



Red Hat Subscription Management 1

Red Hat Subscription Management Integration Guide

Guidance for integrating your application with Red Hat Subscription Management

Red Hat Subscription Management 1 Red Hat Subscription Management Integration Guide

Guidance for integrating your application with Red Hat Subscription Management

Red Hat Subscription Management Documentation Team

rhsm-docs@redhat.com

Legal Notice

Copyright © 2017 Red Hat, Inc..

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, 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 Software Collections 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.

Abstract

This document provides information about using APIs exposed by Red Hat Subscription Management in order to query and update customer cases.

Table of Contents

1. INTRODUCTION	2
2. CLIENTS	2
3. API COMMANDS IN RED HAT SUBSCRIPTION MANAGEMENT	2

1. INTRODUCTION

It is now possible to integrate with the Customer Portal at <http://access.redhat.com> using an XML-based REST API ^[1]. This document outlines common use cases including examples in common clients and frameworks. The official API documentation can be found on the Customer Portal at https://access.redhat.com/site/documentation/Red_Hat_Customer_Portal/ and includes both XML Schema and RESTful resource information.

2. CLIENTS

The Customer Portal API is, by nature, client-agnostic. It is expressed as a set of resource URLs which send and receive XML data. Some common clients and platforms include, but are not limited to, the following:

Client	Platform/Environment	Comments
cURL ^[a]	Command Line (Linux, UNIX, Mac OS X, Microsoft Windows)	Easy and transparent way to test commands and do simple integration
Apache HttpComponents ^[b]	Java	Most common Java library with which to talk HTTP; no intrinsic binding of XML
RESTEasy Client ^[c]	Java	Full Java model integration; no need to think about HTTP or XML; uses Apache HTTP Client underneath
<p>[a] http://curl.haxx.se/</p> <p>[b] https://hc.apache.org/</p> <p>[c] http://docs.jboss.org/resteasy/docs/2.0.0.GA/userguide/html/RESTEasy_Client_Framework.html</p>		

3. API COMMANDS IN RED HAT SUBSCRIPTION MANAGEMENT

Table 1. Activation Keys

Command	Details
GET/activation_keys	Lists activation keys
DELETE/activation_keys/{activation_key_id}	Removes an activation key
GET/activation_keys/{activation_key_id}	Lists an activation key

Command	Details
PUT/activation_keys/{activation_key_id}	Update an activation key
GET/activation_keys/{activation_key_id}/pools	Lists activation key pools
DELETE/activation_keys/{activation_key_id}/pools/{pool_id}	Deletes a pool from an activation key
PUT/activation_keys/{activation_key_id}/pools/{pool_id}	Updates a pool from an activation key
DELETE/activation_keys/{activation_key_id}/product/{product_id}	Removes a product from an activation key
POST/activation_keys/{activation_key_id}/product/{product_id}	Adds a product to an activation key

Table 2. Admin

Command	Details
DELETE/admin/cache/product	Clears the product cache
GET/admin/init	Initializes the Candlepin database
GET/admin/queues	Retrieves basic information on the HornetQ queues

Table 3. Atom

Command	Details
GET/atom	Retrieves an event atom feed

Table 4. CDN

Command	Details
GET/cdn	Retrieves a list of CDNs
DELETE/cdn/{label}	Removes a CDN
POST/cdn/{label}	Updates a CDN @return a CDN object

Table 5. Consumers

Command	Details
GET/consumers	Retrieves a list of consumers
POST/consumers	Creates a consumer
GET/consumers/{consumer_uuid}	Retrieves a single consumer
POST/consumers/{consumer_uuid}	Updates a consumer
DELETE/consumers/{consumer_uuid}	Deletes a consumer
POST/consumers/{consumer_uuid}	Retrieves a single consumer
GET/consumers/{consumer_uuid}/accessible content	Retrieves the body of the Content Access Certificate for the consumer
GET/consumers/{consumer_uuid}/atom	Retrieves an event feed for a consumer
PUT/consumers/{consumer_uuid}/certificates	Generates the entitlement certificates for the consumer
GET/consumers/{consumer_uuid}/certificates	Retrieves the entitlement certificates for the consumer
DELETE/consumers/{consumer_uuid}/certificates/{ serial}	Removes an entitlement certificate using the certificate serial
GET/consumers/{consumer_uuid}/certificates/seri als	Retrieves a list of the certificate seal's client metadata for the consumer
GET/{consumer_uuid}/compliance	Retrieves the compliance status of the consumer
DELETE/consumers/{consumer_uuid}/deletionreco rd	Removes the deletion record for a consumer allowed
GET/consumers/{consumer_uuid}/entitlements	Retrieves a list of entitlements for the consumer
DELETE/consumers/{consumer_uuid}/entitlements	Unbinds entitlements for a consumer
GET/consumers/{consumer_uuid}/entitlements/dry -run	Retrieves a list of pools and quantities that will auto- bind
DELETE/consumers/{consumer_uuid}/entitlements /pool/{pool_id}	Removes all entitlements for a consumer
PUT/entitlements/product/{product_id}	Regenerates an entitlement certificate for a product

Table 6. Environments

Command	Details
GET/environments	Retrieves a list of environments
DELETE/environments/{env_id}	Deletes an environment
GET/environments/{env_id}	Retrieves a single environment
POST/environments/{env_id}/consumers	Creates an environment
DELETE/environments/content	Demotes content from an environment
POST/environments/content	Promotes content into an environment

Table 7. Events

Command	Details
GET/events	Retrieves a list of events
GET/events/{uuid}	Retrieves a single event

Table 8. Hypervisors

Command	Details
POST/hypervisors/{owner}	Creates or updates the list of hypervisor hosts which allow agents such as virt-who to update hosts' information

Table 9. Jobs

Command	Details
GET/jobs	Retrieves a list of job status
DELETE/jobs/{job_id}	Deletes a single job
POST/jobs/{job_id}	Retrieves a job status and removes it if completed
POST/jobs/schedule/{task}	Begins specific tasks
GET/jobs/scheduler	Retrieves the scheduler status
POST/jobs/scheduler	Updates the scheduler status

Table 10. Owners

Command	Details
GET/owners	Retrieves a list of owners
POST/owners	Create an owner
DELETE/owners/{owner_key}	Removes an owner
GET/owners/{owner_key}	Retrieves a single owner
PUT/owners/{owner_key}	Updates an owner
GET/owners/{owner_key}/activation_keys	Retrieves a list of activation keys for an owner
POST/owners/{owner_key}/activation_keys	Creates an activation key for an owner
GET/owners/{owner_key}/atom	Retrieves an event atom feed for an owner
GET/owners/{owner_key}/consumers	Retrieves a list of consumers for the owner
GET/owners/{owner_key}/consumers/count	Retrieves a count of consumers for the owner
GET/owners/{owner_key}/content	Retrieves a list of content
POST/owners/{owner_key}/content	Creates a content
DELETE/owners/{owner_key}/content/{content_id}	Deletes a content
GET/owners/{owner_key}/content/{content_id}	Retrieves a single content
PUT/owners/{owner_key}/content/{content_id}	Updates a content
POST/owners/{owner_key}/content/batch	Creates contents in bulk
GET/owners/{owner_key}/entitlements	Retrieves the list of entitlements for an owner
POST/owners/{owner_key}/entitlements	Starts an asynchronous healing for an owner
GET/owners/{owner_key}/environments	Retrieves a list of environments for an owner
POST/owners/{owner_key}/environments	Creates an environment for an owner
GET/owners/{owner_key}/events	Retrieves a list of events for an owner
GET/owners/{owner_key}/hypervisor	Retrieves a list of hypervisors for an owner

Command	Details
DELETE/owners/{owner_key}/imports	Removes imports for an owner
GET/owners/{owner_key}/imports	Retrieves a list of import records for an owner
POST/owners/{owner_key}/imports/async	Initiates an asynchronous manifest import for the given organization
GET/owners/{owner_key}/info	Retrieves the owner info for an owner
DELETE/owners/{owner_key}/log	Remove the log level of the owner
PUT/owners/{owner_key}/log	Sets the log level of an owner
GET/owners/{owner_key}/pools	Retrieves the list of pools for an owner
POST/owners/{owner_key}/pools	Creates a custom pool for an owner
PUT/owners/{owner_key}/pools	Updates a pool for an owner
GET/owners/{owner_key}/products	Retrieves a list of products
POST/owners/{owner_key}/products	Creates a product
DELETE/owners/{owner_key}/products/{product_id}	Removes a product
GET/owners/{owner_key}/products/{product_id}	Retrieves a single product
PUT/owners/{owner_key}/products/{product_id}	Updates a product
POST/owners/{owner_key}/products/{product_id}/batch_content	Adds content to a product batch node
GET/owners/{owner_key}/products/{product_id}/certificate	Retrieves a certificate for a product
DELETE/owners/{owner_key}/products/{product_id}/content/{content_id}	Removes content from a product
POST/owners/{owner_key}/products/{product_id}/content/{content_id}	Adds content to a product single mode
PUT/owners/{owner_key}/products/{product_id}/content/subscriptions	Refreshes pools by product
GET/owners/{owner_key}/servicelevels	Retrieves a list of support levels for a customer

Command	Details
GET/owners/{owner_key}/subscriptions	Retrieves a list of subscriptions for an owner
PUT/owners/{owner_key}subscriptions	Tickle an owner to have all of their entitlement pools sync with their subscriptions
GET/owners/{owner_key}/uebercert	Retrieves the Ueber entitlement certificate
POST/owners/{owner_key}/uebercert	Creates an Ueber entitlement certificate
GET/owners/{owner_key}/upstream_customers	Retrieves a list of upstream customers for an owner

Table 11. Pools

Command	Details
DELETE/pools/{pool_id}	Remove a pool
GET/pools/{pool_id}	Retrieves a single pool
GET/pools/{pool_id}/cdn	Retrieves a CDN for a pool
GET/pools/{pool_id}/entitlements	Retrieves a list of entitlements for a pool

Table 12. Products

Command	Details
GET/products/{uuid}	Retrieves a single product
GET/products/{product_uuid}/certificate	Retrieves a certificate for a product
GET/products/owners	Retrieves a list of owners by product
PUT/products/subscriptions	Refreshes pools by product

Table 13. Roles

Command	Details
GET/roles	Retrieves a list of roles
POST/roles	Creates a role
DELETE/roles/{role_id}	Removes a role

Command	Details
GET/roles/{role_id}	Retrieves a single role
PUT/roles/{role_id}	Removes a permission from a role
DELETE/roles/{roles_id}permissions/username}	Removes a user from a role
POST/roles/{roles_id}permissions/username}	Adds a user to a role

Table 14. Rules

Command	Details
DELETE/rules	Deletes a rule
GET/rules	Retrieves a rule
POST/rules	Updates a rule

Table 15. Serials

Command	Details
GET/serials	Retrieves a list of certificate serials
GET/serials/{serial_id}	Retrieves a single certificate serial

Table 16. Status

Command	Details
GET/status	Returns the status of a server

Table 17. Subscriptions

Command	Details
GET/subscriptions	Retrieves a list of subscriptions
POST/subscriptions	Activates a subscription
DELETE/subscriptions/{subscription_id}	Removes a subscription
GET/subscription/{subscription_id}	Retrieves a single subscription

Command	Details
GET/subscriptions/{subscription_id}/cert	Retrieves a subscription certificate

Table 18. Users

Command	Details
GET/users	Retrieves a list of users
POST/user	Creates a user
DELETE/users/{username}	Removes a user
GET/users/{username}	Retrieves a single user
POST/users/{username}	Updates a user
GET/users/{username}/owners	Retrieves a list of owners that the user can register systems to
GET/users/{username}/roles	Retrieves a list of roles by the user

[1] http://en.wikipedia.org/wiki/Representational_State_Transfer