Red Hat Subscription Management

Using APIs in Red Hat Subscription Management

authorizing, managing, and troubleshooting APIs in Red Hat Subscription Management
Using APIs in Red Hat Subscription Management

authorizing, managing, and troubleshooting APIs in Red Hat Subscription Management
Abstract

Red Hat’s Subscription Management has developed and documented APIs to help you better automate, manage, and track your subscriptions to Red Hat products.
# Table of Contents

1. USING APIS IN RED HAT SUBSCRIPTION MANAGEMENT ........................................... 2

2. USING TOKENS FOR AUTHENTICATION ................................................................... 2
   2.1. Generating a new offline token ................................................................. 2
   2.2. Generating a new refresh token ............................................................... 3

3. ACCESSING AVAILABLE RHSM APIs ................................................................... 4
   3.1. Examples of API calls .............................................................................. 4

4. TROUBLESHOOTING API ERRORS ....................................................................... 4
   4.1. Troubleshooting error 403 ....................................................................... 5
   4.2. Troubleshooting error 429 ....................................................................... 5

CHAPTER 1. APPENDIX A. REVISION HISTORY ......................................................... 6
1. USING APIs IN RED HAT SUBSCRIPTION MANAGEMENT

Using APIs in RHSM can help you more effectively keep track of and automate how you manage your Red Hat subscriptions and entitlement usage. By using APIs in RHSM, you can:

- Control which tooling you use for which products
- Better manage your system inventory
- 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 RHSM API Tokens page
- Client ID = rhsm-api

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 Customer Portal account.

**WARNING**

Please use password management that is consistent with networking 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 would 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 RHSM 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 RHSM API Tokens page.
2. Click the Generate Token button.
2.2. Generating a new refresh token

Once 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 passed in the header to authenticate your Customer Portal user to the RHSM 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='eyJhbGciOiJSUzI1NiIsInR5cCI6Iiwia2lkIjA6ICItNGVsY19WZE5fV3NPVVlmMkc0UXhyOEdjd0l4X0t0WFVDaXRhdExLbEx3In0.eyJqdGkiOiJhODZlZDczZS00ZTRlLTQ2NTUtYzI5ZS1iMzI0ZWNiYzQ2OTMiLCJlbmFtZSI6IjAiLCJhY2Nlc3Npb25faWQiOiJmMGRiYjhkNC00ZTRlLTQ2NTUtYzI5ZS1iMzI0ZWNiYzQ2OTMiLCJ0eXAiOiJodHRwczovL3Nzby5yZWRoYXQua29tL3Nzc3NpthZ3YoIn0.Ae5xyNz4iKCIKQlSvXcin15OAhS5SBOw5mT9m0kko7Ao114nQgDH1Ak5Z1zS49tYdY7Ju4DQWYd0p3JbwOwJF21O36y-BnJ9w5-pj74zHh97LQBjkh4L7QNSy0T6FpUZs6jGz-aQ1Rb96RswABy3m71UFRQ5S8dDk7Jr1h2mLW6c8Bq-oXjLvs2a-n9Dtt60-V5vH6x-hLl8L8bY1fBxwT8YaI5D2U-5YlYuA32uBQ1ySUDFlW49rw451B4Ql84b547v72v4J8dYl8u58nU8dOu4UD9bJq43J8d6AaObQp6JfZ7k-aZTjumr8U-V1702x-yu2hGwqj6vktF9zR9aBiGiv-IQgYgBd1HrjzU0BzC1r1Lm3cOG7y4S-R3k8L6j80jF1nWcmJ3m2c9J39ASurur9hj1UmqBk2shb0B1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
The access_token is what needs to be set/used as an authorization token to perform the API call.

```
# 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 | jsonValue
```

### 3. ACCESSING AVAILABLE RHSM APIs

Red Hat provides a Swagger file to describe the specifications of the Red Hat Subscription Management APIs. The Swagger specification includes information about the API endpoints available, input parameters, expected output, and possible error responses. The swagger file can be imported into REST clients like Postman or RESTlet to automatically build a library of API calls.

#### 3.1. Examples of API calls

The following examples describe how to use 'curl' to search for information about your deployment in Red Hat Subscription Management. These examples include both the actual command and some sample output.

### 4. TROUBLESHOOTING API ERRORS

<table>
<thead>
<tr>
<th>Code</th>
<th>Explanation</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>BadRequest error</td>
<td>Validate that you entered the API call correctly and try again.</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
<td>Generate a new authorization token.</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
<td>Generate a new authorization token.</td>
</tr>
</tbody>
</table>
### 4.1. Troubleshooting error 403

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

**Procedure**

1. To authenticate through the RHSM 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.
# CHAPTER 1. APPENDIX A. REVISION HISTORY

Table 1.1. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Changes Made</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision 1.1-0</td>
<td>Tues Sept 19 2019</td>
<td>Procedures for offline, refresh tokens have changed</td>
<td>Anni Bond</td>
</tr>
<tr>
<td>Revision 1.0-1</td>
<td>Wed May 8 2019</td>
<td>Added detail to prerequisites about where to get a secret</td>
<td>Anni Bond</td>
</tr>
<tr>
<td>Version 1.0-0</td>
<td>Fri May 3 2019</td>
<td>Initial creation</td>
<td>Anni Bond</td>
</tr>
</tbody>
</table>