



# Subscription Central 1-latest

## Using APIs in Red Hat Subscription Management

authorizing, managing, and troubleshooting subscription management APIs



# Subscription Central 1-latest Using APIs in Red Hat Subscription Management

---

authorizing, managing, and troubleshooting subscription management APIs

## Legal Notice

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

Red Hat subscription services offer developed and documented APIs to help you better automate, manage, and track your subscriptions to Red Hat products.

---

## Table of Contents

<b>1. USING APIS IN RED HAT SUBSCRIPTION MANAGEMENT</b> .....	<b>2</b>
<b>2. USING TOKENS FOR AUTHENTICATION</b> .....	<b>2</b>
2.1. Generating a new offline token	2
2.2. Generating a new refresh token	3
<b>3. ACCESSING AVAILABLE SUBSCRIPTION MANAGEMENT APIS</b> .....	<b>4</b>
<b>4. TROUBLESHOOTING API ERRORS</b> .....	<b>4</b>
4.1. Troubleshooting error 403	5
4.2. Troubleshooting error 429	5

## 1. USING APIS IN RED HAT SUBSCRIPTION MANAGEMENT

Using APIs in subscription services can help you more effectively track and automate your Red Hat subscriptions usage in the following ways:

- Control which tools that you use for each product
- Better manage your system and subscription inventories
- Update and secure your systems more efficiently
- Continue receiving official support for your Red Hat products

Red Hat Subscription Management APIs use OAuth 2.0 for authorization. To obtain a token and access the APIs, you will need the following pieces of information:

- Offline token generated on the Red Hat Subscription Manager API Tokens page
- Client ID = rhsm-api
- Token URL = <https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token>

## 2. USING TOKENS FOR AUTHENTICATION

Offline and refresh tokens are used by Red Hat Subscription Management to authenticate your system after you set up your account using your secret to authenticate your Red Hat Single Sign On (SSO) account.



### WARNING

Please use password management that is consistent with security best practices. It is never safe to store any passwords or credentials in plaintext. Treat your offline token with the same security measures that you use for a password to protect it against unauthorized use.

### 2.1. Generating a new offline token

An offline token never expires as long as it is used at least once every 30 days and is used to create access tokens for the Red Hat Subscription Manager APIs. It works as a password and allows you to continue being able to authenticate your account without having to create new refresh tokens.

#### Procedure

1. Visit the [Red Hat Subscription Manager API Tokens page](#) .

2. Click the **Generate Token** button.

## 2.2. Generating a new refresh token

After you have created the offline token, you can use that token to create a new refresh token, which includes an access token that is valid for five minutes. Access tokens are used in the header to authenticate your Customer Portal user account to the Red Hat Subscription Manager APIs.

### Procedure

1. Set the offline token value. In this example, we set it in plaintext and shorten the token value for clarity:

```
# offline_token='eyJhbGciOiJSUzI1NiIsInR5cCI6IkpzZW50L3N1bWU6IiwiaWF0IjoiMjAxNjA0MjEzLjE1LjE1In0='
```

2. Enter the following command to create a function to easily filter out JSON values.

```
# function jsonValue() {
KEY=$1
num=$2
awk -F"[,:]" '{for(i=1;i<=NF;i++){if($i~/\"$KEY\"/){print $(i+1)}}}' | tr -d "" | sed -n ${num}p
}
```

3. Enter the following command to extract the **access\_token**:

```
# curl https://sso.redhat.com/auth/realms/redhat-external/protocol/openid-connect/token -d
grant_type=refresh_token -d client_id=rhsm-api -d refresh_token=$offline_token
```

The expected output is similar to the following, where **access\_token** is the token used to authorize your account:

```
{"access_token":"oiZjo1MjhhkNzZmZi1mNzA4LTQzZWQtOGNkNS1mZTE2ZjRmZTBjZTY6cmh
uLXN1cHBvcnQta3RvcnRldXliLCJ0eXAiOiJCZWZyZXIiLCJhenAiOiJyaHNtLWFwaSIsImF1dG
hfdGltZSI6MTU2NzQwODU5Nywic2Vzc2l2bWVzZGF0ZSI6ImYwZGJiOGQ0LTRINGUtNDY1N
C04NDRjLTZmMzcwNGM4NDQyMiIsImF1dGltZSI6ImYwZGJiOGQ0LTRINGUtNDY1NjYwZGJi
FsbV9hY2Nlc3MiOnsicm9sZXMiOlsicG9ydGFsX21hbmFnZV9zdWJzY3JpcHRpb25zliwib2Zmb
GluZV9hY2Nlc3MiLCJjYW5kbGVwaW5fc3lzdGVtX2FjY2Vzc192aWV3X2VkaXRfYWxsliwiYW
aW46b3JnOmFsbCIsInBvcnRhbF9tYW5hZ2VfY2FzZXMiLCJwb3J0YWxfxc3lzdGVtX21hbmFnZ
W1lbnQiLCJwb3J0YWxfZG93bmVxYWQiXX0sInJlc291cmNIX2FjY2Vzcy16e30sImFjY291bnRfa
WQiOiIiOTc5NzEwliwibmFtZSI6Iktlbm55IFRvcnRldXJzliwihJlZmVycmVkaX3VzZXJuYW1lIjoic
mhuLXN1cHBvcnQta3RvcnRldXliLCJnaXZlbnUyYW1lIjoic2VubnkiLCJmYW1pbHlfbmFtZSI6IIR
vcnRldXJzliwizW1haWwiOiJrdG9yZGV1ckByZWRoYXQuY29tIn0.JfStOgLvGFUAIMb7aVfm-
dWxd4wN5oqk377Q6oyDe55pM4zDiZ0f1yJfHsWL8RHeb3r0tj8DY_UAyAFkxAnjyWjq52d7h2Ef
JUPOs1p1P8Yeu5hDwOrA34Es2maN-ZbJcC4sOb7stGhxSCU15CfvPFIRR5tgSQ17-Mx-
x4ZnK_fwpOK6DqQpNzZ0Krz3U1a-NH86XJ8dT8IC3o03YrdlcZx_-wv6-
PehqNQa2Hb9vt1csX8QIL3PEyBVNPZXaaTHvyFYx0orGyjKA83Qq-
LihbWBXzNjf_rIEfsPJYi-
uQHIT_zjaOPYo2rXi7VTPJC2qRSxF2yaRGlhZHxkDzMOTITnaDeMhbx1zvRr-
R9eXocEUzsU9j-
Yx7h3WYCFjb8zdfXTBHV8SCaMdH1u9Eesa5gmHOOki8882RRR85i1fjpBayFTS36y4S-
yDebUYiukXOnw8mMMKy04NhVpFGfWtJ8--
Jy4Ypndqqk_OS_PiWBsFFN6IMv5S6DZWWpjE-
CENHK9ceA4MlerBBXLY02Xz9h0biiQUZrd-NLy11j4os124Mai1mmlNOLz993hw0gl-
vKKno_bYOv8dEEemKtSLISPVdW5X_0vBU0BtQuSEVctz_8zsRKHpT-
```





Code	Explanation	Resolution
403	Forbidden	Generate a new authorization token.
404	Not found	Resource not found or does not exist.
429	Too many requests	Reduce the frequency of requests
500	Internal server error	The problem is on Red Hat's end. Wait a minute and try your request again.

### 4.1. Troubleshooting error 403

Error 403 is a "not authorized" error, meaning that the authentication you are using for Red Hat Subscription Manager APIs has failed. There are two possible solutions you can try.

#### Procedure

1. To authenticate through the Red Hat Subscription Manager gateway, ensure the authorization header includes the text "bearer" before entering your API call:

```
curl -H "Authorization: Bearer <token>" <api_url>
```

2. If the header is correct, create a new token. Refresh tokens last for five minutes.

### 4.2. Troubleshooting error 429

Error 429 is a "rate limiting" error, meaning that your account has exceeded the number of allowed requests per second. This limit applies to all users of a single Red Hat account.

#### Procedure

Extract the header of the response, which includes: \* **X-RateLimit-Limit**: the total requests/sec allowed \* **X-RateLimit-Remaining**: the number of requests/sec remaining (this will be a negative integer) \* **X-RateLimit-Delay**: the number of seconds the requester should wait before trying again

Adjust the rate of requests to the X-RateLimit-Limit value and start again once the X-RateLimit-Delay time has passed.