "value": "fact value 2"
  }
],
"created": "2020-05-18T20:25:36.328741Z",
"display_name": "demo baseline",
"fact_count": 3,
"id": "9565f205-5816-43b4-953e-a29fed8b40ec",
"updated": "2020-05-18T20:25:36.328745Z"
}

```

3.2. 复制现有系统以创建新基线

您可以复制现有系统以通过 API 创建新基准。

create-from-inv.json 文件示例如下：

```

{"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:

请注意，一些事实已从响应中删除，以提高可读性。

```

{
  [
    {
      "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"
  }
],
"Created": "2019-08-27T15:34:38.504298Z",
"Display_name": "from-inv1",
"Fact_count": 20,
"Id": "UUID",
"Updated": "2019-08-27T15:34:38.504298Z"
}

```

3.3. 将现有基线复制到创建新基线

您可以使用 GET 和 POST 方法复制现有基线，以通过 API 创建新基线。

1. 通过 GET 获取现有基线 ID 列表：

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

2. 选择您要复制的基线 ID 来创建新基线。
3. 通过 POST 方法使用所选基线 ID 来创建新基线。您必须提供一个新的显示名称，否则会导致 400 错误。

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

3.4. 复制系统配置文件以创建新基线

您可以复制历史系统配置集，以通过 API 创建新基线。

create-from-hsp.json 文件示例如下：

```

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

```

Request:

```
$ 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:

请注意，一些事实已从响应中删除，以提高可读性。

```

{
[
{
  "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"
}
],
"Created": "2019-08-27T15:34:38.504298Z",
"Display_name": "from-inv1",
"Fact_count": 20,
"HSP_uuid": "UUID",
"Updated": "2019-08-27T15:34:38.504298Z"
}
```

3.5. 排序基本行

您可以使用 GET 方法通过 API 对基准进行排序，方法是附加 **order_by** 和 **order_how** 参数。

- **order_by** 参数接受以下值：**display_name**、**create_on** 和 **更新**
- **order_how** 参数接受下列值来对结果进行排序：**ASC**（升序）和 **DESC**（降序）

默认情况下，结果按照显示名称中的字母顺序排序。

Request:

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

3.6. 编辑基本行

您可以使用 **curl** 命令通过 API 编辑基准。

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

3.7. 导出基本行

您可以使用 GET 方法通过 API 将基准导出为 CSV 文件。

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

3.8. 删除基本行

您可以使用 API 删除基准。

使用以下 curl 命令，通过 API 删除基准。

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


第 4 章 历史系统配置集 API

Red Hat Enterprise Linux 清单中 Insights 中系统配置集的历史系统配置集捕获点。配置集返回特定系统的系统配置集列表。

历史系统配置集有两个主要方法：

- 系统方法
- 配置集方法

4.1. 系统方法

system 方法返回您在 **uuid** 参数中指定的系统的所有历史系统配置集列表。

使用以下 **curl** 命令获取系统 **uuid** 的所有历史系统配置集。

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

对于每个配置集，服务会返回：

- **captured_date**-- 表示系统配置集何时捕获的日期
- **id** IANA-prompt，它是配置文件 **uuid**。
- **system_id**-- 系统 **uuid**，它与请求中指定的内容相同。

以下是您可以看到的系统方法输出的部分示例：

```
{
  "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": "2020-10-27T06:14:13+00:00",
          "id": "89235117-e9f5-4a2c-86d4-c3e908a392a7",
          "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
        },
        {
          "captured_date": "2020-10-26T07:34:17+00:00",
          "id": "6652ba31-f5e5-45c6-ae59-8f8d51181358",
```

```

    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
  },
  {
    "captured_date": "2020-10-25T06:40:18+00:00",
    "id": "b66e8bb8-916c-409b-96d8-93119944e71d",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
  },
  {
    "captured_date": "2020-10-24T06:23:19+00:00",
:context: {parent-context}
    "id": "b7dadf95-90ec-4289-b50a-6c8d19124ccf",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
  },
  {
    "captured_date": "2020-10-23T06:34:14+00:00",
    "id": "3c9ce38d-22da-4d06-876d-35133eb5959e",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
  },
  {
    "captured_date": "2020-10-22T05:59:09+00:00",
    "id": "18cd803d-3f1f-4ce7-9bbf-6a6585dc9c03",
    "system_id": "563f782e-6266-41f0-a761-0e0eab29463b"
  }
]
}

```

4.2. 配置集方法

配置集端点返回您在 **uuid** 参数中指定的每个配置集 ID 的系统配置集信息。API 使用一个或多个以逗号分隔的历史系统配置集 id，并返回每个 id 的系统配置集信息。

使用以下 curl 命令获取以逗号分隔的参数传递的历史系统配置集：

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

以下是您可以看到的 Profiles 方法输出的部分示例：

```

{
  "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-a761-0e0eab29463b",
      "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": [ ],
"context": {parent-context}      "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": [ ]
}
}
]
}
```

对红帽文档提供反馈

我们感谢您对我们文档的反馈。要提供反馈，请突出显示文档中的文本并添加注释。

先决条件

- 已登陆到红帽客户门户网站。
- 在红帽客户门户网站中，文档采用 **Multi-page HTML** 查看格式。

流程

要提供反馈，请执行以下步骤：

1. 点击 **文档** 右上角的反馈按钮查看现有的反馈。



注意

反馈功能仅在多页 HTML 格式中启用。

2. 高亮标记您要提供反馈的文档中的部分。
3. 点在高亮文本旁弹出的 **Add Feedback**。
文本框会出现在页面右侧的反馈部分中。
4. 在文本框中输入您的反馈，然后点 **Submit**。
已创建一个文档问题。
5. 要查看问题，请点击反馈视图中的问题链接。