



NECS 4.0 API Guide

Version 1.1
NECS AUTOMATION (NECS 4.0)



20 JAN. 2020

Contents

| | | |
|------|--|----|
| 1.0 | NECS Automation Programming Guide | 3 |
| 1.1 | Intended Audience..... | 3 |
| 2.0 | Overview of NECS 4.0 Automation REST API..... | 4 |
| 2.1 | REST API Services..... | 5 |
| 2.2 | Using the NECS Automation REST API..... | 5 |
| 3.0 | CloudClient CLI Usage | 6 |
| 3.1 | Installing CloudClient | 6 |
| 3.2 | Executing CloudClient..... | 8 |
| 3.3 | Executing Scripts on CloudClient..... | 10 |
| 4.0 | CloudClient CLI Use Cases | 11 |
| 4.1 | Introduction | 11 |
| 4.2 | List Catalog Items..... | 12 |
| 4.3 | List Machines | 12 |
| 4.4 | Power Off Machines | 12 |
| 4.5 | Create vSphere Reservation | 12 |
| 4.6 | Deploy Amazon Server..... | 13 |
| 4.7 | Get Computer Resource Field Values | 13 |
| 4.8 | Get Computer Resource Field Values | 15 |
| 4.9 | Deploy vSphere Server | 15 |
| 4.10 | Approve Request..... | 16 |
| 4.11 | Deploy a Server..... | 16 |
| 4.12 | Workflow Detail | 16 |
| 5.0 | REST API Authentication | 17 |
| 5.1 | About HTTP Bearer Tokens..... | 17 |
| 5.2 | Request an HTTP Bearer Token..... | 17 |
| 5.3 | Example: Token Request and Response..... | 18 |
| 5.4 | Syntax for Requesting an HTTP Bearer Token..... | 18 |
| 5.5 | Validate an HTTP Bearer Token..... | 20 |
| 5.6 | Example: Validate Token Request and Response | 20 |
| 5.7 | Delete an HTTP Bearer Token..... | 21 |
| 6.0 | Requesting a Product..... | 23 |
| 6.1 | Request a Product (e.g. new user) | 23 |
| 6.2 | Request a Product (example VM request) | 25 |

| | | |
|-----|--|-----|
| 6.3 | Catalog Service Examples for Requesting a Product | 27 |
| 6.4 | Syntax for Getting a Template Request for a Catalog Item | 35 |
| 6.5 | Syntax for Viewing Details of a Machine Request | 43 |
| 7.0 | Approving a Product Request | 47 |
| 7.1 | Approve a Machine Request | 47 |
| 7.2 | Work Item Service Examples for Approving a Machine Request | 48 |
| 7.3 | Syntax for Listing Work Items | 49 |
| 7.4 | Syntax for Getting Work Item Details | 55 |
| 7.5 | Syntax for Constructing a JSON File to Approve a Machine Request | 60 |
| 7.6 | Syntax for Approving a Submitted Product Request | 63 |
| 8.0 | Listing Provisioned Resources | 84 |
| 8.1 | Prerequisites for Listing Provisioned Resources | 85 |
| 8.2 | Display Your Provisioned Resources Example | 85 |
| 8.3 | Display Provisioned Resources by Resource Type Example | 89 |
| 8.4 | Display All Available Resource Types Example | 92 |
| 8.5 | View Resource Details Example | 94 |
| 8.6 | Using the API to Get Deployment Details | 96 |
| 9.0 | Managing Provisioned Deployments | 106 |
| 9.1 | Manage Provisioned Deployments | 106 |
| 9.2 | Power Off | 107 |
| 9.3 | Catalog Service Examples for Managing Provisioned Deployments | 110 |
| 9.4 | Syntax for Navigating to the Children of a Deployed Resource | 114 |

1.0 NECS Automation Programming Guide

The *API Guide* provides information about the NECS Automation REST APIs, including how to use the REST API services and resources, create HTTP bearer tokens for authentication and authorization, and construct REST API service calls.

1.1 Intended Audience

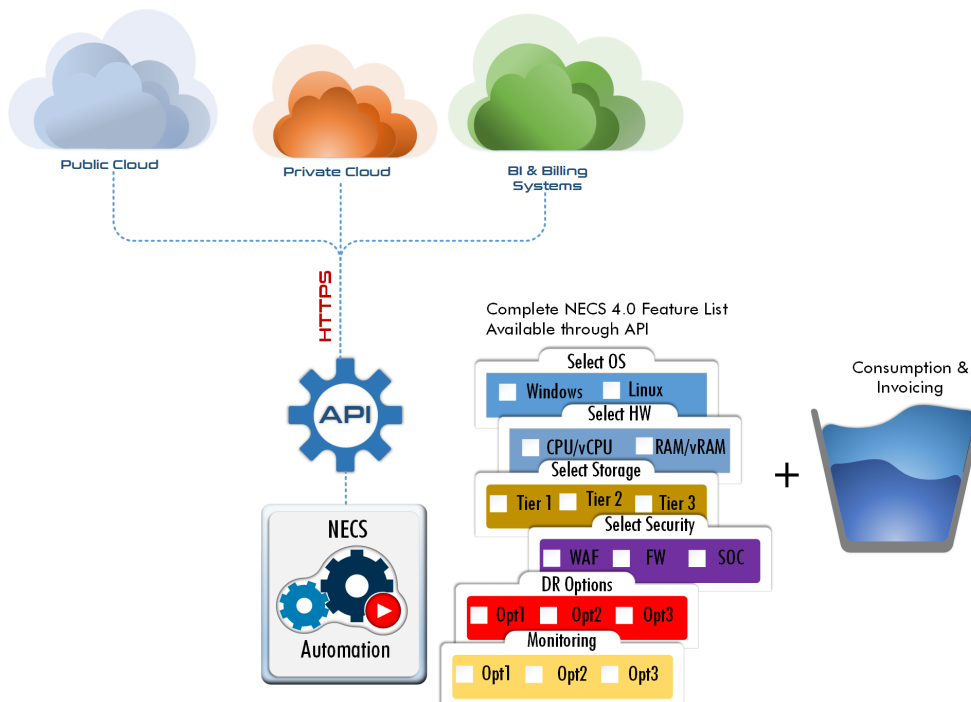
This information is intended for administrators and programmers who want to configure and manage NECS programmatically using the NECS Automation REST API. This guide focuses on common use cases

2.0 Overview of NECS 4.0 Automation REST API

The NRB Enterprise Cloud Services (NECS) Platform offers the entire API library of all NECS features open to our end users. Although the entire API is open for our end users, connectivity is secured over a site-to-site VPN connectivity between the two endpoints as a pre-requisite.

- NECS 4.0 REST API enables the creation of custom services by exposing any orchestration workflow to our Service Catalog to provide custom functionality such as Blueprints, Resource Mappings and Actions.
- Tenant administrators can create and manage the workflow subscriptions that are specific to their tenant.
- System administrators can create and manage system workflow subscriptions. The created system workflow subscriptions are active for events in any tenant and for system events.

The NECS Automation REST API provides consumer, administrator, and provider-level access to our service catalog with the same services that are presented in the console user interface. You can perform API Automation functions programmatically by using REST API service calls.



The NECS Automation REST API provides consumer, administrator, and provider-level access to the service catalog with the same services that support the NECS Cloud Management Portal (CMP) user interface. You can perform NECS Automation functions programmatically by using REST API service calls.

This chapter includes the following topics:

- [REST API Services](#)
- [Using the NECS Automation REST API](#)
- [About the API Use Cases](#)

2.1 REST API Services

The NECS Automation REST API offers the following services and functions.

Table 1-1. NECS Automation REST API Services

| Service | Description |
|-------------------|---|
| Catalog Service | Retrieve global and entitled catalog items, and entitlements for a catalog item and its service that the current user can review. A consumer can retrieve, edit, and submit a request form for a catalog item. A provider can retrieve, register, update, and delete catalog items. Provision and manage systems. |
| Work Item Service | Retrieve, create, update, complete, cancel, and delete a work item. Also retrieve form data, metadata, detail forms, and submission forms from service providers. service. |

2.2 Using the NECS Automation REST API

To make NECS Automation REST API service calls, you can use a browser application or an HTTP client program to send requests and review responses.

You can use:

An REST API client application such as Postman

Any programmatic REST API Client

Command-line tools such as cURL or Powershell

3.0 CloudClient CLI Usage

vRealize CloudClient is a command-line utility that provides verb-based access with a unified interface across vRealize Automation APIs.

This section describes the steps and different examples on how to use CloudClient with NECS API.

Please note, always execute the env.sh to initialize the CloudClient with the correct configuration and credentials.

3.1 Installing CloudClient

You can download the latest version of CloudClient from here:

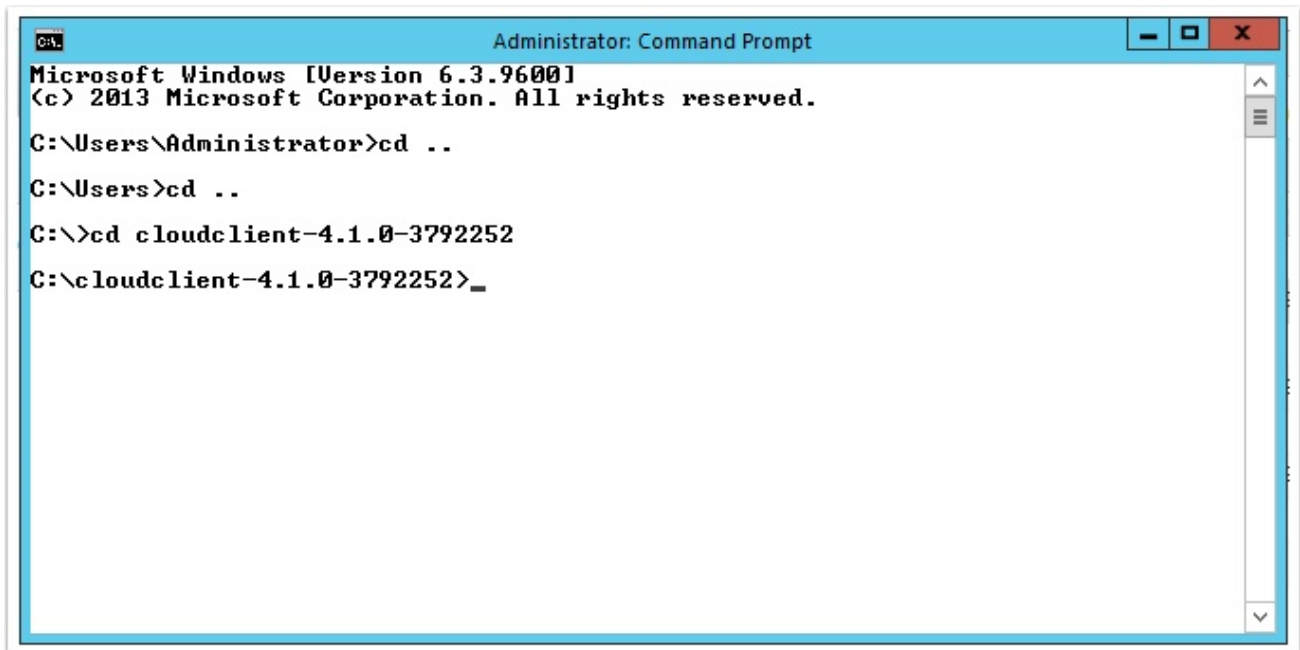
<https://code.vmware.com/tool/cloudclient/4.1.0>

Cloud Client supports both Windows and Linux, if you want to use windows your install is done, if you want to use Linux, you need to SCP (Secure Copy) the folder to the Linux Machine you want to use.

You can download WinSCP from here:

<https://winscp.net/eng/download.php>

Now your ready to use Cloud Client, on windows or Linux change to the directory where you copied cloud client to



```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd ..
C:\Users>cd ..
C:\>cd cloudclient-4.1.0-3792252
C:\cloudclient-4.1.0-3792252>_
```

Now change to the bin directory

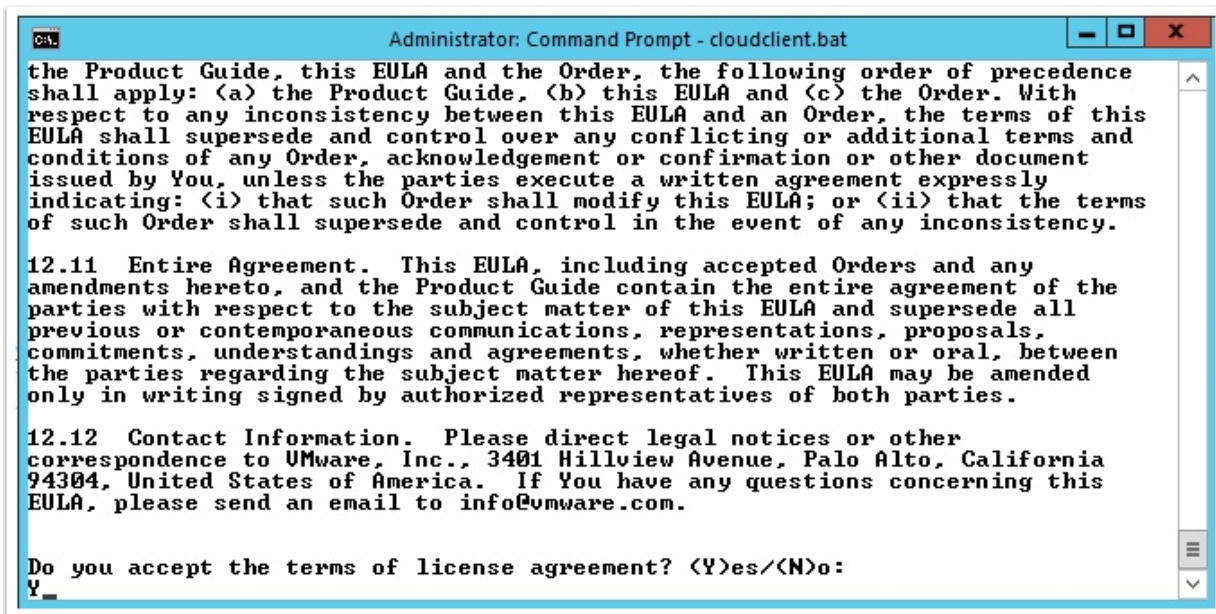
A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window has a blue title bar and standard Windows window controls (minimize, maximize, close) on the right. The command history is as follows:
Microsoft Windows [Version 6.3.9600]
<C> 2013 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>cd ..
C:\Users>cd ..
C:\>cd cloudclient-4.1.0-3792252
C:\cloudclient-4.1.0-3792252>cd bin
C:\cloudclient-4.1.0-3792252\bin>_
The cursor is at the end of the last command line.

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
<C> 2013 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>cd ..
C:\Users>cd ..
C:\>cd cloudclient-4.1.0-3792252
C:\cloudclient-4.1.0-3792252>cd bin
C:\cloudclient-4.1.0-3792252\bin>_
```


3.2 Executing CloudClient

- For windows, use the cloudclient.bat command
- For Linux, use the cloudclient.sh command (./cloudclient.sh)

Scroll through the EUL Agreement and accept the license agreement.



```
Administrator: Command Prompt - cloudclient.bat

the Product Guide, this EULA and the Order, the following order of precedence
shall apply: (a) the Product Guide, (b) this EULA and (c) the Order. With
respect to any inconsistency between this EULA and an Order, the terms of this
EULA shall supersede and control over any conflicting or additional terms and
conditions of any Order, acknowledgement or confirmation or other document
issued by You, unless the parties execute a written agreement expressly
indicating: (i) that such Order shall modify this EULA; or (ii) that the terms
of such Order shall supersede and control in the event of any inconsistency.

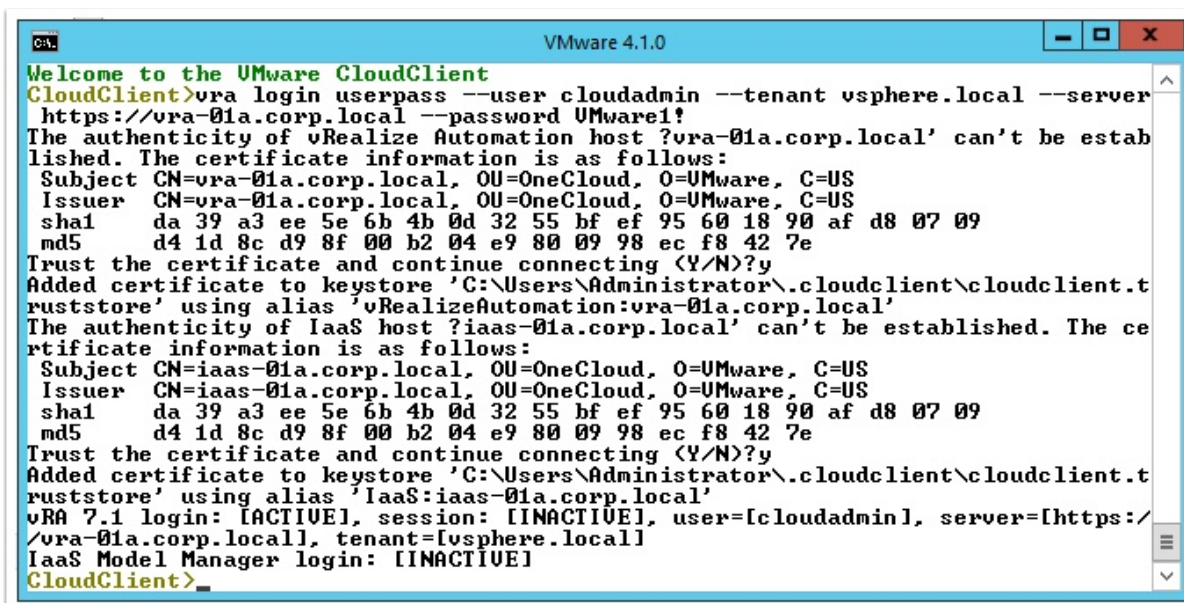
12.11 Entire Agreement. This EULA, including accepted Orders and any
amendments hereto, and the Product Guide contain the entire agreement of the
parties with respect to the subject matter of this EULA and supersede all
previous or contemporaneous communications, representations, proposals,
commitments, understandings and agreements, whether written or oral, between
the parties regarding the subject matter hereof. This EULA may be amended
only in writing signed by authorized representatives of both parties.

12.12 Contact Information. Please direct legal notices or other
correspondence to VMware, Inc., 3401 Hillview Avenue, Palo Alto, California
94304, United States of America. If You have any questions concerning this
EULA, please send an email to info@vmware.com.

Do you accept the terms of license agreement? (Y)es/(N)o:
Y_
```

Now login to the vRealize Automation with the following command:

(Note: if you are using a self-signed certificate, you will also need to accept them here).



```
VMware 4.1.0

Welcome to the VMware CloudClient
CloudClient>vra login userpass --user cloudadmin --tenant vsphere.local --server
https://vra-01a.corp.local --password VMware1!
The authenticity of vRealize Automation host 'vra-01a.corp.local' can't be estab
lished. The certificate information is as follows:
  Subject CN=vra-01a.corp.local, OU=OneCloud, O=VMware, C=US
  Issuer  CN=vra-01a.corp.local, OU=OneCloud, O=VMware, C=US
  sha1    da 39 a3 ee 5e 6b 4b 0d 32 55 bf ef 95 60 18 90 af d8 07 09
  md5     d4 1d 8c d9 8f 00 b2 04 e9 80 09 98 ec f8 42 7e
Trust the certificate and continue connecting (Y/N)?y
Added certificate to keystore 'C:\Users\Administrator\.cloudclient\cloudclient.t
ruststore' using alias 'vRealizeAutomation:vra-01a.corp.local'
The authenticity of IaaS host 'iaas-01a.corp.local' can't be established. The ce
rtificate information is as follows:
  Subject CN=iaas-01a.corp.local, OU=OneCloud, O=VMware, C=US
  Issuer  CN=iaas-01a.corp.local, OU=OneCloud, O=VMware, C=US
  sha1    da 39 a3 ee 5e 6b 4b 0d 32 55 bf ef 95 60 18 90 af d8 07 09
  md5     d4 1d 8c d9 8f 00 b2 04 e9 80 09 98 ec f8 42 7e
Trust the certificate and continue connecting (Y/N)?y
Added certificate to keystore 'C:\Users\Administrator\.cloudclient\cloudclient.t
ruststore' using alias 'IaaS:iaas-01a.corp.local'
vRA 7.1 login: [ACTIVE], session: [INACTIVE], user=[cloudadmin], server=[https:/
/vra-01a.corp.local], tenant=[vsphere.local]
IaaS Model Manager login: [INACTIVE]
CloudClient>
```

```
vra login userpass --user cloudadmin --tenant vsphere.local --server https://vra-01a.corp.local --password VMware1!
```

You may also add CloudClient bin directory to PATH and then run cloudclient.bat or cloudclient.sh directly without the need to change the directories to the directory where you extracted the CloudClient files.

On Windows, open a command prompt on your workstation, and run the command

```
SET PATH=%PATH%;DriveLetter:\full\path\to\CloudClient\bin\directory\
```

On Unix or Linux or MacOSX, open a terminal on your workstation, and run the command

```
export PATH=$PATH:/full/path/to/CloudClient/bin/directory/
```

The Cloud Client interface appears.

```
$ sh bin/cloudclient.sh
JRE Version: 1.8.0_65
=====
CloudClient
=====
Version : 4.0.0
*Tip* : You can hit tab to move through commands and arguments
*Tip* : You can autologon with a CloudClient.properties file

Welcome to the VMware CloudClient
CloudClient>
```

3.3 Executing Scripts on CloudClient

Please note: you need to execute the following env.sh shell script to establish the correct running environment before running any scripts via CloudClient.

env.sh

You will need to update the cloudclient_home variable to reflect the current installation path you have chosen for CloudClient.

Please update this file with the correct variable definitions for your NECS Environment:

```
#!/bin/sh

export vra_server=server_name
export vra_username=user1
export vra_keyfile=keyfile.enc
#export vra_password=#If you want to use cleartext password, use this instead of
vra_keyfile
export vra_tenant=vsphere.local

#####
# use these variables where applicable, otherwise hash them out
#####

export vco_server=10.10.10.10
export vco_username=administrator@vsphere.local
export vco_password=mypassword

# Uncomment below and replace with valid cloudclient home path (unless you already
# this environment variable set)
#export cloudclient_home=/path/to/cloudclient
```

Please refer to the table below for the variable definitions.

| Variable | Description |
|-------------------|--|
| vra_server | vRA Server Name or IP Address |
| vra_tenant | Tenant to connect to, defaults to vsphere.local if left empty |
| vra_username | vRA Username - to login to Top level system administrator the username is ""administrator@vsphere.local |
| vra_password | vRA Password |
| vra_keyfile | Location to encrypted keyfile |
| vra_iaas_server | vRA Infrastructure Server Name, if left blank it is automatically discovered |
| vra_iaas_username | vRA NTLM Username, ie: Administrator |
| vra_iaas_password | vRA NTLM Password |
| vra_iaas_keyfile | Location to encrypted keyfile |

4.0 CloudClient CLI Use Cases

4.1 Introduction

These scripts will help highlight how to use CloudClient with our APIs. Please remember to include the .env.sh script as this defines the environment and paths.

Although this document has REST API calls that are issued with the curl command, the homologous CLI commands can be issued with CloudClient.

With CloudClient, there is no need to set heading values, including the Authorization header. The `$host/$servicename/api` is eliminated from the URL and the service name becomes a separate parameter. For example, `consumer/entitled CatalogItems/{id}/request/template`

Some parameters on the API request are common to all content management service import and export commands. These parameters are listed below.

| Parameter | Description |
|-----------|--|
| \$Host | The host name and fully qualified domain name or IP address of the vRealize Automation identity server |
| \$token | A valid HTTP bearer token that includes necessary credentials. |

In order to use the CLI (CloudClient) to issue the same commands as what is presented in section 5.0 in the REST API section, you simply need to execute the env.sh, run the command client with the verbs for the specific catalog you require and input the json file given in section 5.0 for each respect API call.

For example:

- a) Get a list of the catalog item you wish to execute

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra catalog list --format
JSON --export /tmp/list-ci.txt
```

Or

```
$> cloudclient.sh vra catalog list
```

- b) Execute the CloudClient with the required parameters using the JSON input file. Please see section 5 for the specific input file

```
$cloudclient_home/bin/cloudclient.sh vra [Catalog List Verb] --
inputfile /tmp/temp.json
```

4.2 List Catalog Items

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra catalog list --format
JSON --export /tmp/list-ci.txt
```

4.3 List Machines

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra machines list --format
CSV --export /tmp/list-machines.txt
```

4.4 Power Off Machines

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Provide Machine Name and Action, a guid or name is supported (as variables)

```
export machine='"Finance0011"'
export action='"Power Off"'
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra machines action execute -
-id $machine --action $action
```

4.5 Create vSphere Reservation

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Create a new reservation

```
$cloudclient_home/bin/cloudclient.sh vra reservation add --type
VSPHERE --inputfile /tmp/vsphere-request.json

# A sample json file can be used to create multiple reservations by providing the
values for arguments to 'vra reservation add' command.
# If values are provided for 'vra reservation add' command arguments '--
businessGroupId', '--enable', '--name', '--priority', '--reservationPolicyId', and
'--tenantId', these values are used instead of the ones in the json file provided
in the '--inputfile' argument.
# Using the following command, another reservation can be created with just the
values defined in command arguments being different.

# $cloudclient_home/bin/cloudclient.sh vra reservation add --type
VSPHERE --inputfile /tmp/vsphere-request.json --
businessGroupId='valid-business-group-id-1' --enable false --name
'Reservation-1' --priority 1 --reservationPolicyId 'valid-
reservation-policy-id-1' --tenant 'valid-tenant-id-1'
```

4.6 Deploy Amazon Server

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

Provide CatalogItem/Blueprint and Group Name, a guid or name is supported as variables. These can be added to the env.sh file

```
export catalog=' "Apache Web Server (Amazon) " '
export group=' "CSE Dev" '
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra catalog request submit --
id $catalog --groupid $group --reason test --properties
"__amazon.instanceType=t1.micro,__amazon_advancedCapabilitiesMode=
false,__amazon.ebs.totalStorageRequested=0,Vrm.DataCenter.Location
=us-west-2a"
```

4.7 Get Computer Resource Field Values

Setup environment variables for auto login to CloudClient Shell

```
./env.sh
```

```
# Get the field ids using 'vra reservation schema' command or the included sample
script GetVSPHEREReservationSchema.sh
```

```
# Schema states that there are several fields needed to create a vSphere
reservation. While some of the fields depend on other fields, others don't depend
on any.
# First retrieve the permissible values for all the fields that do not depend on
other fields and choose a value.
# Once the values are chosen for the fields that do not depend on others, provide
them as input in order to get permissible values for fields that depend on other
fields.
# You may dry run the 'vra reservation field values' command without any
dependency values, in order to get the list of dependency fields for a field.
# 'vra reservation field values' command gives permissible values to only those
fields where the user has to select a value from the available values like
networks, storages, etc.
# No need to run 'vra reservation field values' command for fields like
machineQuota, reservationMemory that do not have permissible values as per schema.
```

Provide chosen compute resource id

```
export computeResourceId="fe299b30-c12a-4fc2-b4a1-04ac059f29e8"
```

Execute CloudClient to get values for the other fields.

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId vCNSTransportZone --dependencyValues
computeResource=$computeResourceId --export /tmp/vcns-transport-
zone-values.json
```

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId reservationNetworks --dependencyValues
computeResource=$computeResourceId --export /tmp/reservation-
networks-values.json
```

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId vCNSSecurityGroups --dependencyValues
computeResource=$computeResourceId --export /tmp/vcns-security-
groups-values.json
```

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId reservationStorages --dependencyValues
computeResource=$computeResourceId --export /tmp/reservation-
storages-values.json
```

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId resourcePool --dependencyValues
computeResource=$computeResourceId --export /tmp/resource-pool-
values.json
```

#Fields that do not just depend on computeResource such as vCNSRoutedGateways.vCNSNetworkPath and vCNSRoutedGateways.vCNSNetworkProfile need additional dependency values.

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
```

```
--type VSPHERE --fieldId vCNSRoutedGateways --dependencyValues
computeResource=$computeResourceId --export /tmp/vcns-routed-
gateways-values.json
```

Choose a value for vCNSEdge

```
export vcnsEdgeId="c7e5a78b-bacb-4bcd-8e8a-992710643b52"

$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId vCNSRoutedGateways.vCNSNetworkPath --
dependencyValues
computeResource=$computeResourceId,vCNSEdge=$vcnsEdgeId --export
/tmp/vcns-network-path-values.json
```

Choose a value for vCNSNetworkPath

```
export vcnsNetworkPathId="a4b5d10f-08b1-408f-a481-ba11e37d0b61"

$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId vCNSRoutedGateways.vCNSNetworkProfile --
dependencyValues
computeResource=$computeResourceId,vCNSNetworkPath=$vcnsNetworkPat
hId --export /tmp/vcns-network-profile-values.json
```

4.8 Get Computer Resource Field Values

Setup environment variables for auto login to CloudClient Shell

```
. ./env.sh
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra reservation field values
--type VSPHERE --fieldId computeResource --export /tmp/compute-
resource-values.json
```

4.9 Deploy vSphere Server

Setup environment variables for auto login to CloudClient Shell

```
. ./env.sh
```

Provide CatalogItem/Blueprint and Group Name, a guid or name is supported

```
export catalog='"Centos (Dev) "'
export group='"CSE Dev"'
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra catalog request submit --
id $catalog --groupid $group --reason test
```


4.10 Approve Request

Setup environment variables for auto login to CloudClient Shell

```
.. /env.sh
```

Provide CatalogItem/Blueprint and Group Name, a guid or name is supported

```
export requestId='"7c7db360-8747-4b7a-9deb-15ddcdee6918"'
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra request approve --ids  
$requestId --message approved
```

4.11 Deploy a Server

Setup environment variables for auto login to CloudClient Shell

```
.. /env.sh
```

Provide CatalogItem/Blueprint and Group Name, a guid or name is supported

```
export catalog='"Centos (Dev) "'  
export group='"CSE Dev"'
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vra catalog request submit --  
id $catalog --groupid $group --reason test
```

4.12 Workflow Detail

Setup environment variables for auto login to CloudClient Shell

```
.. /env.sh
```

Provide WorkflowId and VCO JSON Payload

```
export wflowId=3c656670-cd72-49bc-9c79-831067c17d42  
export requestFile=vco.json
```

Execute CloudClient

```
$cloudclient_home/bin/cloudclient.sh vco workflow detail --id  
$requestFile --requestfile $requestFile
```

5.0 REST API Authentication

In the REST API, NECS Automation requires HTTP bearer tokens in request headers for the authentication of consumer requests. A consumer request applies to tasks that you can perform in the NECS Cloud Management Portal (CMP), such as requesting a machine.

To acquire an HTTP bearer token, you authenticate with an identity service that manages the communication with the SSO server. The identity service returns an HTTP bearer token that you include in all request headers until the token expires, or you delete it. An HTTP bearer token expires in 24 hours by default, but you can configure the token with a different duration.

This chapter includes the following topics:

- [About HTTP Bearer Tokens](#)
- [Configure the Duration of an HTTP Bearer Token](#)
- [Request an HTTP Bearer Token](#)
- [Validate an HTTP Bearer Token](#)
- [Delete an HTTP Bearer Token](#)

5.1 About HTTP Bearer Tokens

You use HTTP bearer tokens for tasks that you can also perform in the NECS Cloud Management Portal (CMP). You create a request header with the curl command or with some other utility.

You use POST, HEAD, and DELETE methods to manage HTTP bearer tokens.

| Method | URL | Description |
|--------|-------------------------|---|
| POST | /tokens | Authenticate the user with the identity service /tokens and generate a new token. |
| HEAD | /tokens/ <i>tokenId</i> | Validate the token <i>tokenId</i> . |
| DELETE | /tokens/ <i>tokenId</i> | Delete the token <i>tokenId</i> . |

Use the following root URL for HTTP bearer token calls:

```
https://$NECS/identity/api/tokens
```

The variable *\$NECS* represents the appliance name.domain name of the NECS Automation server such as, NECS-appliance-name.company.com.

5.2 Request an HTTP Bearer Token

You use an HTTP bearer token to authenticate a NECS Automation REST API consumer request.

A consumer request must specify the correct component registry service and resource. For example, the URL to obtain an HTTP bearer token must specify the identity service and token resource.

5.3 Example: Token Request and Response

The following sample displays output based on the example request.

```
curl --insecure -H "Accept: application/json" -H 'Content-Type: application/json' --data
'{"username":"necs-user@company.com","password":"necs-user-
password","tenant":"company.com"}' https://$NECS/identity/api/tokens
{"expires":"2017-04-14T04:46:43.000Z","id":"MTQ5Mj...M2RmMA==","tenant":"company.com"}
```

The id is the bearer token to store for future use.

```
export token="MTQ5Mj...M2RmMA=="
```

If the credentials supplied in the Authorization header are invalid, the response includes status code 401 as in the following output.

```
<!DOCTYPE html><html><head><title>Error report</title></head><body><h1>HTTP Status 401 -
Authentication required</h1></body></html>
```

5.4 Syntax for Requesting an HTTP Bearer Token

An HTTP bearer token is required by the REST client to use the NECS Automation REST API. You obtain a bearer token by authenticating to the identity service.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|----------------|--|
| URL | https://\$NECS/identity/api/tokens |
| \$NECS | <i>appliance name.domain name</i> of the NECS Automation server. |
| usname | Tenant administrator user name. |
| passwd | Tenant administrator password. |
| tenantURLtoken | Tenant URL token determined by the system administrator when creating the tenant such as, support. |

Output

The following information is displayed as a result of your HTTP bearer token request.

| Parameter | Description |
|-----------|---|
| expires | Contains the ISO 8601 timestamp indicating when the token expires. |
| id | Contains the HTTP bearer token to use in Authorization header in subsequent requests. |

Response Status Codes

One of the following codes are displayed as a result of your HTTP bearer token request.

| Status Code | Description |
|------------------|--|
| 200 OK | Your request succeeded and the resource was updated. The response body contains the full representation of the resource. |
| 400 BAD REQUEST | The data you provided in the POST failed validation. Inspect the response body for details. |
| 401 UNAUTHORIZED | The request could not authenticate the user or authentication credentials required. |

Example: curl Command to Request HTTP Bearer Token

The following example command requests an HTTP bearer token.

```
curl --insecure -H "Accept: application/json" -H 'Content-Type: application/json' --data
'{"username":"usrname","password":"passwd","tenant":"tenantURLtoken"}' https://$NECS/identity/api/tokens
```

When your request succeeds, the system returns the expiration date and time of the token, and the HTTP bearer token.

5.5 Validate an HTTP Bearer Token

You can validate an existing HTTP bearer token.

Prerequisites

- [Request an HTTP Bearer Token.](#)

Procedure

- 1 Enter the command to validate the HTTP bearer token.

```
curl --insecure -I -H "Accept: application/json" -H "Authorization: Bearer $token" -H "Cache-
Control: no-cache" "https://$NECS/identity/api/tokens/$token"
```

- 2 Examine the response.

A successful request returns status code 204.

5.6 Example: Validate Token Request and Response

The following sample displays output based on the example request.

```
curl --insecure -I -H "Accept: application/json" -H "Authorization: Bearer $token" -H "Cache-Control:
no-cache" "https://$NECS/identity/api/tokens/$token"
HTTP/1.1 204
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-XSS-Protection: 1; mode=block
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Content-Type: application/json;charset=UTF-8
Date: Thu, 13 Apr 2017 21:56:02 GMT
X-Frame-Options: SAMEORIGIN
```

The server returns one of the following status codes.

Table 2-1. Status Codes for Validate a Bearer Token

| Status Code | Description |
|------------------------|--|
| 204 NO CONTENT | The request succeeded. |
| 401 UNAUTHORIZED | You must have authentication credentials to access the resource. All requests must be authenticated. |
| 403 FORBIDDEN | Your authentication credentials do not provide sufficient access to the resource. |
| 404 NOT FOUND | Could not locate the resource based on the specified URI. |
| 405 METHOD NOT ALLOWED | The HEAD method is not supported for the resource. |
| 500 SERVER ERROR | Could not create or update the resource because of an internal server error. |

5.7 Delete an HTTP Bearer Token

You can delete an HTTP bearer token.

Prerequisites

- [Request an HTTP Bearer Token.](#)

Procedure

- 1 Enter the command to delete the HTTP bearer token, as in the following example.

```
curl --insecure -X DELETE -H "Accept: application/json" -H "Authorization: Bearer $token" -H "Cache-Control: no-cache" "https://$NECS/identity/api/tokens/$token"
```

- 2 Examine the response.

A successful request returns status code 204.

Example: Delete Token Request and Response

The following sample displays output based on the example request.

```
curl --insecure -X DELETE -H "Accept: application/json" -H "Authorization: Bearer $token" -H "Cache-Control: no-cache" "https://$NECS/identity/api/tokens/$token"
204 NO CONTENT
```

The server returns one of the following status codes.

Table 2-2. Status Codes for Delete a Bearer Token

| Status Code | Description |
|------------------|--|
| 204 NO CONTENT | The request succeeded. The resource has been deleted. |
| 401 UNAUTHORIZED | You must have authentication credentials to access the resource. All requests must be authenticated. |
| 403 FORBIDDEN | Your authentication credentials do not provide sufficient access to the resource. |
| 404 NOT FOUND | Could not locate the resource based on the specified URI. |

Table 2-2. Status Codes for Delete a Bearer Token (Continued)

| Status Code | Description |
|------------------------|--|
| 405 METHOD NOT ALLOWED | The DELETE method is not supported for the resource. |
| 500 SERVER ERROR | Could not create or update the resource because of an internal server error. |

6.0 Requesting a Product

You use the catalog service to perform tasks related to requesting a machine.

The catalog service is comprised APIs for the consumer, service providers, and service administrators. It is designed to be used by consumers and providers of the service catalog. For example, a consumer would request a catalog item such as a machine. The service provider would fulfill the request.

The catalog service includes Hypermedia as the Engine of Application State (HATEOAS) links. The links function as templates that you can use to complete common tasks supported by the API.

For example, if you submit a template request for a given context, such as: `catalog-service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests/template`. You use the returned template, either as-is or modified, to create a request that you POST or PUT to the target API, such as: `catalog-service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests`.

This chapter includes the following topics:

- [Add a User \(User Management\)](#)
- [Request a Machine](#)
- [Requesting a Consumption or Financial Report](#)

6.1 Request a Product (e.g. new user)

Within NRB's API, a user is considered a product, and hence to request a new user follows the same logic for creating or any product. This example is for Management Portal related API calls.

Prerequisites

- Log in to NECS Automation as a consumer or current business group user.
- Verify that the appliance name and fully qualified domain name of the NECS Automation instance are available.
- Verify that you have a valid HTTP bearer token that matches your login credentials. See [REST API Authentication](#).

Procedure

- 1 List all shared catalog items in the catalog.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/entitledCatalogItemViews
```

For details regarding input and output for this request, see [Syntax for Listing Shared and Private Catalog Items](#).

- 2 Examine the response to find the *catalogItemId*
- 3 Get a template request for a catalog item.

Use the *catalogItemId* to submit the template request for this catalog item. In this example, the *catalogItemId* is dc808d12-3786-4f7c-b5a1-d3r9141g9352.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests/template
```

For details regarding input and output for this request, see [Syntax for Getting a Template Request for a Catalog Item](#).

A template request for the catalog item is created. The fields and default values are populated based on the configuration of the underlying blueprint. By default, requestMachine.json is the name of the template request.

- 4 Review and edit the template request.

Review the contents of the template request and edit the values if you want to change them from the default prior to submitting the request for a machine. For example, you can specify a value for the description field or change the values for the machine resources if the blueprint allows for a range.

- 5 Submit the request for a machine.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-
d5f997c8ad66/requests --verbose --data
@C:/Temp/requestMachine.json
{
    $contentsOfTemplateFromPrecedingSections
}
```

For details regarding input and output for this request see [Syntax for Requesting a Machine](#).

6 (Optional) View the details of your request.

You can perform a GET on the URI in the Location header to retrieve the updated request details. In this example, the *URI-in-Location-header* is 7aaf9baf-aa4e-47c4-997b-edd7c7983a5b.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b
```

For details regarding input and output for this request, see [Syntax for Viewing Details of a Machine Request \(Which is also based on this API call\)](#).

The sample data to be provided for the catalog item "user" are given here:

```
"requestData": {
  "entries": [
    {
      "key": "firstName",
      "value": "Jeremy"
    },
    {
      "key": "lastName",
      "value": "Jones"
    },
    {
      "key": "emailAddress",
      "value": "Jeremy.jones@jones.com"
    },
    {
      "key": "roles",
      "value": "manager"
    },
    {
      "key": "mobile",
      "value": "+32 475 555 555"
    }
  ]
}
```

6.2 Request a Product (example VM request)

To request a machine, you first list all shared catalog items to find the machine, then make the request for that item using a template.

Prerequisites

- Log in to NECS Automation as a consumer or current business group user.
- Verify that the appliance name and fully qualified domain name of the NECS Automation instance are available.
- Verify that you have a valid HTTP bearer token that matches your login credentials. See [REST API Authentication](#).

Procedure

- 7 List all shared catalog items in the catalog.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/entitledCatalogItemViews
```

For details regarding input and output for this request, see [Syntax for Listing Shared and Private Catalog Items](#).

- 8 Examine the response to find the *catalogItemId*

- 9 Get a template request for a catalog item.

Use the *catalogItemId* to submit the template request for this catalog item. In this example, the *catalogItemId* is dc808d12-3786-4f7c-b5a1-d5f997c8ad66.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests/template
```

For details regarding input and output for this request, see [Syntax for Getting a Template Request for a Catalog Item](#).

A template request for the catalog item is created. The fields and default values are populated based on the configuration of the underlying blueprint. By default, requestMachine.json is the name of the template request.

- 10 Review and edit the template request.

Review the contents of the template request and edit the values if you want to change them from the default prior to submitting the request for a machine. For example, you can specify a value for the description field or change the values for the machine resources if the blueprint allows for a range.

- 11 Submit the request for a machine.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-
d5f997c8ad66/requests --verbose --data
@C:/Temp/requestMachine.json
{
    $contentsOfTemplateFromPrecedingSections
}
```

For details regarding input and output for this request see [Syntax for Requesting a Machine](#).

12 (Optional) View the details of your request.

You can perform a GET on the URI in the Location header to retrieve the updated request details. In this example, the *URI-in-Location-header* is `7aaf9baf-aa4e-47c4-997b-edd7c7983a5b`.

```
curl --insecure -H "Accept: application/json" -H "Content-Type: application/json" -H
"Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b
```

For details regarding input and output for this request, see [Syntax for Viewing Details of a Machine Request](#).

6.3 Catalog Service Examples for Requesting a Product

Syntax for each service example lists input parameters, output parameters, and curl commands.

- [Syntax for Listing Shared and Private Catalog Items](#)

GET `/api/consumer/entitledCatalogItemViews` retrieves a list of all shared viewable catalog items for the current user. Shared catalog items do not belong to a specific business group. This service also retrieves a list of all shared and private catalog items that can be viewed, including their business groups.

- [Syntax for Getting Information for a Catalog Item](#)

GET `/api/consumer/entitledCatalogItemViews/{id}` gets information about a specific catalog item.

- [Syntax for Getting a Template Request for a Catalog Item](#)

GET `/api/consumer/entitledCatalogItems/{id}/requests/template` retrieves a template request for a specific catalog item. NRB supplies a number of templates to help you create different types of machine requests.

- [Syntax for Requesting a Machine](#)

POST `/api/consumer/entitledCatalogItems/{id}/requests` submits a request for a specific catalog item with input provided in a JSON file.

- [Syntax for Viewing Details of a Machine Request](#)

GET `/api/consumer/requests/{requestId}` provides the details of a machine request, where *requestId* is the URI in the Location header.

Syntax for Listing Catalog Items

GET `/api/consumer/entitledCatalogItemViews` retrieves a list of all shared viewable catalog items for the current user. Shared catalog items do not belong to a specific business group. This service also retrieves a list of all shared and private catalog items that can be viewed, including their business groups.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-------------|---|
| URL | https://\$NECS/catalog-service/api/consumer/entitledCatalogItemViews |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| page number | The page number. Default is 1. |
| limit | The number of entries per page. The default is 20. |
| \$orderby | <p>Multiple comma-separated properties sorted in ascending or descending order. Valid OData properties include the following:</p> <ul style="list-style-type: none"> ■ name - filter based on catalog item name. ■ status - filter based on catalog item status. ■ service/id - filter based on catalog item service id. ■ service/name - filter based on catalog item service name. ■ organization/subTenant/id - filter based on catalog item business group ID, which you can find in the catalogItem payload under organization > subtenantRef ■ organization/subTenant/name - filter based on catalog item business group name, which you can find in catalogItem payload under organization >subtenantLabel ■ outputResourceType/id - filter based on catalog item output resource type ID, for example : Infrastructure.Virtual ■ outputResourceType/name - Filter based on catalog item output resource type name, for example: "VirtualMavhine". ■ catalogItemType/id - filter based on catalog item type ID, for example: "Infrastructure.Virtual". ■ catalogItemType/name - filter based on catalog item type name, forexample: "VirtualMachine". ■ icon/id - filter based on catalog item icon ID. |
| \$top | Sets the number of returned entries from the top of the response |
| \$skip | Sets the number of entries to skip. |

| Parameter | Description |
|-----------------|--|
| <i>\$filter</i> | <p>Boolean expression for whether a particular entry should be included in the response. Valid OData properties include the following:</p> <ul style="list-style-type: none"> ■ name - filter based on catalog item name. ■ status - filter based on catalog item status. ■ service/id - filter based on catalog item service id. ■ service/name - filter based on catalog item service name. ■ organization/subTenant/id - filter based on catalog item business group ID, which you can find in the catalogItem payload under organization > subtenantRef ■ organization/subTenant/name - filter based on catalog item business group name, which you can find in catalogItem payload under organization >subtenantLabel ■ outputResourceType/id - filter based on catalog item output resource type ID, for example : Infrastructure.Virtual ■ outputResourceType/name - Filter based on catalog item output resource type name, for example: "VirtualMavhine". ■ catalogItemType/id - filter based on catalog item type ID, for example: "Infrastructure.Virtual". ■ catalogItemType/name - filter based on catalog item type name, forexample: "VirtualMachine". ■ icon/id - filter based on catalog item icon ID. |
| serviceId | (Optional) Query parameter to filter the returned catalog items by one specific service. |
| onBehalfOf | (Optional) Query parameter that provides the value of the user ID when making a request on behalf of another user. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|-----------------------|--|
| outputResourceTypeRef | Specifies the type of the resource that results from requesting the catalog item. |
| catalogItemId | Specifies the catalog item identifier. |
| name | Specifies the user-friendly name of the catalog item. Specifies the property type is string. |
| description | Specifies a short description of the catalog item. Specifies the property type is string. |
| catalogItemTypeRef | Specifies the type of the catalog item. |
| serviceRef | Specifies the catalog service that contains the catalog item. |
| iconId | Specifies the associated icon representing this item. |
| isNoteworthy | Specifies if the catalog item should be highlighted to users for a period of time. |
| dateCreated | Specifies the date that this item was created in the catalog. |
| lastUpdatedDate | Specifies the date that this item was last updated in the catalog. |
| entitledOrganizations | Specifies the organizations in which the catalog item can be consumed by the current user. |

Example: curl Command to List All Shared Catalog Items

The following example command retrieves information about all shared catalog items of type `ConsumerEntitledCatalogItemView`.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token" https://$NECS/catalog-service/api/consumer/entitledCatalogItemViews
```

If backward compatibility is required, use the following example command instead.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token" https://$NECS/catalog-service/api/consumer/entitledCatalogItems
```

The following JSON output is returned based on the command input.

```
{
  "links": [],
  "content": [
    {
      "@type": "ConsumerEntitledCatalogItemView",
      "links": [
        {
          "@type": "link",
          "rel": "GET: Request Template",
          "href": "https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/7c8275d6-1bd6-452a-97c4-d6c053e4baa4/requests/template"
        },
        {
          "@type": "link",
          "rel": "POST: Submit Request",
          "href":
"https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/7c8275d6-1bd6-452a-97c4-d6c053e4baa4/requests"
        }
      ],
      "entitledOrganizations": [
        {
          "tenantRef": "mycompany",
          "tenantLabel": "mycompany",
          "subtenantRef": "c0683388-6db2-4cb5-9033-b24d15ad3766",
          "subtenantLabel": "Demo Group"
        }
      ],
      "catalogItemId": "dc808d12-3786-4f7c-b5a1-d5f997c8ad66",
      "name": "Linux",
      "description": "Linux blueprint for API demo",
      "isNoteworthy": false,
      "dateCreated": "2015-07-29T03:54:28.141Z",
      "lastUpdatedDate": "2015-07-29T12:46:56.405Z",
      "iconId": "cafe_default_icon_genericCatalogItem",
      "catalogItemTypeRef": {
        "id": "com.NRB.csp.component.cafe.composition.blueprint",

```

```

    },
    "serviceRef": {
      "id": "057d4095-95f1-47da-b14b-641ac9010c97",
      "label": "Infrastructure Services"
    },
    "outputResourceTypeRef": {
      "id": "composition.resource.type.deployment",
      "label": "Deployment"
    }
  }
],
"metadata": {
  "size": 20,
  "totalElements": 1,
  "totalPages": 1,
  "number": 1,
  "offset": 0
}
}

```

Example: curl Command to Locate the Details of a Specific Catalog Item

To search for specific catalog item, add the \$catalogItemId. The following example command retrieves information about a catalog item with the name \$catalogItemName.

```

curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token" https://$NECS/catalog-service/api/consumer/entitledCatalogItemViews?
$filter=name+eq+%27$catalogItemName%27

```

Syntax for Getting Information for a Catalog Item

GET /api/consumer/entitledCatalogItemViews/{id} gets information about a specific catalog item.

REST API Catalog Service

The REST API supports OData filtering. For more information about supported OData filters, refer to the NECS Automation API Reference, particularly the REST API Tips page located at [https://\\$NECS/component-registry/services/docs/odata.html](https://$NECS/component-registry/services/docs/odata.html).

For specific information about catalog service filters, see the "Important Notes About catalog-service and OData Queries" topic located at [https://\\$NECS/catalog-service/api/docs/index.html](https://$NECS/catalog-service/api/docs/index.html).

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-------------|---|
| URL | https://\$NECS/catalog-service/api/consumer/entitledCatalogItemViews/{id} |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| page number | The page number. Default is 1. |
| limit | The number of entries per page. The default is 20. |
| \$orderby | <p>Multiple comma-separated properties sorted in ascending or descending order. Valid OData properties include the following:</p> <ul style="list-style-type: none"> ■ name - filter based on catalog item name. ■ status - filter based on catalog item status. ■ service/id - filter based on catalog item service id. ■ service/name - filter based on catalog item service name. ■ organization/subTenant/id - filter based on catalog item business group ID, which you can find in the catalogItem payload under organization > subtenantRef ■ organization/subTenant/name - filter based on catalog item business group name, which you can find in catalogItem payload under organization >subtenantLabel ■ outputResourceType/id - filter based on catalog item output resource type ID, for example : Infrastructure.Virtual ■ outputResourceType/name - Filter based on catalog item output resource type name, for example: "VirtualMavhine". ■ catalogItemType/id - filter based on catalog item type ID, for example: "Infrastructure.Virtual". ■ catalogItemType/name - filter based on catalog item type name, forexample: "VirtualMachine". ■ icon/id - filter based on catalog item icon ID. |
| \$top | Sets the number of returned entries from the top of the response |
| \$skip | Sets the number of entries to skip. |

| Parameter | Description |
|-----------------|--|
| <i>\$filter</i> | <p>Boolean expression for whether a particular entry should be included in the response. Valid OData properties include the following:</p> <ul style="list-style-type: none"> ■ name - filter based on catalog item name. ■ status - filter based on catalog item status. ■ service/id - filter based on catalog item service id. ■ service/name - filter based on catalog item service name. ■ organization/subTenant/id - filter based on catalog item business group ID, which you can find in the catalogItem payload under organization > subtenantRef ■ organization/subTenant/name - filter based on catalog item business group name, which you can find in catalogItem payload under organization >subtenantLabel ■ outputResourceType/id - filter based on catalog item output resource type ID, for example : Infrastructure.Virtual ■ outputResourceType/name - Filter based on catalog item output resource type name, for example: "VirtualMavhine". ■ catalogItemType/id - filter based on catalog item type ID, for example: "Infrastructure.Virtual". ■ catalogItemType/name - filter based on catalog item type name, forexample: "VirtualMachine". ■ icon/id - filter based on catalog item icon ID. |
| serviceId | (Optional) Query parameter to filter the returned catalog items by one specific service. |
| onBehalfOf | (Optional) Query parameter that provides the value of the user ID when making a request on behalf of another user. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|-----------------------|--|
| outputResourceTypeRef | Specifies the type of the resource that results from requesting the catalog item. |
| catalogItemId | Specifies the catalog item identifier. |
| name | Specifies the user-friendly name of the catalog item. Specifies the property type is string. |
| description | Specifies a short description of the catalog item. Specifies the property type is string. |
| catalogItemTypeRef | Specifies the type of the catalog item. |
| serviceRef | Specifies the catalog service that contains the catalog item. |
| iconId | Specifies the associated icon representing this item. |
| isNoteworthy | Specifies if the catalog item should be highlighted to users for a period of time. |
| dateCreated | Specifies the date that this item was created in the catalog. |
| lastUpdatedDate | Specifies the date that this item was last updated in the catalog. |
| entitledOrganizations | The list of organizations in which the current user can consume the catalog item. |

Example: curl Command to Get Information for a Catalog Item

The following example command retrieves information catalog item with the name `$filter=name+eq+%27$catalogItemName%27`.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token" https://$NECS/catalog-service/api/consumer/entitledCatalogItemViews?
$filter=name+eq+%27$catalogItemName%27
```

The following JSON output is returned based on the command input.

```
{
  "links": [],
  "content": [
    {
      "@type": "ConsumerEntitledCatalogItemView",
      "links": [
        {
          "@type": "link",
          "rel": "GET: Request Template",
          "href": "https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/7c8275d6-1bd6-452a-97c4-d6c053e4baa4/requests/template"
        },
        {
          "@type": "link",
          "rel": "POST: Submit Request",
          "href":
            "https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/7c8275d6-1bd6-452a-97c4-d6c053e4baa4/requests"
        }
      ],
      "entitledOrganizations": [
        {
          "tenantRef": "mycompany",
          "tenantLabel": "mycompany",
          "subtenantRef": "c0683388-6db2-4cb5-9033-b24d15ad3766",
          "subtenantLabel": "Demo Group"
        }
      ],
      "catalogItemId": "7c8275d6-1bd6-452a-97c4-d6c053e4baa4",
      "name": "Linux",
      "description": "Linux blueprint for API demo",
      "isNoteworthy": false,
      "dateCreated": "2015-07-29T03:54:28.141Z",
      "lastUpdatedDate": "2015-07-29T12:46:56.405Z",
      "iconId": "cafe_default_icon_genericCatalogItem",
      "catalogItemTypeRef": {
        "id": "com.NRB.csp.component.cafe.composition.blueprint",
        "label": "Composite Blueprint"
      },
      "serviceRef": {
        "id": "057d4095-95f1-47da-b14b-641ac9010c97",
        "label": "Infrastructure Services"
      }
    }
  ]
}
```

```

        "outputResourceTypeRef": {
            "id": "composition.resource.type.deployment",
            "label": "Deployment"
        }
    },
    "metadata": {
        "size": 20,
        "totalElements": 1,
        "totalPages": 1,
        "number": 1,
        "offset": 0
    }
}

```

6.4 Syntax for Getting a Template Request for a Catalog Item

GET /api/consumer/entitledCatalogItems/{id}/requests/template retrieves a template request for a specific catalog item. NRB supplies a number of templates to help you create different types of machine requests.

Overview

In the entitledCatalogItemViews response, a link field contains a value similar to the following.

```

{
    "@type": "link",
    "href": "https://$NECS/catalog-service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests/template",
    "rel": "GET: Request Template"
}

```

This URL is a HATEOAS link for a template request for this catalog item. The rel field provides a description of the link (request template) and indicates the HTTP method to use with the URI in the href field (GET). By using these HATEOAS links, you can make follow-on API calls without having to consult the API documentation for the URI syntax or construct the links programmatically.

Review and Edit the Template Request

The returned template request is specific to the applicable catalog item. The fields and default values are populated based on the configuration of the underlying blueprint.

You can review the contents of the template and optionally edit the values if you want to change them from the default prior to submitting the request. For example, you can specify a value for the description field or change the values for the machine resources if the blueprint allows for a range.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|-------------------------------|
| id | The UUID of the catalog item. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|-----------------------|---|
| entitledOrganizations | The list of organizations in which the current user can consume the catalog item. |
| catalogItemId | Specifies the catalog item identifier. |

Example: curl Command to Get a Template Request for a Catalog Item

The following example command retrieves a template request for the catalog item with ID dc808d12-3786-4f7c-b5a1-d5f997c8ad66.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token" https://$NECS/catalog-
service/api/consumer/entitledCatalogItems/dc808d12-3786-4f7c-b5a1-d5f997c8ad66/requests/template
```

The following JSON output is returned based on the command input.

Note Price is referred to as cost in API commands and output.

```
{
  "type": "com.NRB.vcac.catalog.domain.request.CatalogItemProvisioningRequest",
  "catalogItemId": "7c8275d6-1bd6-452a-97c4-d6c053e4baa4",
  "requestedFor": "csummers@example.com",
  "businessGroupId": "c0683388-6db2-4cb5-9033-b24d15ad3766",
  "description": null,
  "reasons": null,
  "data": {
    "Existing_Network_1": {
      "componentTypeId": "com.NRB.csp.component.cafe.composition",
      "componentId": null,
      "classId": "Blueprint.Component.Declaration",
      "typeFilter": "LinuxDemo*Existing_Network_1",
      "data": {
        "_cluster": 1,
        "_hasChildren": false,
        "description": null,
        "name": "Existing Network",
        "networkname": "Existing Network",
        "subnetmask": "255.255.255.0"
      }
    },
    "vSphere-Linux": {
      "componentTypeId": "com.NRB.csp.component.cafe.composition",
      "componentId": null,
      "classId": "Blueprint.Component.Declaration",
```

```

"typeFilter": "LinuxDemo*vSphere-Linux",
"data": {
  "Cafe.Shim.VirtualMachine.MaxCost": 0,
  "Cafe.Shim.VirtualMachine.MinCost": 0,
  "_cluster": 1,
  "_hasChildren": false,
  "action": "FullClone",
  "allow_storage_policies": false,
  "archive_days": 0,
  "blueprint_type": "1",
  "cpu": 1,
  "custom_properties": [],
  "daily_cost": 0,
  "datacenter_location": null,
  "description": null,
  "disks": [
    {
      "componentTypeId": "com.NRB.csp.iaas.blueprint.service",
      "componentId": null,
      "classId": "Infrastructure.Compute.Machine.MachineDisk",
      "typeFilter": null,
      "data": {
        "capacity": 6,
        "id": 0,
        "initial_location": "",
        "is_clone": false,
        "label": "",
        "storage_reservation_policy": "",
        "userCreated": true,
        "volumeId": 0
      }
    }
  ],
  "display_location": false,
  "guest_customization_specification": null,
  "lease_days": 0,
  "machine_actions": [
    "DESTROY",
    "POWER_ON",
    "CONNECT_RDP_SSH",
    "REPROVISION",
    "POWER_CYCLE",
    "EXPIRE",
    "SUSPEND",
    "CONNECT_REMOTE_CONSOLE",
    "CONNECT_USING_VDI"
  ],
  "machine_prefix": {
    "componentId": null,
    "classId": "Infrastructure.Compute.MachinePrefix",
    "id": "Use group default"
  },
  "max_network_adapters": 0,
  "max_per_user": 0,
  "max_volumes": 60,

```

```

"memory": 4096,
"nics": [
  {
    "componentTypeId": "com.NRB.csp.iaas.blueprint.service",
    "componentId": null,
    "classId": "Infrastructure.Compute.Machine.Nic",
    "typeFilter": null,
    "data": {
      "address": "",
      "assignment_type": "DHCP",
      "custom_properties": null,
      "id": 0,
      "load_balancing": "",
      "network_profile": "Existing Network"
    }
  }
],
"number_of_instances": 1,
"os_arch": "x86_64",
"os_distribution": null,
"os_type": "Linux",
"os_version": null,
"platform_name": "vsphere",
"platform_type": "virtual",
"property_groups": [
  null
],
"provisioning_workflow": {
  "componentId": null,
  "classId": "Infrastructure.Compute.ProvisioningWorkflow",
  "id": "CloneWorkflow"
},
"reservation_policy": {
  "componentId": null,
  "classId": "Infrastructure.Reservation.Policy.ComputeResource",
  "id": "None"
},
"security_groups": [],
"security_tags": [],
"source_machine": null,
"source_machine_external_snapshot": null,
"source_machine_name": "cbpcentos_63_x86",
"source_machine_vmsnapshot": null,
"storage": 6
}
}
}

```

Syntax for Requesting a Product (Example for Consumption or Financial Report)

POST /api/consumer/entitledCatalogItems/{id}/requests submits a request for a specific catalog item with input provided in a JSON file.

Prepare your Request

From the entitledCatalogItemViews response, locate the link field that contains a value similar to the following:

```
{
  "catalogItemId" : "bfe6aa7c-0741-4e6f-8c8a-58a7fd19cd29",
  "requestedFor" : null,
  "businessGroupId" : "709715a9-7331-4df3-a501-00db7b79e2d2",
  "description" : "Daily charge back verification for my beloved CFO",
  "reasons" : "Export Daily Financial Data",
  "data" : {
    "NRB.ReportType" : "Chargeback",
    "NRB.ReportFrequency" : "Daily",
    "NRB.ReportCurrency" : "EUR",
    "NRB.ReportDetails" : "Detailed",
    "NRB.ReportFormat" : "CSV",
    "NRB.ReportDeliveryChanel" : "Email"
  },
  "type" : "com.vmware.vcac.catalog.domain.request.CatalogItemProvisioningRequest"
}
```

Use the information in this response to edit the template construct the URI to request the desired catalog item using a POST command.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------------|--|
| URL | https://\$NECS/catalog-service/api/consumer/entitledCatalogItems/\$catalogId/requests |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| catalogItemId | The identifier of a catalog item. Typically, this is provided by users via the REST URI when making an entitledCatalogItem provisioning request. |
| requestedFor | The user for whom this request is being made. Must be the fully qualified user ID. Typically this is provided by the REST URI when making an entitledCatalogItem provisioning request. |
| businessGroupId | The business group identifier associated with the request. Typically this is provided via the REST URI when making the request. |
| description | The catalog item description. |
| reasons | |
| data | Context-specific properties. Obtain the consumerEntitledCatalogItem template request to identify the properties available for a given context. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|------------------------|---|
| version | Displays the object version number. |
| state | Specifies the item state, such as submitted. |
| approvalStatus | Specifies a status indicating whether this request has been approved, rejected, or is still pending some form of approval. |
| Property | Description |
| waitingStatus | Specifies a status indicating whether this request is waiting on any external users or services before it is able to progress. |
| requestNumber | Specifies a more user-friendly identifier for this request. |
| executionStatus | Specifies the current execution status of the request. |
| stateName | Specifies the localized state name. |
| phase | Specifies the current phase of the request, which is more coarse grained and easier for users to understand. |
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| description | Contains a brief description of this request. |
| reasons | Specifies the business reasons entered by the requestor or owner of this request. |
| requestedFor | Specifies the ID of the user for whom this request is logged. |
| requestedBy | Specifies the ID of the user who actually submitted the request |
| organization | Subtenant and/or tenant owner of this request. |
| requestorEntitlementId | Specified the value of the requestorEntitlement setting. |
| preApprovalId | Specifies the ID of the preApproval setting. |
| postApprovalId | Specifies the ID of the approval generated for the post-provisioning workflow step. |
| dateCreated | Specifies the date when this request was sent to the catalog. |
| lastUpdated | Specifies the date when this request was last updated. |
| dateSubmitted | Specifies the date when this request was first submitted. |
| dateApproved | Specifies the date when this request was approved. |
| dateCompleted | Specifies the date when this request was completed. |
| quote | Contains a quote made by the provider defining the estimated price(es) associated with the request and/or any resources provisioned as a result of the request. |
| requestCompletion | Contains additional request completion information. |
| requestData | Contains a map of the provider-specific field-value pairs collected for this request. |

For financial and consumption reports, please use these key value pairs:

```
"NRB.ReportType" : "Chargeback",
"NRB.ReportFrequency" : "Daily", -> Monthly, Hourly, Weekly etc
"NRB.ReportCurrency" : "EUR", -> USD etc
"NRB.ReportDetails" : "Detailed", -> Simple
"NRB.ReportFormat" : "CSV", -> XML
"NRB.ReportDeliveryChanel" : "Email" -> SharePoint
```

| | |
|--------------------------|---|
| retriesRemaning | <p>Specifies the number of attempts remaining to move this request from its current state to the next state in the request workflow.</p> <p>Some state transitions require calls to external services. These calls may fail due to transient errors such as momentary network errors. In these cases, the catalog will retry the call a number of times before failing.</p> <p>This property defines the number of retries remaining for the current state transition. When it reaches 0, the catalog will stop retrying and mark the request as failed. This property is reset to the default number of retries for every new operation that is triggered.</p> |
| requestedItemName | <p>Specifies the item name.</p> |
| requestedItemDescription | <p>Specifies the item description.</p> |
| components | <p>Returns the list of components associated with the request. The provider supplies this list of components after request initialization.</p> |

Example: curl Command to Request a Machine

To construct your request, refer to the entitledCatalogItemViews response received when you ran the request described in [Syntax for Getting a Template Request for a Catalog Item](#), locate a link field that contains a value similar to the following:

```
{
  "@type":"link",
  "href":"https://$NECS/catalog-service/api/consumer/entitledCatalogItems/f89fcbbf-7716-4a61-addd-a822dd4206f6/requests",
  "rel":"POST: Submit Request"
}
```

The following example command submits a machine request using appropriately edited template content from the entitledCatalogItemViews response.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/entitledCatalogItems/f89fcbbf-7716-4a61-addd-a822dd4206f6/requests
{
  $contentsOfTemplateFromPrecedingSections
}
```

Example: Output with Request and Response Headers

The following sample displays the request and response headers and the command output. Use the indicated JSON text file or inline text as input.

```
{
Accept = application/json Content-
Type = application/json Content-
Length = 2806
}
Response Headers
{
Date = Wed, 03 Dec 2014 20:58:34 GMT
ETag = "0"
Location = https://$NECS/catalog-service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b
{
  $requestObjectDetails
}

Content-Type = application/json;charset=UTF-8
Content-Length = 0
Vary = Accept-Encoding,User-Agent
Keep-Alive = timeout=15, max=100
Connection = Keep-Alive
}
null
```

6.5 Syntax for Viewing Details of a Machine Request

GET /api/consumer/requests/{requestId} provides the details of a machine request, where *requestId* is the URI in the Location header.

Request Status

Typically, the request status information is the most important part of request details. The phase field corresponds to the status displayed in the Requests tab in the interface. You can rerun this command multiple times to monitor the state of a machine request.

Table 4-1. Request Phase Status

| Phase | Description | End State? |
|-----------------------|--|------------|
| UNSUBMITTED | Request was saved but not submitted. | No |
| PENDING_PRE_APPROVAL | Request is subject to approval - pre-provisioning approval required. | No |
| IN_PROGRESS | Request is in progress, machine is being provisioned. | No |
| PENDING_POST_APPROVAL | Request is subject to approval, post-provisioning approval required. | No |
| SUCCESSFUL | Request completed successfully. The machine is available under provisioned resources on the Items tab. | Yes |
| FAILED | Request failed. | Yes |
| REJECTED | Request approval was rejected and will not complete. | Yes |

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-------------|--|
| URL | https://\$NECS/catalog-service/api/consumer/requests/\$requestId |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| \$requestId | Specifies the request ID. See Display Your Provisioned Resources Example to view all of your requests and search for a request ID. The required request ID is located at the end of the Location URL in the response header. The request ID is located in the Location field of the response header if you submitted the request with the <code>--headers</code> flag. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|------------------------|---|
| version | Displays the object version number. |
| state | Specifies the item state, such as submitted. |
| approvalStatus | Specifies a status indicating whether this request has been approved, rejected, or is still pending some form of approval. |
| waitingStatus | Specifies a status indicating whether this request is waiting on any external users or services before it is able to progress. |
| requestNumber | Specifies a more user-friendly identifier for this request. |
| executionStatus | Specifies the current execution status of the request. |
| stateName | Specifies the localized state name. |
| phase | Specifies the current phase of the request, which is more coarse grained and easier for users to understand. |
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| description | Contains a brief description of this request. |
| reasons | Specifies the business reasons entered by the requestor or owner of this request. |
| requestedFor | Specifies the ID of the user for whom this request is logged. |
| requestedBy | Specifies the ID of the user who actually submitted the request |
| organization | Subtenant and/or tenant owner of this request. |
| requestorEntitlementId | Specified the value of the requestorEntitlement setting. |
| preApprovalId | Specifies the ID of the preApproval setting. |
| postApprovalId | Specifies the ID of the approval generated for the post-provisioning workflow step. |
| dateCreated | Specifies the date when this request was sent to the catalog. |
| lastUpdated | Specifies the date when this request was last updated. |
| dateSubmitted | Specifies the date when this request was first submitted. |
| dateApproved | Specifies the date when this request was approved. |
| dateCompleted | Specifies the date when this request was completed. |
| quote | Contains a quote made by the provider defining the estimated price(es) associated with the request and/or any resources provisioned as a result of the request. |
| requestCompletion | Contains additional request completion information. |
| requestData | Contains a map of the provider-specific field-value pairs collected for this request. |
| retriesRemaining | <p>Specifies the number of attempts remaining to move this request from its current state to the next state in the request workflow.</p> <p>Some state transitions require calls to external services. These calls may fail due to transient errors such as momentary network errors. In these cases, the catalog will retry the call a number of times before failing.</p> <p>This property defines the number of retries remaining for the current state transition. When it reaches 0, the catalog will stop retrying and mark the request as failed. This property is reset to the default number of retries for every new operation that is triggered.</p> |
| requestedItemName | Specifies the item name. |

| Property | Description |
|--------------------------|---|
| requestedItemDescription | Specifies the item description. |
| components | Returns the list of components associated with the request. The provider supplies this list of components after request initialization. |

Example: curl Command to View the Details of the Machine Request

The following example command displays details of a request.

```
curl --insecure -H "Content-Type: application/json" -H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b
```

The following sample output contains information about the catalog item request 7aaf9baf-aa4e-47c4-997b-edd7c7983a5b.

```
{
  "@type": "CatalogItemRequest",
  "id": "7aaf9baf-aa4e-47c4-997b-edd7c7983a5b",
  "iconId": "cafe_default_icon_genericCatalogItem",
  "version": 6,
  "requestNumber": 8,
  "state": "SUCCESSFUL",
  "description": "API test",
  "reasons": null,
  "requestedFor": "csummers@example.com",
  "requestedBy": "csummers@example.com",
  "organization": {
    "tenantRef": "mycompany",
    "tenantLabel": "mycompany",
    "subtenantRef": "c0683388-6db2-4cb5-9033-b24d15ad3766",
    "subtenantLabel": "Demo Group"
  },
  "requestorEntitlementId": "1b409157-152c-43c4-b4cc-1cdef7f6adf8",
  "preApprovalId": null,
  "postApprovalId": null,
  "dateCreated": "2015-07-29T13:50:33.689Z",
  "lastUpdated": "2015-07-29T13:55:35.951Z",
  "dateSubmitted": "2015-07-29T13:50:33.689Z",
  "dateApproved": null,
  "dateCompleted": "2015-07-29T13:55:35.949Z",
  "quote": {},
  "requestCompletion": {
    "requestCompletionState": "SUCCESSFUL",
    "completionDetails": null
  },
  "requestData": {
    $detailsOfSubmittedRequest
  },
  "retriesRemaining": 3,
  "requestedItemName": "Linux",
  "requestedItemDescription": "Linux blueprint for API demo",
}
```

```
"stateName": "Successful",
"approvalStatus": "POST_APPROVED",
"executionStatus": "STOPPED",
"waitingStatus": "NOT_WAITING",
"phase": "SUCCESSFUL",
"catalogItemRef": {
  "id": "7c8275d6-1bd6-452a-97c4-d6c053e4baa4",
  "label": "Linux"
}
}
```

Note In the request details, the phase field corresponds to the status that is displayed in the Requests tab in the user interface.

7.0 Approving a Product Request

You use a series of work item service commands to approve a machine request.

Basic components of the work item service are the work item and the assignment. The work item service provides a standard way to present work items to users. For example, a user can view all work items and select the item to perform such as approving a machine request.

This chapter includes the following topics:

- [Approve a Machine Request](#)
- [Work Item Service Examples for Approving a Machine Request](#)

7.1 Approve a Machine Request

To approve a machine request, you first get a work item ID, then specify the ID in the approval.

Prerequisites

- Log in to NECS Automation as an approver with at least one of the following qualifications:
 - You are designated as an approver in an approval policy.
 - You belong to a group which has been designated as an approval group in an approval policy.
 - You are designated as a delegate for someone who is an approver.
- Verify that the appliance name and fully qualified domain name of the NECS Automation instance are available.
- Verify that you have a valid HTTP bearer token that matches your login credentials. See [REST API Authentication](#).

Procedure

- 1 List all available work item IDs.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-
service/api/workitems
```

For details regarding input and output for this request, see [Syntax for Listing Work Items](#).

- 2 Examine the response to find the *workItemId*

3 Get details for a specific work item ID.

Use the *workItemId* to get the details for this work item. In this example, the *workItemId* is 5e3e9519-78ea-4409-a52c-e4aa3bc56511.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-service/api/workitems/5e3e9519-78ea-4409-a52c-e4aa3bc56511
```

For details regarding input and output for this request, see [Syntax for Getting Work Item Details](#).

4 Construct a JSON file that contains the work item ID information that you need to approve a machine request.

- a Copy the appropriate JSON input file template to a new file in an XML editor that maintains formatting.
- b Substitute the input variables in the template with the values you obtained for your specific work item ID, for example 5e3e9519-78ea-4409-a52c-e4aa3bc56511.
- c Save the file with a new name, for example, approve.json.

For details regarding input and output for this request, see [Syntax for Constructing a JSON File to Approve a Machine Request](#).

5 Approve the submitted machine request by specifying the work item ID and including the JSON file as part of the command line.

```
curl --insecure -H "Content-Type:application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-service/api/workitems/5e3e9519-78ea-4409-
a52c-e4aa3bc56511/actions/com.mycompany.csp.core.approval.action.approve
--d @approve.json
```

For details regarding input and output for this request, see [Syntax for Approving a Submitted Machine Request](#).

If the command is successful, the HTTP status is 201 Created. If the command is not successful, the HTTP status is 204 No Content.

7.2 Work Item Service Examples for Approving a Machine Request

Syntax for each service example lists input parameters, output parameters, and curl commands.

■ [Syntax for Listing Work Items](#)

GET /api/workitems lists the unique IDs of all available work items.

■ [Syntax for Getting Work Item Details](#)

GET /api/workitems/{id} retrieves the details of a pending work item. You need these details to submit a completion request.

■ Syntax for Constructing a JSON File to Approve a Machine Request

You can specify a JSON file in your NECS Automation REST API command line input. For example, when you enter a command to approve a machine request, you can include the name of a JSON file that contains all the parameters required to approve the request and complete the work item.

■ Syntax for Approving a Submitted Machine Request

PUT /api/workitems/{id} approves a submitted work item request to complete the request. To construct the approval command, you add work item and work item form details to a JSON file, and call that JSON file from the command line. Use a template to correctly format the JSON file content.

■ Syntax for Updating Price Information

POST /api/blueprints/{id}/costs/upfront of the composition service, updates and displays price information for a deployment. The price of a deployment is based on which blueprint you request plus details of the specific request. For example, if the blueprint allows for a range of CPU, memory, or storage values, the price depends on the value requested.

7.3 Syntax for Listing Work Items

GET /api/workitems lists the unique IDs of all available work items.

Inputs

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|--|
| URL | https://\$NECS/workitem-service/api/workitems |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|-----------------|--|
| Links | Specifies an array of link objects, each of which contains the following parts: <ul style="list-style-type: none"> rel: Specifies the name of the link. <ul style="list-style-type: none"> Self refers to the object that was returned or requested. This property does not exist when you query for a single profile. First, Previous, Next, and Last refer to corresponding pages of pageable lists. Specifies the application or service that determines the other names. href: Specifies the URL that produces the result. |
| work itemNumber | Displays a reference number for the work item. |
| id | Specifies the unique identifier of this resource. |
| version | Displays the object version number. |

| Property | Description |
|--------------------------|--|
| assignees | Displays the list of work item assignees. |
| subTenantId | Optionally associates the work item with a specific business group granting users with management responsibilities over that business group permission to see the approval. |
| tenantId | Specifies the tenant ID for the work item. |
| callbackEntityId | Specifies the callback entity ID for the work item. |
| work itemType | Specifies the work item type for the work item. |
| completedDate | Specifies the date when the work item was completed. |
| assignedDate | Specifies the date when the work item was assigned. |
| createdDate | Specifies the created date of this instance. |
| assignedOrCompleted Date | Specifies the date to be displayed on UI. |
| formUrl | Specifies the URL from which the layout for this work item can be retrieved. |
| serviceId | Specifies the service ID that generated this work item instance. |
| work itemRequest | Specifies the corresponding work item request object. |
| status | Specifies the status of the work item. |
| completedBy | Specifies the principal ID of user who completed the work item. |
| availableActions | Contains a list of relevant work item actions. |
| Metadata | Specifies the paging-related data: <ul style="list-style-type: none"> ■ Size: Specifies the maximum number of rows per page. ■ totalElement: Specifies the number of rows returned. ■ totalPages: Specifies the total number of pages of data available. ■ Number: Specifies the current page number. ■ Offset: Specifies the number of rows skipped. |

Example: curl Command

The following example command retrieves all the available work item IDs.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-
service/api/workitems
```

Example: JSON Output

The following JSON output is returned based on the command input.

Note Price is referred to as cost in API commands and output.

```
{
  "links" : [ ],
  "content" : [ {
    "@type" : "WorkItem",
```

```

    "id" : "1755ef1a-d6f0-4901-9ecd-d03352ae4a05",
    "version" : 1,
    "workItemNumber" : 1,
    "assignees" : [ {
      "principalId" : "tony@example.mycompany.com",
      "principalType" : "USER"
    } ],
    "tenantId" : "MYCOMPANY",
    "callbackEntityId" : "1",
    "workItemType" : {
      "id" : "com.mycompany.cafe.samples.travel.workItem",
      "name" : "Workspace Assignment",
      "pluralizedName" : "Workspace Assignments",
      "description" : "Location Specific Workspace Assignment",
      "serviceTypeId" : "com.mycompany.cafe.samples.travel.api",
      "actions" : [ {
        "id" : "com.mycompany.cafe.samples.travel.workItem.complete",
        "name" : "Reserve Workspace",
        "stateName" : "Completed",
        "icon" : {
          "id" : "baa623db-0ca0-4db7-af41-9a301bc9e152",
          "name" : "Complete Action Icon",
          "contentType" : "image/png",
          "image" : null
        }
      } ], {
        "id" : "com.mycompany.cafe.samples.travel.workItem.cancel",
        "name" : "Workspace Unavailable",
        "stateName" : "Cancelled",
        "icon" : {
          "id" : "b03f994a-e1ec-4aae-8fae-e747ed680a5e",
          "name" : "Cancel Action Icon",
          "contentType" : "image/png",
          "image" : null
        }
      } ],
      "completeByEmail" : true,
      "commentsField" : null,
      "listView" : {
        "columns" : [ {
          "id" : "duration",
          "label" : "Duration",
          "description" : "The length of stay, measured in days.",
          "dataType" : {
            "type" : "primitive",
            "typeId" : "INTEGER"
          },
          "displayAdvice" : null,
          "state" : {
            "dependencies" : [ ],
            "facets" : [ ]
          },
          "filterable" : false,
          "sortable" : false,
          "isMultiValued" : false
        } ]
      }
    }
  ]
}

```

```

    }, {
      "id" : "location",
      "label" : "Destination",
      "description" : "The destination to which travel is being requested.",
      "dataType" : {
        "type" : "ref",
        "componentTypeId" : null,
        "componentId" : null,
        "classId" : "location",
        "typeFilter" : null,
        "label" : null
      },
      "displayAdvice" : null,
      "state" : {
        "dependencies" : [ ],
        "facets" : [ ]
      },
      "filterable" : false,
      "sortable" : false,
      "isMultiValued" : false
    }, {
      "id" : "arrivalDate",
      "label" : "Arrival Date",
      "description" : "The date of arrival at the destination",
      "dataType" : {
        "type" : "primitive",
        "typeId" : "DATE_TIME"
      },
      "displayAdvice" : null,
      "state" : {
        "dependencies" : [ ],
        "facets" : [ ]
      },
      "filterable" : false,
      "sortable" : false,
      "isMultiValued" : false
    } ],
    "defaultSequence" : [ "location", "arrivalDate", "duration" ]
  },
  "version" : 3,
  "forms" : {
    "workItemDetails" : {
      "type" : "external",
      "formId" : "travel.seating.task"
    },
    "workItemSubmission" : {
      "type" : "external",
      "formId" : "travel.seating.task"
    },
    "workItemNotification" : {
      "type" : "external",
      "formId" : "travel.itinerary.details"
    }
  }
},

```

```

-
-
-

"completedDate" : null,
"assignedDate" : "2014-02-20T23:55:31.600Z",
"createdDate" : "2014-02-20T23:55:31.600Z",
"assignedOrCompletedDate" : "2014-02-20T23:55:31.600Z",
"serviceId" : "2af18227-6a00-49e9-a76b-96de3ee767d2",
"workItemRequest" : {
  "itemId" : "531660fd-b540-4946-9917-38c023b61c02",
  "itemName" : "test travel 1",
  "itemDescription" : "test travel 1",
  "itemRequestor" : "tony@example.mycompany.com",
  "itemCost" : 0.0,
  "itemData" : {
    "entries" : [ {
      "key" : "requestLeaseTotal",
      "value" : {
        "type" : "money",
        "currencyCode" : null,
        "amount" : 1065.0
      }
    }, {
      "key" : "approvalId",
      "value" : {
        "type" : "string",
        "value" : "7a8b6054-1922-4f82-9266-245dffaa957c"
      }
    }, {
      "key" : "requestClassId",
      "value" : {
        "type" : "string",
        "value" : "request"
      }
    }, {
      "key" : "requestedFor",
      "value" : {
        "type" : "string",
        "value" : "tony@example.mycompany.com"
      }
    }, {
      "key" : "requestReasons"
    }, {
      "key" : "requestedItemName",
      "value" : {
        "type" : "string",
        "value" : "test travel 1"
      }
    }, {
      "key" : "requestInstanceId",
      "value" : {
        "type" : "string",

```

```

        "value" : "1cfe7177-74e3-4d68-a559-ea17587022ca"
    }, {
        "key" : "requestRef",
        "value" : {
            "type" : "string",
            "value" : "15"
        }
    }, {
        "key" : "requestedItemDescription",
        "value" : {
            "type" : "string",
            "value" : "test travel 1"
        }
    }, {
        "key" : "requestLeaseRate",
        "value" : {
            "type" : "moneyTimeRate",
            "cost" : {
                "type" : "money",
                "currencyCode" : null,
                "amount" : 213.0
            },
            "basis" : {
                "type" : "timeSpan",
                "unit" : "DAYS",
                "amount" : 1
            }
        }
    }, {
        "key" : "requestingServiceId",
        "value" : {
            "type" : "string",
            "value" : "f91d044a-04f9-4b96-8542-375e3e4e1dc1"
        }
    }, {
        "key" : "policy",
        "value" : {
            "type" : "string",
            "value" : "test travel approval policy"
        }
    }, {
        "key" : "phase",
        "value" : {
            "type" : "string",
            "value" : "PreApproval"
        }
    }, {
        "key" : "requestDescription",
        "value" : {
            "type" : "string",
            "value" : "t"
        }
    }, {
        "key" : "requestLease",

```

```

        "value" : {
            "type" : "timeSpan",
            "unit" : "DAYS",
            "amount" : 5
        }, {
            "key" : "requestedBy",
            "value" : {
                "type" : "string",
                "value" : "tony@example.mycompany.com"
            }
        } ]
    },
    "status" : "Active",
    "availableActions" : [ ]
} ],
"metadata":{
    "size" : 20,
    "totalElements" : 7,
    "totalPages" : 1,
    "number" : 1,
    "offset" : 0
}
}

```

7.4 Syntax for Getting Work Item Details

GET /api/workitems/{id} retrieves the details of a pending work item. You need these details to submit a completion request.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-------------|--|
| URL | https://\$NECS/workitem-service/api/workitems/workitem_ID |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| workitem_ID | Specifies the unique identifier of a work item. See Syntax for Listing Work Items . |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|--------------------------|---|
| Links | <p>Specifies an array of link objects, each of which contains the following parts:</p> <ul style="list-style-type: none"> ■ rel: Specifies the name of the link. <ul style="list-style-type: none"> ■ Self refers to the object that was returned or requested. This property does not exist when you query for a single profile. ■ First, Previous, Next, and Last refer to corresponding pages of pageable lists. ■ Specifies the application or service that determines the other names. ■ href: Specifies the URL that produces the result. |
| work itemNumber | Displays a reference number for the work item. |
| id | Specifies the unique identifier of this resource. |
| version | Displays the object version number. |
| assignees | Displays the list of work item assignees. |
| subTenantId | Optionally associates the work item with a specific business group granting users with management responsibilities over that business group permission to see the approval. |
| tenantId | Specifies the tenant ID for the work item. |
| callbackEntityId | Specifies the callback entity ID for the work item. |
| work itemType | Specifies the work item type for the work item. |
| completedDate | Specifies the date when the work item was completed. |
| assignedDate | Specifies the date when the work item was assigned. |
| createdDate | Specifies the created date of this instance. |
| assignedOrCompleted Date | Specifies the date to be displayed on UI. |
| formUrl | Specifies the URL from which the layout for this work item can be retrieved. |
| serviceId | Specifies the service ID that generated this work item instance. |
| work itemRequest | Specifies the corresponding work item request object. |
| status | Specifies the status of the work item. |
| completedBy | Specifies the principal ID of user who completed the work item. |
| availableActions | Contains a list of relevant work item actions. |
| Metadata | <p>Specifies the paging-related data:</p> <ul style="list-style-type: none"> ■ Size: Specifies the maximum number of rows per page. ■ totalElement: Specifies the number of rows returned. ■ totalPages: Specifies the total number of pages of data available. ■ Number: Specifies the current page number. ■ Offset: Specifies the number of rows skipped. |

Example: curl Command

The following example command retrieves the necessary details for the specified work item.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-service/api/workitems/5e3e9519-78ea-4409-a52c-e4aa3bc56511
```

Example: JSON Output

The following JSON output is returned based on the command input.

Note Price is referred to as cost in API commands and output.

To view the contents of a JSON output file, for example `workItemDetails.json`, use the `!` command with `more` in UNIX or `type` in Windows.

- (UNIX) `vcac-shell> ! more workItemDetails.json`
- (Windows) `vcac-shell> ! CMD /C type workItemDetails.json`

```
vcac-shell> ! more workItemDetails.json
{
  "id" : "5e3e9519-78ea-4409-a52c-e4aa3bc56511",
  "version" : 0,
  "workItemNumber" : 8,
  "assignees" : [ {
    "principalId" : "tony@example.mycompany.com",
    "principalType" : "USER"
  } ],
  "subTenantId" : "eab762cb-6e75-4379-83ef-171a71c9f00e",
  "tenantId" : "MYCOMPANY",
  "callbackEntityId" : "069dc3ce-a260-4d6a-b191-683141c994c0",
  "workItemType" : {
    "id" : "com.mycompany.csp.core.approval.workitem.request",
    "name" : "Approval",
    "pluralizedName" : "Approvals",
    "description" : "",
    "serviceTypeId" : "com.mycompany.csp.core.cafe.approvals",
    "actions" : [ {
      "id" : "com.mycompany.csp.core.approval.action.approve",
      "name" : "Approve",
      "stateName" : "Approved",
      "icon" : {
        "id" : "c192b6a7-5b35-4a3b-8593-107ffcf8c3a8",
        "name" : "approved.png",
        "contentType" : "image/png",
        "image" : null
      }
    } ],
    {
      "id" : "com.mycompany.csp.core.approval.action.reject",
      "name" : "Reject",
```

```

    "stateName" : "Rejected",
    "icon" : {
      "id" : "61c6da67-1164-421d-b575-10a245c89e10",
      "name" : "rejected.png",
      "contentType" : "image/png",
      "image" : null
    }
  } ],
  "completeByEmail" : true,
  "commentsField" : "businessJustification",
  "listView" : {
    "columns" : [ {
      "id" : "requestedItemName",
      "label" : "Requested Item",
      "description" : "",
      "dataType" : {
        "type" : "primitive",
        "typeId" : "STRING"
      },
      "displayAdvice" : null,
      "state" : {
        "dependencies" : [ ],
        "facets" : [ ]
      },
      "filterable" : false,
      "sortable" : false,
      "isMultiValued" : false
    },
    -
    -
    -

    {
      "id" : "requestLease",
      "label" : "Lease",
      "description" : "",
      "dataType" : {
        "type" : "primitive",
        "typeId" : "TIME_SPAN"
      },
      "displayAdvice" : null,
      "state" : {
        "dependencies" : [ ],
        "facets" : [ ]
      },
      "filterable" : false,
      "sortable" : false,
      "isMultiValued" : false
    } ],
    "defaultSequence" : [ "requestRef", "requestedItemName", "requestedFor", "requestLease",
"requestLeaseRate", "requestLeaseTotal" ]
  },
  "version" : 1,
  "forms" : {

```

```

    "workItemDetails" : {
      "type" : "external",
      "formId" : "approval.details"
    },
    "workItemSubmission" : {
      "type" : "external",
      "formId" : "approval.submission"
    },
    "workItemNotification" : {
      "type" : "external",
      "formId" : "approval.notification"
    }
  },
  "completedDate" : null,
  "assignedDate" : "2014-02-25T01:26:07.153Z",
  "createdDate" : "2014-02-25T01:26:07.153Z",
  "assignedOrCompletedDate" : "2014-02-25T01:26:07.153Z",
  "serviceld" : "2af18227-6a00-49e9-a76b-96de3ee767d2",
  "workItemRequest" : {
    "itemId" : "069dc3ce-a260-4d6a-b191-683141c994c0",
    "itemName" : "test-blueprint",
    "itemDescription" : "",
    "itemRequestor" : "fritz@example.mycompany.com",
    "itemCost" : 0.0,
    "itemData" : {
      "entries" : [ {
        "key" : "requestLeaseTotal"
      }, {
        "key" : "approvalId",
        "value" : {
          "type" : "string",
          "value" : "469c11ae-ed27-4790-baf1-c6839f35d474"
        }
      }, {
        "key" : "requestClassId",
        "value" : {
          "type" : "string",
          "value" : "request"
        }
      }, {
        "key" : "requestedFor",
        "value" : {
          "type" : "string",
          "value" : "fritz@example.mycompany.com"
        }
      }, {
        "key" : "requestReasons",
        "value" : {
          "type" : "string",
          "value" : ""
        }
      }, {
        "key" : "requestedItemName",
        "value" : {

```

```

        "type" : "string",
        "value" : "test-blueprint"
    }

    -
    -
    -

    }, {
        "key" : "requestLease"
    }, {
        "key" : "requestedBy",
        "value" : {
            "type" : "string",
            "value" : "fritz@example.mycompany.com"
        }
    } ]
}
},
"status" : "Active",
"availableActions" : [ ]
}

```

7.5 Syntax for Constructing a JSON File to Approve a Machine Request

You can specify a JSON file in your NECS Automation REST API command line input. For example, when you enter a command to approve a machine request, you can include the name of a JSON file that contains all the parameters required to approve the request and complete the work item.

Template JSON File Values

Copy the following template to start constructing a properly formatted JSON file in a text editor. Replace the highlighted values with your obtained work item details. After you create the JSON file, you can include it, or its contents, when you approve a submitted machine request. See [Syntax for Approving a Submitted Machine Request](#).

```

{
  "formData": {
    "entries": [
      {
        "key": "source-source-provider-Cafe.Shim.VirtualMachine.NumberOfInstances",
        "value": {
          "type": "integer",
          "value": 1
        }
      },
      {
        "key": "source-source-provider-VirtualMachine.Memory.Size",
        "value": {
          "type": "integer",
          "value": 512
        }
      }
    ]
  }
}

```

```

    }
  },
  {
    "key": "source-source-provider-VirtualMachine.CPU.Count",
    "value": {
      "type": "integer",
      "value": 1
    }
  },
  {
    "key": "source-businessJustification",
    "value": {
      "type": "string",
      "value": "solves abx request"
    }
  },
  {
    "key": "source-source-provider-VirtualMachine.LeaseDays",
    "value": {
      "type": "integer",
      "value": 0
    }
  }
]
},
"workItemId": "5e3e9519-78ea-4409-a52c-e4aa3bc56511",
"workItemActionId": "com.mycompany.csp.core.approval.action.approve"
}

```

Certain parameters are available to use in the JSON template.

Table 5-1. JSON Template Value Table

| JSON File Parameter Name | Description of Value |
|---|---|
| workItemId | Specifies the value of the corresponding work item ID obtained from the work item list. |
| source-source-provider-Cafe.Shim.VirtualMachine.NumberOfInstances value | Specifies the number of instances requested. |
| source-source-provider-VirtualMachine.Memory.Size | Specifies the amount of memory requested in GB. |
| source-source-provider-VirtualMachine.CPU.Count | Specifies the number of CPUs requested. |
| source-businessJustification | Specifies the text description of reason for request. |
| source-source-provider-VirtualMachine.LeaseDays | Specifies the number of days to lease. |
| workItemActionId | <p>To approve a request, include the approve statement, for example com.mycompany.csp.core.approval.action.approve..</p> <p>To reject a request, include the reject statement, for example com.mycompany.csp.core.approval.action.reject.</p> |

Example: JSON Input File

Use the following JSON input file sample when constructing a file.

```
{
  "@type": "CatalogItemRequest",
  "catalogItemRef": {
    "id": "65fbca06-a28e-46f3-bced-c6e5fb3a66f9"
  },
  "organization": {
    "tenantRef": "MYCOMPANY",
    "subtenantRef": "cccd7a7e-5283-416b-beb0-45eb4e924dcb"
  },
  "requestedFor": "fritz@example.mycompany.com",
  "state": "SUBMITTED",
  "requestNumber": 0,
  "requestData": {
    "entries": [{
      "key": "provider-blueprintId",
      "value": {
        "type": "string",
        "value": "e16edcf9-6a10-4bc7-98e2-a33361aeb857"
      }
    },
    {
      "key": "provider-provisioningGroupId",
      "value": {
        "type": "string",
        "value": "cccd7a7e-5283-416b-beb0-45eb4e924dcb"
      }
    },
    {
      "key": "requestedFor",
      "value": {
        "type": "string",
        "value": "fritz@example.mycompany.com"
      }
    },
    {
      "key": "provider-VirtualMachine.CPU.Count",
      "value": {
        "type": "integer",
        "value": 1
      }
    },
    {
      "key": "provider-VirtualMachine.Memory.Size",
      "value": {
        "type": "integer",
        "value": 1024
      }
    },
    {
      "key": "provider-VirtualMachine.LeaseDays",
```

```

        "value": {
            "type": "integer",
            "value": 30
        },
        {
            "key": "provider-Notes",
            "value": {
                "type": "string",
                "value": "MYCOMPANY machine"
            }
        },
        {
            "key": "provider-VirtualMachine.Disk0.Size",
            "value": {
                "type": "string",
                "value": "1"
            }
        },
        {
            "key": "provider-VirtualMachine.Disk0.Letter",
            "value": {
                "type": "string",
                "value": "C"
            }
        },
        {
            "key": "provider-VirtualMachine.Disk0.Label",
            "value": {
                "type": "string",
                "value": "main"
            }
        }
    ]
}

```

7.6 Syntax for Approving a Submitted Product Request

PUT /api/workitems/{id} approves a submitted work item request to complete the request. To construct the approval command, you add work item and work item form details to a JSON file, and call that JSON file from the command line. Use a template to correctly format the JSON file content.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|---|
| URL | https://\$NECS/workitem-service/api/workitems/workitem_ID |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |

| Parameter | Description |
|--------------------|---|
| <i>\$token</i> | Specifies a valid HTTP bearer token with necessary credentials. |
| <i>workitem_ID</i> | Specifies the unique identifier of a work item. See Syntax for Listing Work Items . |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|--------------------------|--|
| Links | Specifies an array of link objects, each of which contains the following parts: <ul style="list-style-type: none"> ■ rel: Specifies the name of the link. <ul style="list-style-type: none"> ■ Self refers to the object that was returned or requested. This property does not exist when you query for a single profile. ■ First, Previous, Next, and Last refer to corresponding pages of pageable lists. ■ Specifies the application or service that determines the other names. ■ href: Specifies the URL that produces the result. |
| work itemNumber | Displays a reference number for the work item. |
| id | Specifies the unique identifier of this resource. |
| version | Displays the object version number. |
| assignees | Displays the list of work item assignees. |
| subTenantId | Optionally associates the work item with a specific business group granting users with management responsibilities over that business group permission to see the approval. |
| tenantId | Specifies the tenant ID for the work item. |
| callbackEntityId | Specifies the callback entity ID for the work item. |
| work itemType | Specifies the work item type for the work item. |
| completedDate | Specifies the date when the work item was completed. |
| assignedDate | Specifies the date when the work item was assigned. |
| createdDate | Specifies the created date of this instance. |
| assignedOrCompleted Date | Specifies the date to be displayed on UI. |
| formUrl | Specifies the URL from which the layout for this work item can be retrieved. |
| serviceId | Specifies the service ID that generated this work item instance. |
| work itemRequest | Specifies the corresponding work item request object. |
| status | Specifies the status of the work item. |
| completedBy | Specifies the principal ID of user who completed the work item. |

| Property | Description |
|------------------|--|
| availableActions | Contains a list of relevant work item actions. |
| Metadata | Specifies the paging-related data: <ul style="list-style-type: none"> Size: Specifies the maximum number of rows per page. totalElement: Specifies the number of rows returned. totalPages: Specifies the total number of pages of data available. Number: Specifies the current page number. Offset: Specifies the number of rows skipped. |

Example: Example: curl Command

Approve a submitted machine request by specifying its work item ID and using a JSON file named approve.json to pass arguments to the command line.

```
curl -X PUT --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/workitem-service/api/workitems/5e3e9519-78ea-4409-
a52c-e4aa3bc56511/actions/com.mycompany.csp.core.approval.action.approve
--d @approve.json
```

Error Conditions

If the same request is submitted a second time, the following error response is received:

```
Command failed [Rest Error]: {Status code: 400}, {Error code: 12005} ,
{Error Source: null}, {Error Msg: Work item 5e3e9519-78ea-4409-a52c-e4aa3bc56511
is in COMPLETED state. Requested operation cannot be performed.}, {System Msg:
Work item 5e3e9519-78ea-4409-a52c-e4aa3bc56511 is in COMPLETED state. Requested
operation cannot be performed.}
```

If a user who is not authorized to approve the request submits the request, the following error response is received:

```
Command failed [Rest Error]: {Status code: 400}, {Error code: 12017} ,
{Error Source: null}, {Error Msg: User fritz@example.mycompany.com not authorized to
complete work item with ID 5e3e9519-78ea-4409-a52c-e4aa3bc56511.}, {System Msg:
User fritz@example.mycompany.com not authorized to complete Work item with id
5e3e9519-78ea-4409-a52c-e4aa3bc56511.}
```

8.0 Listing Provisioned Resources

You use the catalog service to list provisioned resources.

The catalog service is designed to be used by consumers and providers of the service catalog. For example, a consumer might want to list resources provisioned by a provider. The consumer can also list the resources in multiple ways.

Each example for this use case lists a curl command with respective JSON response, plus input and output parameters. The same set of prerequisites applies to each example.

This chapter includes the following topics:

- [Prerequisites for Listing Provisioned Resources](#)
- [Display Your Provisioned Resources Example](#)
- [Display Provisioned Resources by Resource Type Example](#)
- [Display All Available Resource Types Example](#)
- [Display Provisioned Resources by Business Groups You Manage Example](#)
- [View Machine Details Example](#)

8.1 Prerequisites for Listing Provisioned Resources

Satisfy the following conditions before performing any tasks for this use case.

- Log in to NECS Automation as a **business group manager**.
- Verify that the appliance name and fully qualified domain name of the NECS Automation instance are available.
- Verify that you have a valid HTTP bearer token that matches your login credentials. See [2 REST API Authentication](#).

8.2 Display Your Provisioned Resources Example

GET /api/consumer/resources/{id} displays a list of all the provisioned resources that you own.

curl Command

The following example displays all applicable provisioned resources.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/resources/?page=1&limit=n&$orderby=name
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "links" : [ {
    "@type" : "link",
    "rel" : "next",
    "href" : "https://NECS152-009-067.mycompany.com/catalog-service/api/consumer/resources/?
page=2&limit=1"
  } ],
  "content" : [ {
    "@type" : "ConsumerResource",
    "id" : "c24e8c75-c201-489c-b51c-8d7009c23563",
    "iconId" : "Travel_100.png",
    "resourceTypeRef" : {
      "id" : "com.mycompany.mystuff.samples.travel.packageType",
      "label" : "Reservation"
    },
    "name" : "example",
    "description" : "asd",
    "status" : "ACTIVE",
    "catalogResource" : {
      "id" : "6fddafcd-bc3d-4753-8a2a-5fa3f78a5a90",
      "label" : "example"
    },
    "requestId" : "55e7fcf3-4c77-4b11-a442-1f282333ac91",
    "providerBinding" : {
      "bindingId" : "1",
      "providerRef" : {
        "id" : "f60f5d1e-d6e9-4d98-9c48-f70a3e405346",
        "label" : "travel-service"
      }
    }
  },
  ...
}
```

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|---|
| URL | https://\$NECS/catalog-service/api/consumer/resources/ |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| \$page | Specifies a page number. |
| \$limit | Specifies the number of entries to display on a page. Maximum value is 5000. If not specified, defaults to 20. For information regarding limits to the number of elements displayed, see Example: Retrieve 10,000 Resources Ordered By Name . |
| \$orderby | Specifies how to order multiple comma-separated properties sorted in ascending or descending order. Values include: <ul style="list-style-type: none"> ■ \$orderby=id ■ \$orderby=name ■ \$orderby=dateCreated ■ \$orderby=lastUpdated ■ \$orderby=status ■ \$orderby=description |
| \$top | Specifies the number of returned entries from the top of the response (total number per page in relation to skip). |
| \$skip | Specifies the number of entries to skip. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|-----------------|--|
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| resourceTypeRef | Specifies the resource type. |
| name | Specifies the resource name. |
| description | Specifies the resource description. |
| status | Specifies the resource status. |
| catalogItem | Specifies the catalog item that defines the service this resource is based on. |
| requestId | Specifies the request ID that provisioned this resource. |
| providerBinding | Specifies the provider binding. |
| owners | Species the owners of this resource. |
| organization | Specifies the subtenant or tenant that owns this resource. |
| dateCreated | Specifies the data and time at which the resource was created. |

| Property | Description |
|--------------------|--|
| lastUpdated | Specifies the date and time at which the resource was most recently modified. |
| hasLease | Returns true if the resource is subject to a lease. |
| lease | Displays the resource's current lease as start and end time stamps. |
| leaseForDisplay | Specifies the resource's current lease, #getLease, with time units synchronized with #getCosts. |
| hasCosts | Returns true if the resource is subject to per-time price. |
| costs | Displays an optional rate of the price charges for the resource. This parameter is deprecated. |
| costToDate | Displays an optional rate of the current price charges for the resource. This parameter is deprecated. |
| totalCost | Displays an optional rate of the price charges for the entire lease period. This parameter is deprecated. |
| expenseMonthToDate | The expense of the resource from the beginning of the month to the current date. This value is updated daily by NECS Business for Cloud. |
| parentResourceRef | Displays the parent of this resource. |
| childResources | Displays the children of this resource. |
| operations | Specifies the sequence of available operations that can be performed on this resource. |
| forms | Specifies the forms used to render this resource. |
| resourceData | Displays the extended provider-defined properties of the resource. |

Example: Retrieve 10,000 Resources Ordered By Name

Since the catalog service limits the number of elements that can be retrieved with a single API call to 5000, retrieving 10,000 resources requires two calls. The first call displays the first 5000 elements and the second call displays the second 5000 elements. You can make the two calls by specifying either the page and limit values or the skip and top values.

Specifying page and limit values, you make the following two calls.

```
curl --insecure -H "Content-Type: application/json" -H "Accept: application/json"
-H "Authorization: $AUTH" "https://$NECS/catalog-service/api/consumer/resources/?page=1&limit=5000&
$orderby=name"
curl --insecure -H "Content-Type: application/json" -H "Accept: application/json"
-H "Authorization: $AUTH" "https://$NECS/catalog-service/api/consumer/resources/?page=2&limit=5000&
$orderby=name"
```

Specifying skip and top values, you make the following two calls.

```
curl --insecure -H "Content-Type: application/json" -H "Accept: application/json"
-H "Authorization: $AUTH" "https://$NECS/catalog-service/api/consumer/resources/?$skip=0&$top=5000&
$orderby=name"
curl --insecure -H "Content-Type: application/json" -H "Accept: application/json"
-H "Authorization: $AUTH" "https://$NECS/catalog-service/api/consumer/resources/?$skip=5000&
$top=5000&$orderby=name"
```

If both page and limit values and skip and top values are specified, the skip and top values take priority.

8.3 Display Provisioned Resources by Resource Type Example

GET /api/consumer/resourceTypes/{id} displays a list of the provisioned resources that you own filtered by machine resource type.

curl Command

The following example displays the provisioned resources by resource type.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/resourceTypes/Infrastructure.Machine/?page=1&limit=n&
$orderby=id
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "links" : [ ],
  "content" : [ {
    "@type" : "ConsumerResource",
    "id" : "3bfde906-81b9-44c3-8c2d-07d2c9768168",
    "iconId" : "cafe_default_icon_genericCatalogResource",
    "resourceTypeRef" : {
      "id" : "Infrastructure.Virtual",
      "label" : "Virtual Machine"
    },
    },
    "name" : "test2",
    "description" : null,
    "status" : "ACTIVE",
    "catalogResource" : {
      "id" : "e2f397be-72ad-4ec4-a688-c017560fa1a3",
      "label" : "test-blueprint"
    },
    },
    "requestId" : "b013d2fa-4ba4-416c-b46b-98bb8cc7b076",
    "providerBinding" : {
      "bindingId" : "8a4581a0-84f9-4e80-9af6-75d79633e382",
      "providerRef" : {
        "id" : "6918cd49-b737-467f-94bf-d14d52c78fba",
        "label" : "iaas-service"
      }
    },
    },
    "owners" : [ {
      "tenantName" : "MYCOMPANY",
      "ref" : "fritz@example.mycompany.com",
      "type" : "USER",
      "value" : "Fritz Arbeiter"
    } ],
    "organization" : {
      "tenantRef" : "MYCOMPANY",
```

```

    "tenantLabel" : "QETenant",
    "subtenantRef" : "eab762cb-6e75-4379-83ef-171a71c9f00e",
    "subtenantLabel" : "MyTestAgentBusinessGroup"
  },
  ...
}

```

The output includes the following highlighted items:

- Resource ID. **3bfde906-81b9-44c3-8c2d-07d2c9768168** corresponds to a provisioned machine owned by the logged-in user. The resource IDs are used in requests to retrieve the details for the corresponding machines.
- subtenantRef ID. **eab762cb-6e75-4379-83ef-171a71c9f00e** corresponds to the business group of the logged-in user. If the user who is logged-in is also the manager of the business group, the subtenantRef ID is used to get resources from all business groups that the user manages.

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|---|
| URL | https://\$NECS/catalog-service/api/consumer/resourceTypes |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| page | Specifies a page number. |
| limit | Specifies the number of entries to display on a page. Maximum value is 5000. If not specified, defaults to 20. For information regarding limits to the number of elements displayed, see Example: Retrieve 10,000 Resources Ordered By Name . |
| \$orderby | Specifies how to order multiple comma-separated properties sorted in ascending or descending order. Values include: <ul style="list-style-type: none"> ■ \$orderby=id ■ \$orderby=name ■ \$orderby=dateCreated ■ \$orderby=lastUpdated ■ \$orderby=status ■ \$orderby=description |
| top | Specifies the number of returned entries from the top of the response (total number per page in relation to skip). |
| skip | Specifies the number of entries to skip. |

Filter by the following resource types:

- Infrastructure.Machine
- Infrastructure.AppService

- Infrastructure.Cloud
- Infrastructure.Physical
- Infrastructure.vApp
- Infrastructure.Virtual

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|--------------------|--|
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| resourceTypeRef | Specifies the resource type. |
| name | Specifies the resource name. |
| description | Specifies the resource description. |
| status | Specifies the resource status. |
| catalogItem | Specifies the catalog item that defines the service this resource is based on. |
| requestId | Specifies the request ID that provisioned this resource. |
| providerBinding | Specifies the provider binding. |
| owners | Species the owners of this resource. |
| organization | Specifies the subtenant or tenant that owns this resource. |
| dateCreated | Specifies the data and time at which the resource was created. |
| lastUpdated | Specifies the date and time at which the resource was most recently modified. |
| hasLease | Returns true if the resource is subject to a lease. |
| lease | Displays the resource's current lease as start and end time stamps. |
| leaseForDisplay | Specifies the resource's current lease, #getLease, with time units synchronized with #getCosts. |
| hasCosts | Returns true if the resource is subject to per-time price. |
| costs | Displays an optional rate of the price charges for the resource. This parameter is deprecated. |
| costToDate | Displays an optional rate of the current price charges for the resource. This parameter is deprecated. |
| totalCost | Displays an optional rate of the price charges for the entire lease period. This parameter is deprecated. |
| expenseMonthToDate | The expense of the resource from the beginning of the month to the current date. This value is updated daily by NECS Business for Cloud. |
| parentResourceRef | Displays the parent of this resource. |
| childResources | Displays the children of this resource. |
| operations | Specifies the sequence of available operations that can be performed on this resource. |
| forms | Specifies the forms used to render this resource. |
| resourceData | Displays the extended provider-defined properties of the resource. |

8.4 Display All Available Resource Types Example

GET /api/consumer/resourcesTypes displays all the resource types that are available on the system.

curl Command

The following example displays all available resource types.

```
curl --insecure -H "Content-Type: application/json" -H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/resourceTypes
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "links" : [ ],
  "content" : [ {
    "@type" : "ResourceType",
    "id" : "Infrastructure.Machine",
    "name" : "Machine",
    "pluralizedName" : "Machines",
    "description" : "The common parent type for all types of machines",
    "primary" : true,
    "schema" : {
      "classId" : "Infrastructure.Machine.Schema",
      "typeFilter" : null
    },
    "forms" : {
      "catalogResourceInfoHidden" : true,
      "details" : {
        "type" : "extension",
        "extensionId" : "csp.places.iaas.resource.details",
        "extensionPointId" : null
      }
    }
  }
]
```

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|--|
| URL | https://\$NECS/catalog-service/api/consumer/resourceTypes |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|--------------------|--|
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| resourceTypeRef | Specifies the resource type. |
| name | Specifies the resource name. |
| description | Specifies the resource description. |
| status | Specifies the resource status. |
| catalogItem | Specifies the catalog item that defines the service this resource is based on. |
| requestId | Specifies the request ID that provisioned this resource. |
| providerBinding | Specifies the provider binding. |
| owners | Species the owners of this resource. |
| organization | Specifies the subtenant or tenant that owns this resource. |
| dateCreated | Specifies the data and time at which the resource was created. |
| lastUpdated | Specifies the date and time at which the resource was most recently modified. |
| hasLease | Returns true if the resource is subject to a lease. |
| lease | Displays the resource's current lease as start and end time stamps. |
| leaseForDisplay | Specifies the resource's current lease, #getLease, with time units synchronized with #getCosts. |
| hasCosts | Returns true if the resource is subject to per-time price. |
| costs | Displays an optional rate of the price charges for the resource. This parameter is deprecated. |
| costToDate | Displays an optional rate of the current price charges for the resource. This parameter is deprecated. |
| totalCost | Displays an optional rate of the price charges for the entire lease period. This parameter is deprecated. |
| expenseMonthToDate | The expense of the resource from the beginning of the month to the current date. This value is updated daily by NECS Business for Cloud. |
| parentResourceRef | Displays the parent of this resource. |
| childResources | Displays the children of this resource. |
| operations | Specifies the sequence of available operations that can be performed on this resource. |
| forms | Specifies the forms used to render this resource. |
| resourceData | Displays the extended provider-defined properties of the resource. |

Example: curl Command

The following example command displays all available resource types.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/resourceTypes
```

Example: JSON Output

The following JSON output is returned based on the command input.

```
{
  "links" : [ ],
  "content" : [ {
    "@type" : "ResourceType",
    "id" : "Infrastructure.Machine",
    "name" : "Machine",
    "pluralizedName" : "Machines",
    "description" : "The common parent type for all types of machines",
    "primary" : true,
    "schema" : {
      "classId" : "Infrastructure.Machine.Schema",
      "typeFilter" : null
    },
    "forms" : {
      "catalogResourceInfoHidden" : true,
      "details" : {
        "type" : "extension",
        "extensionId" : "csp.places.iaas.resource.details",
        "extensionPointId" : null
      }
    }
  }
]
```

8.5 View Resource Details Example

GET /api/consumer/requests/{id}/resourceViews displays the machine details for a provisioned machine.

curl Command

The following example displays machine details for a provisioned machine with *resourceID=7aaf9baf-aa4e-47c4-997b-edd7c7983a5b*.

```
curl --insecure -H "Content-Type: application/json"
-H "Authorization: Bearer $token"
http://$NECS/catalog-service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b/resourceViews
```

JSON Output

The following JSON output is returned based on the command input.

```
{
  "links": [],
  "content": [
    {
      "@type": "CatalogResourceView",
      "links": [
        {
          "@type": "link",
          "rel": "GET: Catalog Item",
          "href": "https://$NECS/catalog-
service/api/consumer/entitledCatalogItemViews/7c8275d6-1bd6-452a-97c4-d6c053e4baa4"
        },
        {
          "@type": "link",
          "rel": "GET: Request",
          "href": "https://$NECS/catalog-
service/api/consumer/requests/7aaf9baf- aa4e-47c4-997b-edd7c7983a5b"
        },
        {
          "@type": "link",
          "rel": "GET Template:
{com.NRB.csp.component.cafe.composition@resource.action.deployment.destroy.name}",
          "href": "https://$NECS/catalog-service/api/consumer/resources/c4d3db3e-e397-44ff-
a1c9-0ecebdba12f4/actions/416e6bb1-3357-448b-8396-e268d5f7343b/requests/template"
        },
        {
          "@type": "link",
          "rel": "POST:
{com.nrb.csp.component.cafe.composition@resource.action.deployment.destroy.name}",
          "href": "https://$NECS/catalog-service/api/consumer/resources/c4d3db3e-e397-44ff-
a1c9-0ecebdba12f4/actions/416e6bb1-3357-448b-8396-e268d5f7343b/requests"
        },
        {

```

```

        "@type": "link",
        "rel": "GET: Child Resources",
        "href": "https://$NECS/catalog-service/api/consumer/resourceViews?
managedOnly=false&withExtendedData=true&withOperations=true&%24filter=parentResource%20eq
%20%27c4d3db3e-e397-44ff-a1c9-0ecebdba12f4%27"
    }
],
    "resourceId": "c4d3db3e-e397-44ff-a1c9-0ecebdba12f4",
    "iconId": "cafe_default_icon_genericCatalogItem",
    "name": "Linux-80813151",
    "description": null,
    "status": null,
    "catalogItemId": "7c8275d6-1bd6-452a-97c4-d6c053e4baa4",
    "catalogItemLabel": "Linux",
    "requestId": "7aaf9baf-aa4e-47c4-997b-edd7c7983a5b",
    "resourceType":
"{com.nrb.csp.component.cafe.composition@resource.type.deployment.name}",
    "owners": [
        "Connie Summers"
    ],
    "businessGroupId": "c0683388-6db2-4cb5-9033-b24d15ad3766",
    "tenantId": "mycompany",
    "dateCreated": "2015-07-29T13:51:36.368Z",
    "lastUpdated": "2015-07-29T13:55:35.963Z",
    "lease": null,
    "costs": null,
    "costToDate": null,
    "totalCost": null,
    "parentResourceId": null,
    "hasChildren": true,
    "data": {}
}
],
"metadata":{
    "size": 20,
    "totalElements": 1,
    "totalPages": 1,
    "number": 1,
    "offset": 0
}
}

```

8.6 Using the API to Get Deployment Details

To view deployed machine details, append `/resourceViews` to the request details URI that you generated when you retrieved request details.

```
http://$NECS/catalog-service/api/consumer/requests/$requestId/resourceViews
```

See [Syntax for Viewing Details of a Machine Request](#).

In addition to general information about the provisioned deployment such as its name, description, and ID, the response contains additional HATEOAS links that enable you to obtain additional details and information.

Table 6-1. HATEOAS Link Functions as Defined by rel Field

| Link | Description |
|---|--|
| GET: Catalog Item | URI to get the catalog item details (as described in sections 3.2.1 and 3.2.2) from which this catalog item was provisioned. |
| GET: Request | URI to get the request details that provisioned this item. |
| GET: Template {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to get a template request for a specific action that you can perform on this resource. Typically, on a deployment the action will be Delete. |
| POST: {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to which to post the request to perform an action, based on the corresponding template. |
| GET: Child Resources | If the deployment contains child resources (nodes specified in the composite blueprint), this is the URI to get a list of the resourceViews for the children of this deployment. |

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|--------------|---|
| URL | https://\$NECS/catalog-service/api/consumer/resources/\$resourceId |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| \$resourceId | Specifies a resource ID. See Display Your Provisioned Resources Example to view all of your requests and search for a request ID. |
| managedOnly | If true, the returned requests are from the user's managed subtenants. |
| page | Specifies a page number. |
| limit | Specifies the number of entries to display on a page. |
| \$orderby | Specifies how to order multiple comma-separated properties sorted in ascending or descending order. |
| \$top | Specifies the number of returned entries from the top of the response (total number per page in relation to skip). |
| \$skip | Specifies the number of entries to skip. |
| filter | Contains a Boolean expression to determine if a particular entry is included in the response. |

Output

The command output contains property names and values based on the command input parameters.

| Property | Description |
|--------------------|--|
| id | Specifies the unique identifier of this resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| resourceTypeRef | Specifies the resource type. |
| name | Specifies the resource name. |
| description | Specifies the resource description. |
| status | Specifies the resource status. |
| catalogItem | Specifies the catalog item that defines the service this resource is based on. |
| requestId | Specifies the request ID that provisioned this resource. |
| providerBinding | Specifies the provider binding. |
| owners | Species the owners of this resource. |
| organization | Specifies the subtenant or tenant that owns this resource. |
| dateCreated | Specifies the data and time at which the resource was created. |
| lastUpdated | Specifies the date and time at which the resource was most recently modified. |
| hasLease | Returns true if the resource is subject to a lease. |
| lease | Displays the resource's current lease as start and end time stamps. |
| leaseForDisplay | Specifies the resource's current lease, #getLease, with time units synchronized with #getCosts. |
| hasCosts | Returns true if the resource is subject to per-time price. |
| costs | Displays an optional rate of the price charges for the resource. This parameter is deprecated. |
| costToDate | Displays an optional rate of the current price charges for the resource. This parameter is deprecated. |
| totalCost | Displays an optional rate of the price charges for the entire lease period. This parameter is deprecated. |
| expenseMonthToDate | The expense of the resource from the beginning of the month to the current date. This value is updated daily by NECS Business for Cloud. |
| parentResourceRef | Displays the parent of this resource. |
| childResources | Displays the children of this resource. |
| operations | Specifies the sequence of available operations that can be performed on this resource. |
| forms | Specifies the forms used to render this resource. |
| resourceData | Displays the extended provider-defined properties of the resource. |

9.0 Managing Provisioned Deployments

You use the catalog service to manage provisioned deployments.

The catalog service is designed to be used by consumers of the service catalog. For example, a consumer might want to list all provisioned resources then submit a request to power off a resource.

This chapter includes the following topics:

- [Manage Provisioned Deployments](#)
- [Power Off](#)
- [Change Lease](#)
- [Catalog Service Examples for Managing Provisioned Deployments](#)

9.1 Manage Provisioned Deployments

You use the catalog service to log in to NECS Automation and view information about provisioned resources.

Prerequisites

- Log in to NECS Automation as a **business group manager**.
- Verify that the appliance name and fully qualified domain name of the NECS Automation instance are available.
- Verify that you have a valid HTTP bearer token that matches your login credentials. See [REST API Authentication](#).

Procedure

- 1 Display a list of all provisioned resources.

```
$curl --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer $token"
http://$NECS/catalog-service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-
edd7c7983a5b/resourceViews
```

For details regarding input and output of this sample, see [Syntax for Getting Deployment Details](#).

- 2 Examine the response for the HATEOAS links that you need to obtain additional information about specific deployed resources.

- 3 Use the GET: Child Resources HATEOAS link to retrieve a list of child nodes of a deployment.

```
$curl --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer $token"
https:// $NECS/catalog-service/api/consumer/resourceViews?
managedOnly=false&withExtendedData=true&withOperations=true&%24filter=parentResource%20eq
%20%27c4d3db3e-e397-44ff-a1c9-0ecebdba12f4%27
```

For details regarding input and output of this sample, see [Syntax for Navigating to the Children of a Deployed Resource](#).

What to do next

Use the HATEOS links obtained from retrieving the list of child nodes to perform actions.

- See [Power Off](#).
- See [Change Lease](#).

9.2 Power Off

You use the catalog service to perform a power off action. For simple actions that require no user input, the process is straightforward.

This command leverages the links for the power off action from the command used in the [Syntax for Navigating to the Children of a Deployed Resource](#) example.

```
{
  "@type": "link",
  "rel": "GET Template: {...iaas.proxy.provider@resource.action.name.machine.PowerOff}",
  "href": "https://$NECS/api/consumer/resources/dd3...a4a/actions/02ba...e38/requests/template"
},
{
  "@type": "link",
  "rel": "POST: {com.nrb.iaas.proxy.provider@resource.action.name.machine.PowerOff}",
  "href": "https://$NECS/api/consumer/resources/dd3...a4a/actions/02b...e38/requests"
}
```

Procedure

- 1 Get the template for the resource action request.

```
$curl --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer $token"
https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-b5be-
b229453eca4a/actions/02bad06d-f92b-4cf8-b964-37bb5d57be38/requests/template
```

2 Examine the response.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Cache-Control: no-cache, no-store
Pragma: no-cache
Expires: Sat, 01 August 2015 23:04:50 GMT
Content-Type: application/json;charset=UTF-8
Date: Sat, 01 August 2015 13:04:50 GMT
{
  "type": "com.nrb.vcac.catalog.domain.request.CatalogResourceRequest", "resourceId":
  "dd37b7a1-829c-4773-b5be-b229453eca4a",
  "actionId": "02bad06d-f92b-4cf8-b964-37bb5d57be38",
  "description": null,
  "data": {
    "description": null,
    "reasons": null
  }
}
```

- 3 Edit the template as desired. The template is populated with default values. For example, you may want to provide custom values for the description and reasons.
- 4 Use a POST command to send the template without modification to the corresponding URI.

```
$curl --verbose --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer
$token"https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-b5be-
b229453eca4a/actions/02bad06d-f92b-4cf8-b964-37bb5d57be38/requests
{
  "type": "com.nrb.vcac.catalog.domain.request.CatalogResourceRequest", "resourceId":
  "dd37b7a1-829c-4773-b5be-b229453eca4a",
  "actionId": "02bad06d-f92b-4cf8-b964-37bb5d57be38",
  "description": null,
  "data": {
    "description": null,
    "reasons": null
  }
}
```

This POST command returns a response indicating success or failure. HTTP/1.1 201 CREATED indicates that the request was submitted successfully.

9.3 Catalog Service Examples for Managing Provisioned Deployments

Syntax for each service example lists input parameters, output parameters, and curl commands.

- [Syntax for Getting Deployment Details](#)

GET /api/consumer/requests/{id}/resourceViews retrieves resources provisioned by a given request.

- [Syntax for Navigating to the Children of a Deployed Resource](#)

GET /api/consumer/resourceViews retrieves a list of the child nodes of a deployment, including virtual machines, networks, and other objects you may have configured on the design canvas.

Syntax for Getting DeploymentDetails

GET /api/consumer/requests/{id}/resourceViews retrieves resources provisioned by a given request.

Accessing Links to ProvisionedItems

You can access links to provisioned items from a given request by appending /resourceViews to the request details URI. For instance, you can edit the example request URI from as follows:

```
http://$NECS/catalog-service/api/consumer/requests/$requestId/resourceViews
```

In addition to the general information about the provisioned deployment returned in the response, such as its name, description and ID, the response contains additional HATEOAS links.

Table 7-1. HATEOAS Link Deployment Details Functions

| Link | Description |
|--|---|
| GET: Catalog Item | URI to get the catalog item details from which this catalog item was provisioned. See Syntax for Viewing Details of a Machine Request . |
| GET: Request | URI to get the request details that provisioned this item. |
| GET:Template {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to get a template request for a specific action that you can perform on this resource. Typically, on a deployment, the action will be Delete. |

Table 7-1. HATEOAS Link Deployment Details Functions (Continued)

| Link | Description |
|---|---|
| POST: {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to which to post the request to perform an action, based on the corresponding template. |
| GET: Child Resources | If the deployment contains child resources, such as nodes specified in the composite blueprint, this is the URI to get a list of the resourceViews for the children of this deployment. |

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|-----------|--|
| URL | https://\$NECS/catalog-service/api/consumer/resources/\$resourceId |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| id | UUID of a request. |
| page | Specifies a page number. |
| limit | Specifies the number of entries to display on a page. |
| \$orderby | Specifies how to order multiple comma-separated properties sorted in ascending or descending order. |
| \$stop | Specifies the number of returned entries from the top of the response (total number per page in relation to skip). |
| \$skip | Specifies the number of entries to skip. |
| filter | Contains a Boolean expression to determine if a particular entry is included in the response. |

Output

The command output contains property names and values based on the command input parameters.

Note Price is referred to as cost in API commands and output.

Table 7-2. Output Parameters

| Property | Description |
|---------------|---|
| resourceId | The unique identifier of the resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| name | The user friendly name of the resource. |
| description | An extended user friendly description of the resource. |
| status | The status of the resource. For example, On, Off, etc. |
| catalogItemId | The identifier of the catalog item associated with this provisioned resource. |

Table 7-2. Output Parameters (Continued)

| Property | Description |
|--------------------|---|
| catalogItemLabel | The label of the catalog item associated with this provisioned resource. |
| requestId | The unique identifier of the request that created this provisioned resource. |
| businessGroupId | The unique identifier of the business group that owns this resource. |
| tenantId | The unique identifier of the tenant that owns this resource. |
| owners | The owner of this resource. |
| resourceType | The type identifier of this resource. For example, Virtual Machine. |
| parentResourceId | The unique identifier of the parent resource. Used for child resources of a multi-machine resource. |
| hasChildren | Returns true if this resource has child resources. Used if this is a multi-machine resource. |
| dateCreated | The date and time at which the resource was created. |
| lastUpdated | The date and time at which the resource was most recently modified. |
| lease | The current lease of the resource. |
| costs | An optional rate card of the prices and charges levied against the resource. This parameter is deprecated. |
| costToDate | An optional rate card of the existing prices and charges levied against the resource. This parameter is deprecated. |
| totalCost | An optional rate card of the prices and charges levied for the entire lease period. This parameter is deprecated. |
| expenseMonthToDate | The expense of the resource from the beginning of the month to the current date. |
| data | The extended, provider defined properties of the resource. |

Example Curl Command

This example retrieves all children of the resource with an ID of 7aaf9baf-aa4e-47c4-997b-edd7c7983a5b.

```
$curl --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer $token"
http://$NECS/catalog-service/api/consumer/requests/7aaf9baf-aa4e-47c4-997b-edd7c7983a5b/resourceViews
```

Example: JSON Output

```
{
  "links": [],
  "content": [
    {
      "@type": "CatalogResourceView",
      "links": [
        {
          "@type": "link",
          "rel": "GET: Catalog Item",
          "href": "https://$NECS/catalog-
service/api/consumer/entitledCatalogItemViews/7c8275d6-1bd6-452a-97c4-d6c053e4baa4"
```

```

        },
        {
            "@type": "link",
            "rel": "GET: Request",
            "href": "https://$NECS/catalog-service/api/consumer/requests/7aaf9baf-
aa4e-47c4-997b-edd7c7983a5b"
        },
        {
            "@type": "link",
            "rel": "GET Template:
{com.NRB.csp.component.cafe.composition@resource.action.deployment.destroy.name}",
            "href": "https://$NECS/catalog-service/api/consumer/resources/c4d3db3e-e397-44ff-
a1c9-0ecebdba12f4/actions/416e6bb1-3357-448b-8396-e268d5f7343b/requests/template"
        },
        {
            "@type": "link",
            "rel": "POST:
{com.NRB.csp.component.cafe.composition@resource.action.deployment.destroy.name}",
            "href": "https://$NECS/catalog-service/api/consumer/resources/c4d3db3e-e397-44ff-
a1c9-0ecebdba12f4/actions/416e6bb1-3357-448b-8396-e268d5f7343b/requests"
        },
        {
            "@type": "link",
            "rel": "GET: Child Resources",
            "href": "https://$NECS/catalog-service/api/consumer/resourceViews?
managedOnly=false&withExtendedData=true&withOperations=true&%24filter=parentResource%20eq
%20%27c4d3db3e-e397-44ff-a1c9-0ecebdba12f4%27"
        }
    ],
    "resourceId": "c4d3db3e-e397-44ff-a1c9-0ecebdba12f4",
    "iconId": "cafe_default_icon_genericCatalogItem",
    "name": "Linux-80813151",
    "description": null,
    "status": null,
    "catalogItemId": "7c8275d6-1bd6-452a-97c4-d6c053e4baa4",
    "catalogItemLabel": "Linux",
    "requestId": "7aaf9baf-aa4e-47c4-997b-edd7c7983a5b",
    "resourceType":
"{com.NRB.csp.component.cafe.composition@resource.type.deployment.name}", "owners": [
    "Connie Summers"
],
    "businessGroupId": "c0683388-6db2-4cb5-9033-b24d15ad3766",
    "tenantId": "mycompany",
    "dateCreated": "2015-07-29T13:51:36.368Z",
    "lastUpdated": "2015-07-29T13:55:35.963Z",
    "lease": null,
    "costs": null,
    "costToDate": null,
    "totalCost": null,
    "parentResourceId": null,
    "hasChildren": true,
    "data": {}
}
],
1,

```

```

    "metadata":{
      "size": 20,
      "totalElements": 1,
      "totalPages": 1,
      "number": 1,
      "offset": 0
    }
  }
}

```

9.4 Syntax for Navigating to the Children of a Deployed Resource

GET /api/consumer/resourceViews retrieves a list of the child nodes of a deployment, including virtual machines, networks, and other objects you may have configured on the design canvas.

Using the REST API to Get Additional Deployment Information

In addition to general information about the provisioned resource, the response contains additional HATEOAS links that enable you to obtain additional details and information about each returned child resource.

Table 7-3. HATEOAS Link Functions as Defined by rel Field

| Link | Description |
|--|--|
| GET: Parent Resource | URI to get the resourceView for the parent item. See Syntax for Getting Deployment Details . |
| GET:Template {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to get a template request for a specific action that you can perform on this resource. |
| POST: {com.NRB.csp.component.cafe.composition@resource.action.deployment.\$actionName} | URI to which to post the request to perform an action, based on the corresponding template. |

Input

Use the supported input parameters to control the command output.

| Parameter | Description |
|--------------|--|
| URL | https://\$NECS/catalog-service/api/consumer/resources/\$resourceId |
| \$NECS | Specifies the appliance name and fully qualified domain name, or IP address of the NECS Automation server. |
| \$token | Specifies a valid HTTP bearer token with necessary credentials. |
| \$resourceId | Specifies a resource ID. See Syntax for Getting Deployment Details to view all of your requests and search for a request ID. |
| managedOnly | If true, the returned requests are from the user's managed subtenants. |
| page | Specifies a page number. |
| limit | Specifies the number of entries to display on a page. |

| Parameter | Description |
|------------------|--|
| <i>\$orderby</i> | Specifies how to order multiple comma-separated properties sorted in ascending or descending order. |
| <i>\$top</i> | Specifies the number of returned entries from the top of the response (total number per page in relation to skip). |
| <i>\$skip</i> | Specifies the number of entries to skip. |
| <i>filter</i> | Contains a Boolean expression to determine if a particular entry is included in the response. |

Output

The command output contains property names and values based on the command input parameters.

Note Price is referred to as cost in API commands and output.

Table 7-4. Output Parameters

| Property | Description |
|------------------|---|
| resourceId | The unique identifier of the resource. |
| iconId | Specifies an icon for this request based on the requested object type. |
| name | The user friendly name of the resource. |
| description | An extended user friendly description of the resource. |
| status | The status of the resource. For example, On, Off, etc. |
| catalogItemId | The identifier of the catalog item associated with this provisioned resource. |
| catalogItemLabel | The label of the catalog item associated with this provisioned resource. |
| requestId | The unique identifier of the request that created this provisioned resource. |
| businessGroupId | The unique identifier of the business group that owns this resource. |
| tenantId | The unique identifier of the tenant that owns this resource. |
| owners | The owner of this resource. |
| resourceType | The type identifier of this resource. For example, Virtual Machine. |
| parentResourceId | The unique identifier of the parent resource. Used for child resources of a multi-machine resource. |
| hasChildren | Returns true if this resource has child resources. Used if this is a multi-machine resource. |
| dateCreated | The date and time at which the resource was created. |
| lastUpdated | The date and time at which the resource was most recently modified. |
| lease | The current lease of the resource. |
| costs | An optional rate card of the prices and charges levied against the resource. This parameter is deprecated. |
| costToDate | An optional rate card of the existing prices and charges levied against the resource. This parameter is deprecated. |
| totalCost | An optional rate card of the prices and charges levied for the entire lease period. This parameter is deprecated. |

Table 7-4. Output Parameters (Continued)

| Property | Description |
|--------------------|---|
| expenseMonthToDate | The expense of the resource from the beginning of the month until the current date. This value is updated daily by NECS Business for Cloud. |
| data | The extended, provider defined properties of the resource. |

Example Curl Command

This example retrieves all children of the resource with an ID of c4d3db3e-e397-44ff-a1c9-0ecebdba12f4%27.

```
$curl --insecure -s -H "Content-Type: multipart/form-data" -H "Authorization: Bearer $token" https://$NECS/catalog-service/api/consumer/resourceViews?managedOnly=false&withExtendedData=true&withOperations=true&%24filter=parentResource%20eq%20%27c4d3db3e-e397-44ff-a1c9-0ecebdba12f4%27
```

Example: JSON Output

The validation output displays the validation status of each content item within the package.

```
{
  "links": [],
  "content": [
    {
      "@type": "CatalogResourceView",
      "links": [
        {
          "@type": "link",
          "rel": "GET: Parent Resource",
          "href": "https://$NECS/catalog-service/api/consumer/resourceViews/c4d3db3e-e397-44ff-a1c9-0ecebdba12f4"
        },
        {
          "@type": "link",
          "rel": "GET Template:
{com.nrb.csp.component.iaas.proxy.provider@resource.action.name.machine.ChangeLease}",
          "href": "https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-b5be-b229453eca4a/actions/b5739e30-871d-48c7-9012-f2a7cf431dc1/requests/template"
        },
        {
          "@type": "link",
          "rel": "POST:
{com.nrb.csp.component.iaas.proxy.provider@resource.action.name.machine.ChangeLease}",
          "href": "https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-b5be-b229453eca4a/actions/b5739e30-871d-48c7-9012-f2a7cf431dc1/requests"
        },
        {
          "@type": "link",
          "rel": "GET Template:
{com.nrb.csp.component.iaas.proxy.provider@resource.action.name.machine.PowerOff}",
          "href": "https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-
```

```

b5be-b229453eca4a/actions/02bad06d-f92b-4cf8-b964-37bb5d57be38/requests/template"
    },
    {
        "@type": "link",
        "rel": "POST":
{com.NRB.csp.component.iaas.proxy.provider@resource.action.name.machine.PowerOff}},
        "href": "https://$NECS/catalog-service/api/consumer/resources/dd37b7a1-829c-4773-
b5be-b229453eca4a/actions/02bad06d-f92b-4cf8-b964-37bb5d57be38/requests"
    }
],
"resourceId": "dd37b7a1-829c-4773-b5be-b229453eca4a",
"iconId": "cafe_default_icon_genericCatalogItem",
"name": "DEMO-002",
"description": null,
"status": "On",
"catalogItemId": null,
"catalogItemLabel": null,
"requestId": null,
"resourceType":
"{com.NRB.csp.component.iaas.proxy.provider@resource.type.registration.name.Infrastructure.Virtual}"
,
    "owners": [
        "Connie Summers"
    ],
    "businessGroupId": "c0683388-6db2-4cb5-9033-b24d15ad3766",
    "tenantId": "mycompany",
    "dateCreated": "2015-07-29T13:54:58.804Z",
    "lastUpdated": "2015-07-29T13:55:01.371Z",
    "lease": {
        "start": "2015-07-29T13:51:33.000Z"
    },
    "costs": {
        "leaseRate": {
            "type": "moneyTimeRate",
            "cost": {
                "type": "money",
                "currencyCode": "USD",
                "amount": 0
            },
        },
        "basis": {
            "type": "timeSpan",
            "unit": "DAYS",
            "amount": 1
        }
    },
    "costToDate": {
        "type": "money",
        "currencyCode": "USD",
        "amount": 0
    },
    "totalCost": null,
    "parentResourceId": "c4d3db3e-e397-44ff-a1c9-0ecebdba12f4",
    "hasChildren": false,
    "data": {

```

```

"ChangeLease": true,
"ConnectViaRdp": true,
"ConnectViaVmrc": true,
"DISK_VOLUMES": [
  {
    "componentTypeId": "com.NRB.csp.component.iaas.proxy.provider", "componentId":
    null,
    "classId": "dynamicops.api.model.DiskInputModel",
    "typeFilter": null,
    "data": {
      "DISK_CAPACITY": 6,
      "DISK_INPUT_ID": "DISK_INPUT_ID1"
    }
  },
  {
    "componentTypeId": "com.NRB.csp.component.iaas.proxy.provider", "componentId":
    null,
    "classId": "dynamicops.api.model.DiskInputModel",
    "typeFilter": null,
    "data": {
      "DISK_CAPACITY": 6,
      "DISK_INPUT_ID": "DISK_INPUT_ID2"
    }
  }
],
"Destroy": true,
"EXTERNAL_REFERENCE_ID": "vm-38153",
"Expire": true,
"IS_COMPONENT_MACHINE": false,
"MachineBlueprintName": "system_blueprint_vsphere",
"MachineCPU": 1,
"MachineDailyCost": 0,
"MachineDestructionDate": null,
"MachineExpirationDate": null,
"MachineGroupName": "Demo Group",
"MachineGuestOperatingSystem": null,
"MachineInterfaceDisplayName": "vSphere (vCenter)",
"MachineInterfaceType": "vSphere",
"MachineMemory": 4096,
"MachineName": "DEMO-002",
"MachineReservationName": "vCenter55",
"MachineStorage": 12,
"MachineType": "Virtual",
"NETWORK_LIST": [
  {
    "componentTypeId": "com.NRB.csp.component.iaas.proxy.provider", "componentId":
    null,
    "classId": "dynamicops.api.model.NetworkViewModel",
    "typeFilter": null,
    "data": {
      "NETWORK_MAC_ADDRESS": "00:50:56:ba:6b:85",
      "NETWORK_NAME": "VM Network SQA"
    }
  }
],

```

```

        "PowerOff": true,
        "Reboot": true,
        "Reconfigure": true,
        "Reprovision": true,
        "Reset": true,
        "SNAPSHOT_LIST": [],
        "Shutdown": true,
        "Suspend": true,
        "ip_address": "10.118.194.213",
        "machineId": "f3579990-a3c4-4e17-9593-1f8893636876"
    }
},
{
    "@type": "CatalogResourceView",
    "links": [
        {
            "@type": "link",
            "rel": "GET: Parent Resource",
            "href": "https://$NECS/catalog-service/api/consumer/resourceViews/c4d3db3e-
e397-44ff-a1c9-0ecebdba12f4"
        },
        {
            "@type": "link",
            "rel": "GET Template:
{com.NRB.csp.component.network.service@resource.action.destroy.name,
{{{com.NRB.csp.component.iaas.proxy.provider@network.network.type.registration.name.Infrastructure.N
etwork.Network.Existing}}}}",
            "href": "https://$NECS/catalog-
service/api/consumer/resources/f735b57a-fe6f-4108-876f-1c1055ca2cec/actions/ec5c522d-
7b5b-4d0b-b9f2-1aedf01a2f0c/requests/template"
        },
        {
            "@type": "link",
            "rel": "POST:
{com.NRB.csp.component.network.service@resource.action.destroy.name,
{{{com.NRB.csp.component.iaas.proxy.provider@network.network.type.registration.name.Infrastructure.N
etwork.Network.Existing}}}}",
            "href": "https://$NECS/catalog-service/api/consumer/resources/f735b57a-
fe6f-4108-876f-1c1055ca2cec/actions/ec5c522d-7b5b-4d0b-b9f2-1aedf01a2f0c/requests"
        }
    ],
    "resourceId": "f735b57a-fe6f-4108-876f-1c1055ca2cec",
    "iconId": "cafe_default_icon_genericCatalogItem",
    "name": "Existing Network",
    "description": null,
    "status": null,
    "catalogItemId": null,
    "catalogItemLabel": null,
    "requestId": null,
    "resourceType":
"{com.NRB.csp.component.iaas.proxy.provider@network.network.type.registration.name.Infrastructure.Ne
twork.Network.Existing}",
    "owners": [
        "Connie Summers"
    ],
    "businessGroupId": "c0683388-6db2-4cb5-9033-b24d15ad3766",

```



www.nrb.be



www.linkedin.com/company/nrb



[@daringtocommIT](https://twitter.com/daringtocommIT)

info@nrb.be | +32 (0)4 249 72 11

NRB S.A. / nv Parc Industriel des Hauts-Sarts - 2^e Avenue 65 - 4040 Herstal | Rue d'Arlon / Aarlenstraat 53 - 1040 Bruxelles / Brussel

