Cisco Process Orchestrator REST Web Services Guide

Release 3.2.2
May 2016

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Contents

Preface

The REST Web services documentation describes the REST web services API used with Cisco Process Orchestrator. This documentation describes the JSON and XML formatting used to present the input and output of jobs processed via Web services as well as configuration of the http ports used to access the services.

Audience

The information in this guide is intended for experienced users; typically, your IT organization. With Cisco Process Orchestrator REST Web services, your IT developers can, for example:

- Start Cisco Process Orchestrator processes and monitor the started process until its completion.
- View the process instance information of a started process.
- Programmatically automate the process of creating targets, runtime user accounts, target properties, global variables and tasks using the Web service.

Related Documentation

For detailed REST API paths, and required inputs and outputs to expect for each path, see the Cisco Process Orchestrator REST API Service

For more information about the Cisco Process Orchestrator and related products, see the Cisco Process Orchestrator Documentation Overview.

Configuring Cisco Process Orchestrator REST Web Services

In Cisco Process Orchestrator, the end user can expose a Northbound REST Web service into the Cisco Process Orchestrator server. This REST Web service is disabled by default, but the end user can expose it either via HTTP or HTTPS end points, on the port of their choosing.

After the REST Web service is exposed, it can be used by other tools as an integration point to start processes, disable/enable targets and perform other actions.

Configuring HTTP Settings in the Console

Use the Web service to modify Cisco Process Orchestrator REST Web service configuration settings. The fields displayed on this property page allow users to modify the HTTP and HTTPS ports as well as the authentication of the HTTP endpoints.

Securing the Cisco Process Orchestrator REST Web service

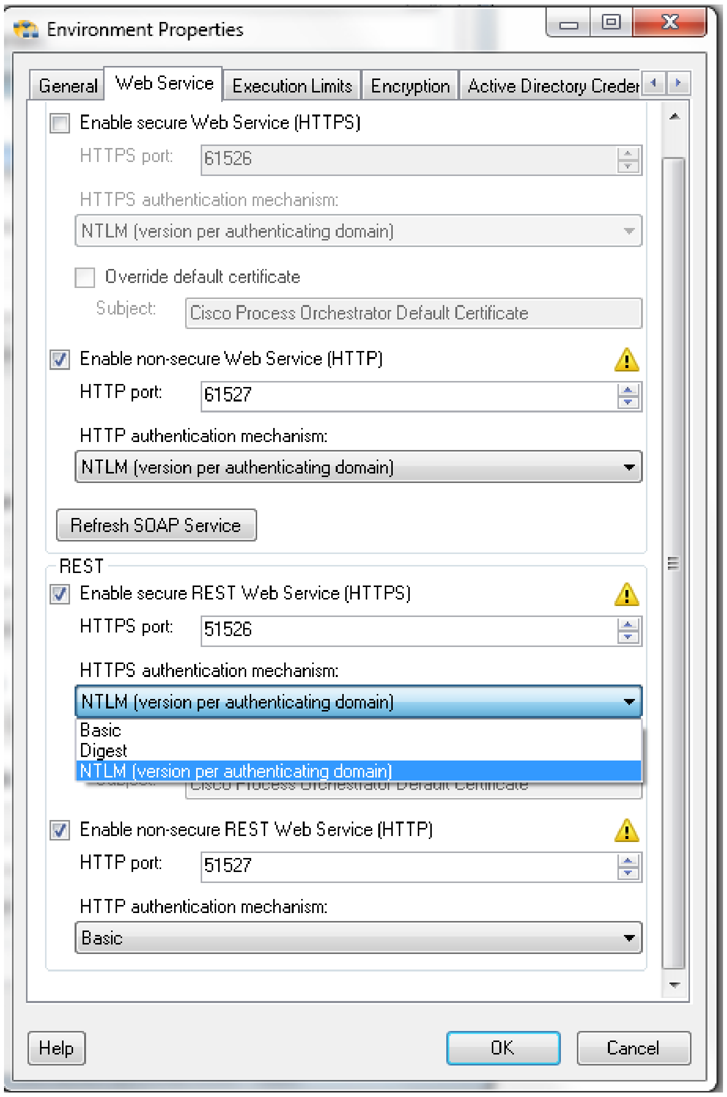
Cisco Process Orchestrator allows users the ability to modify the authentication for the HTTP endpoints. Use the following steps to secure the Cisco Process Orchestrator REST Web service.

To enable the HTTPs REST Web service:

Step 1 On the Cisco Process Orchestrator Console, choose **File** > **Environment Properties**. The Environment Properties dialog box displays.

Step 2 Click the **Web service** tab to continue.

Figure 1: Server Properties Dialog Box, Web service Tab



Environment Properties

General | Web Service | Execution Limits | Encryption | Active Directory Creder

☐ Enable secure Web Service (HTTPS)
HTTPS port: 61526
HTTPS authentication mechanism: NTLM (version per authenticating domain)
☐ Override default certificate
Subject: Cisco Process Orchestrator Default Certificate

☑ Enable non-secure Web Service (HTTP)
HTTP port: 61527
HTTP authentication mechanism: NTLM (version per authenticating domain)

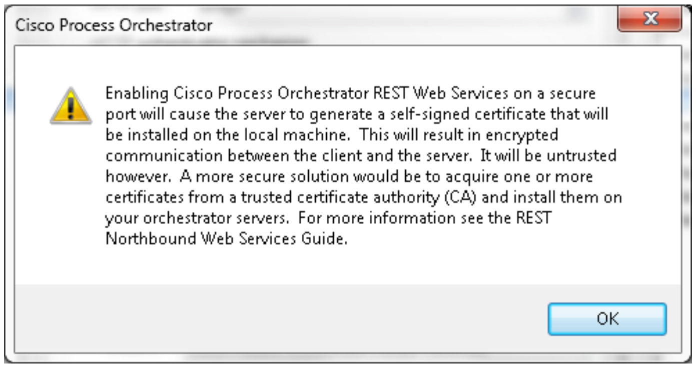
Refresh SOAP Service

REST
☑ Enable secure REST Web Service (HTTPS)
HTTPS port: 51526
HTTPS authentication mechanism: NTLM (version per authenticating domain)
Basic
Digest
NTLM (version per authenticating domain)

☑ Enable non-secure REST Web Service (HTTP)
HTTP port: 51527
HTTP authentication mechanism: Basic

Help | OK | Cancel

Step 3 Check the **Enable secure Web service (HTTPS)** check box to configure the authentication for the HTTPS endpoint.

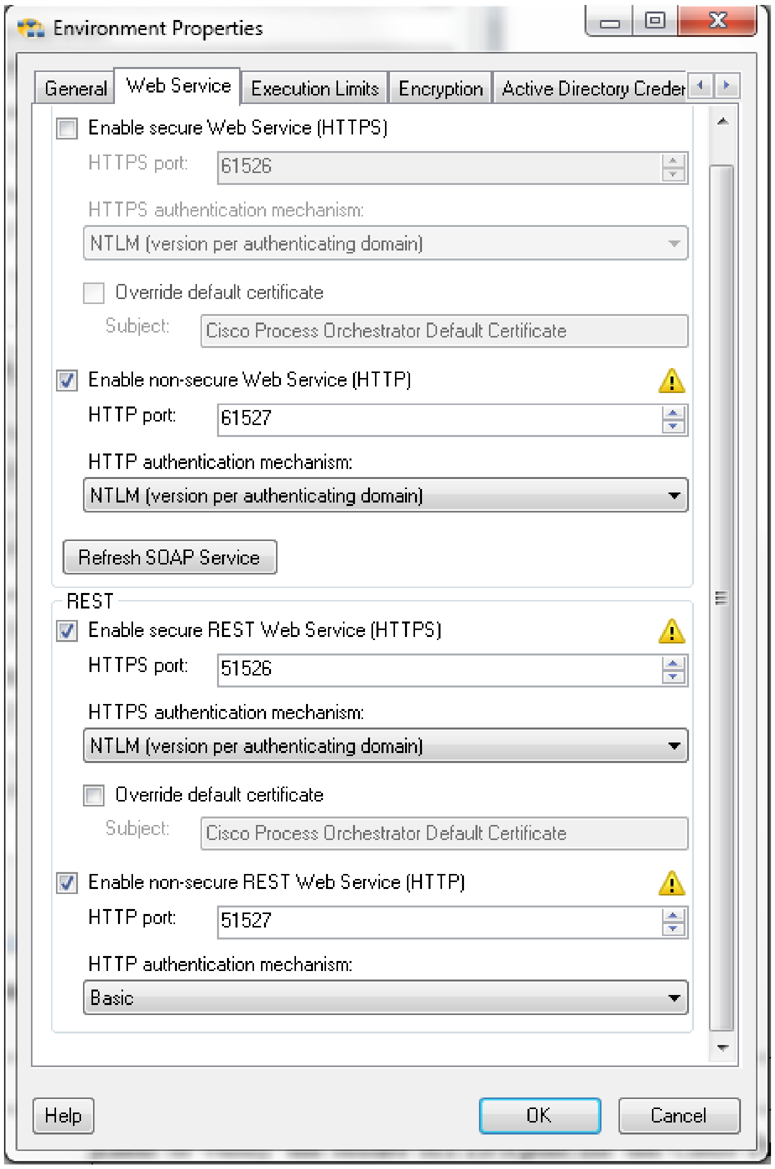


Cisco Process Orchestrator

Enabling Cisco Process Orchestrator REST Web Services on a secure port will cause the server to generate a self-signed certificate that will be installed on the local machine. This will result in encrypted communication between the client and the server. It will be untrusted however. A more secure solution would be to acquire one or more certificates from a trusted certificate authority (CA) and install them on your orchestrator servers. For more information see the REST Northbound Web Services Guide.

OK

Step 4 Click **OK** to continue.


Environment Properties

Enable secure Web Service (HTTPS)

HTTPS port: 61526

HTTPS authentication mechanism: NTLM (version per authenticating domain)

Override default certificate

Subject: Cisco Process Orchestrator Default Certificate

☑ Enable non-secure Web Service (HTTP)

HTTP port: 61527

HTTP authentication mechanism: NTLM (version per authenticating domain)

REST

☑ Enable secure REST Web Service (HTTPS)

HTTPS port: 51526

HTTPS authentication mechanism: NTLM (version per authenticating domain)

Override default certificate

Subject: Cisco Process Orchestrator Default Certificate

☑ Enable non-secure REST Web Service (HTTP)

HTTP port: 51527

HTTP authentication mechanism: Basic

Step 5 Complete the following fields, as necessary.

Field	**Description**
HTTPS port	Enter or verify the secure HTTPS port for the Cisco Process Orchestrator REST Web service. (Default: 51526)

HTTPS authentication mechanism	Choose the appropriate authentication for the Web service. ▪ Basic—sends a username and password as the method of authentication. It's the simplest method of authentication, but the least secure. ▪ Digest—sends cryptographic representation of the password rather than the password itself. This authentication method is more secure than basic authentication. ▪ Ntlm—authentication protocol used on networks that include systems running on the Windows operating system. This option can be used to return to the normal mode of operation.
Override default certificate	If left unchecked the default certificate will be created, installed and associated with the HTTPs port. If checked, the certificate whose subject is listed in the Subject field will be associated with the HTTPs port. If using a certificate from a certificate authority (CA) the certificate must be installed in the local computer's certificate store.

Step 6 Click **OK** to save the settings.

Enabling a Non-Encrypted Endpoint of the REST Web service

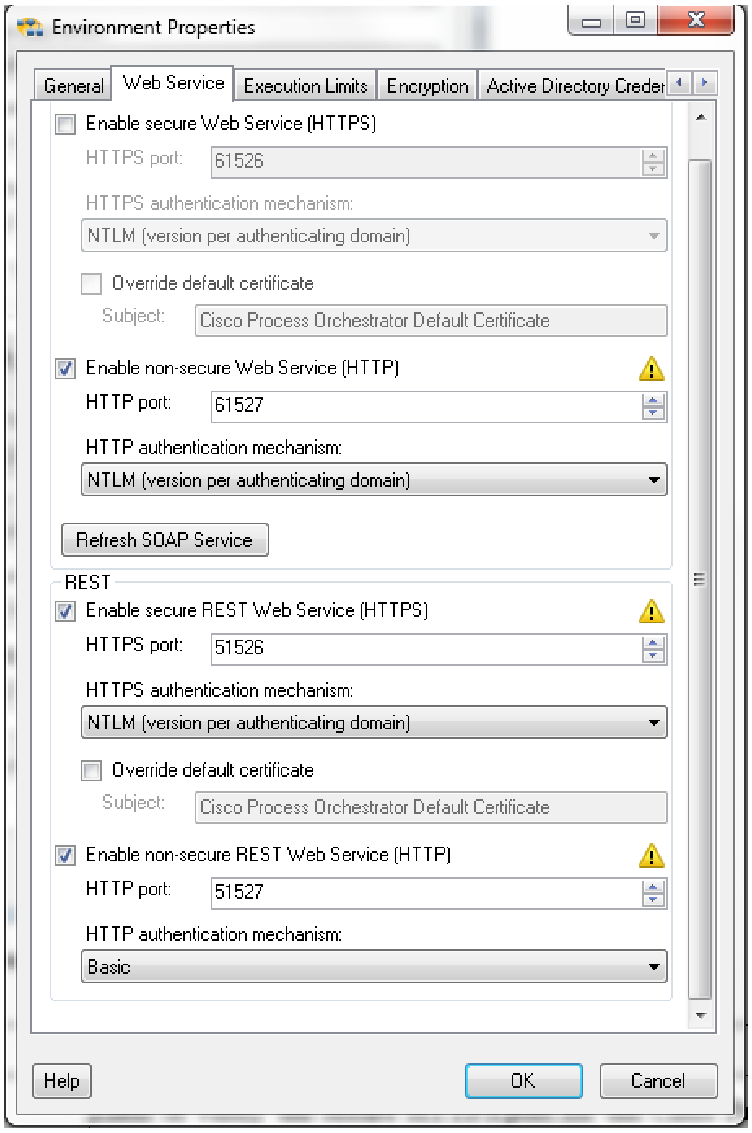
Use the following steps to open a non-encrypted endpoint of the Cisco Process Orchestrator REST Web service.

To open a non-encrypted endpoint:

Step 1 On the Cisco Process Orchestrator Console, choose **File** > **Environment Properties**. The Environment Properties dialog box displays.

Step 2 Click the **Web Service** tab to continue.

Figure 2: Environment Properties Dialog Box—Web Service Tab



Environment Properties

General | Web Service | Execution Limits | Encryption | Active Directory Creden

- [ ] Enable secure Web Service (HTTPS)
  - HTTPS port: 61526
  - HTTPS authentication mechanism: NTLM (version per authenticating domain)
  - [ ] Override default certificate
  - Subject: Cisco Process Orchestrator Default Certificate
- [✓] Enable non-secure Web Service (HTTP)
  - HTTP port: 61527
  - HTTP authentication mechanism: NTLM (version per authenticating domain)

Refresh SOAP Service

REST

- [✓] Enable secure REST Web Service (HTTPS)
  - HTTPS port: 51526
  - HTTPS authentication mechanism: NTLM (version per authenticating domain)
  - [ ] Override default certificate
  - Subject: Cisco Process Orchestrator Default Certificate
- [✓] Enable non-secure REST Web Service (HTTP)
  - HTTP port: 51527
  - HTTP authentication mechanism: Basic

Help | OK | Cancel

Step 3 Check the **Enable non-secure Web service (HTTP)** check box to unencrypt the HTTP endpoints.

Step 4 Click **OK** to continue.

Step 5 Complete the following fields, as necessary.

Field	Description
HTTP port	Enter or verify the secure HTTP port for the REST Web service. (Default: 51527)

HTTP authentication mechanism	Choose the appropriate authentication for the REST Web service. ▪ Basic—sends a username and password as the method of authentication. It's the simplest method of authentication, but the least secure. ▪ Digest—sends cryptographic representation of the password rather than the password itself. This authentication method is more secure than basic authentication. ▪ Ntlm—authentication protocol used on networks that include systems running on the Windows operating system. This option can be used to return to the normal mode of operation.

Step 6 Click **OK** to save the settings.

All transmissions through the chosen *NonsecuredHttpPort* are unencrypted. Communications over the SSL-enabled ports (and between the server and Console) will all be unaffected by this setting.

Cisco Process Orchestrator REST API Sample Requests

Cisco Process Orchestrator REST API supports inputs in two different formats – XML and JSON. Some sample requests are listed below to get familiar with the syntax. For XML format, there are some conventions that Cisco Process Orchestrator REST API understands which will be listed below. For more information about a specific REST API, and what are the required input parameters, please refer to *Cisco Process Orchestrator REST API Service.*

JSON

If to create a web target, the request will look like following,

```
{
  "Type" : "WebTarget",
  "BaseUrl": "http://localhost:51527/api/v1/",
  "IgnoreCertificateErrors": false,
  "ProxyServerAddress": "10.201.11.121",
  "ProxyPortNumber": "51527",
  "ProxyAuthentication": "None",
  "Enabled": true,
  "Name": "LocalTarget",
  "Description": "",
  "Organization": ""
}
```

To create a Unix/Linux target:

```
{
  "OSName": "Linux",
  "Type": "UnixLinuxSystem",
  "OSVersion": "GNU/Linux",
  "NodeName": "sjc-cent59-rac3.tidalsoft.local",
  "Host": "sjc-cent59-rac3",
  "Port": 22,
  "Protocol": "SSH",
  "DefaultRuntimeUserNameorID": "7ad4bda0-cba5-4ad1-9bde-30a3a7082acf",
  "KshPath": "/usr/bin/ksh",
  "PromptPrefix": "",
  "MaxConcurrentSessions": 3,
  "ExpectTemplateNameOrId": "a93fe037-a0d3-4470-a68c-34bf372159c8",
  "Enabled": true,
  "Description": "",
  "Organization": ""
}
```

To create a SMTP server target:

```
{
  "SMTPServer": "mail.cisco.com",
  "Type": "EmailSMTPServer",
  "SMTPPort": 25,
  "Sender": "ramygane@cisco.com",
  "CredentialRequired": true,
  "DefaultRuntimeUserNameorID": "ramygane@cisco.com",
  "EnableDigitalSignature": false,
  "Enabled": false,
  "Description": "",
  "Organization": ""
}
```

To create a string target property extension:

```
{
  "ValidReferenceTypes": ["2097ba7b-3e94-0c5b-8243-90df2cca8626"],
  "ValidTargetTypes":    [
     "799d63c7-a140-4ab8-9fca-aac2d0456696",
     "894a628d-df98-a8bb-c9e3-4318f10b3835",
     "86e5a024-9ad5-462c-819b-c0e479e34d17"],
  "Type": "String",
  "GroupNames":    ["Custom", "123#"],
  "GroupIndex": 1,
  "Name": "TRP1234",
  "Value": "",
  "Description": ""}
```

If to create a new alert task, the request will look like following,

```
 {
  "AlertClass": 23,
  "Type": "Alerttask",
  "WebFormXSLFileName": "DefaultAlertTaskTransform.xslt",
  "ItilStatus": "New",
  "AffectedTargetConfigurationItemId": "fe16641c-0924-41ea-8730-ecc5da6ae036",
  "ConfigurationItemId": "00000000-0000-0000-0000-000000000000",
  "AffectedServices": "",
  "AffectedOrganizations": "",
  "Severity": "Normal",
  "AutomationSummary": "",
  "Name": "Alert123",
  "Description": "Hello there",
  "DueDate": "9999-12-31T23:59:59.9999999Z",
  "ExpirationDate": "2016-02-07T19:38:58.0750893Z",
  "CompletedTime": "2015-12-09T19:44:40.1383351Z",
  "Priority": "Medium",
  "NotificationRecipients": [],
  "ExternalSystem": "",
  "ExternalId": "",
  "RelatedTaskIds": [],
  "CategoryIds": [“230b73e8-a781-42dc-894f-339971db75bb”],
  "Parameters": []
}
```

To create a windows runtime user:

```
{
  "type": "WindowsUser",
  "UserName": "ramygane",
  "Name": "ramygane",
  "Password": "cisco,1212",
  "Id": "5c9a2c31-3e37-44b4-b1a8-09452db6369a",
  "Domain": "tidalsoft.local",
  "Description": ""
}
```

XML

To create a web target in XML format, the request body will look like following. Input parameters need to be wrapped by a <value> tag

```
<value>
<type>WebTarget</type>
<baseUrl>http://localhost:51527/api/v1/</baseUrl>
<IgnoreCertificateErrors>false</IgnoreCertificateErrors>
<ProxyServerAddress>10.201.11.121</ProxyServerAddress>
<ProxyPortNumber>51527</ProxyPortNumber>
<ProxyAuthentication>None</ProxyAuthentication>
<Enabled>true</Enabled>
<Description></Description>
<organization></organization>
</value>
```

In some cases where a list of items need to be provided, add “ArrayOf” in front of the input parameter name, and wrap each item with <item> tag. For example, the API to create a string target property will look like following

```
<value>
    <type>string</type>
    <value>1234</value>
    <ArrayOfValidTargetTypes>
        <item>WebTarget</item>
   </ArrayOfValidTargetTypes>
   <ArrayOfGroupNames>
       <item>API</item>
       <item>Rest</item>
   </ArrayOfGroupNames>
</value>
```

In order to get process instance statuses, you need to provide a list of process instance Ids.

```
<ArrayOfValue>
    <item>fcd66a6d-d316-f488-67cd-021e64ff3da4</item>
    <item>1184b101-320d-e8ef-9dd9-fcc01340090e</item>
</ArrayOfValue>
```

Online REST API Document

Swagger

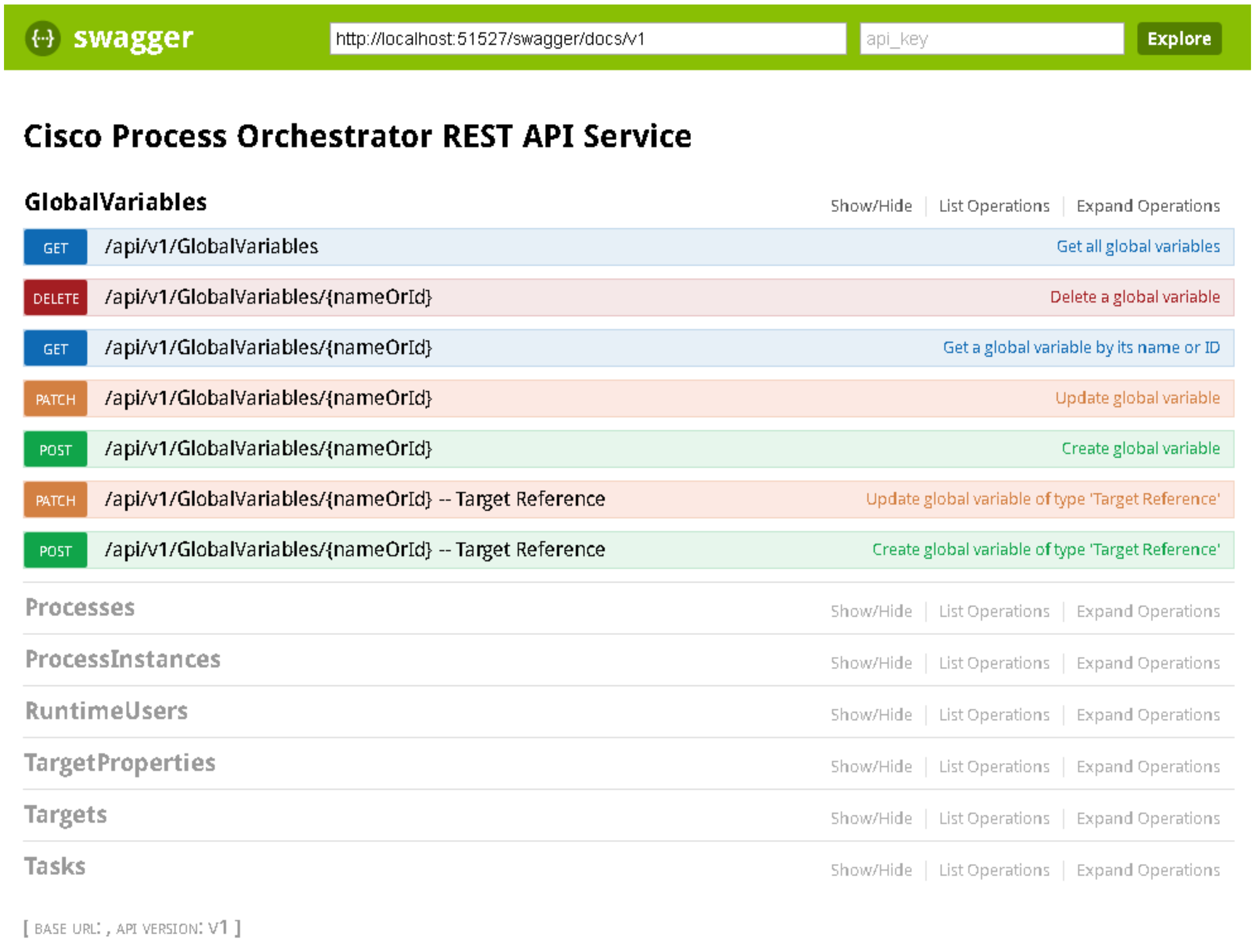
Cisco Process Orchestrator REST API is Swagger-enabled API. If either one of secured REST web service or non-secured REST web service is enabled, an online Swagger platform is enabled by default. It provides interactive documentation, and parameters each API expects. If secured Cisco Process Orchestrator REST API is enabled, please access swagger at:

```
https://{Host}:{SecuredPort}/swagger
```

If non-secured Cisco Process Orchestrator REST API is enabled, please access swagger at:

```
http://{Host}:{NonSecuredPort}/swagger
```

The platform will look like following:



swagger
http://localhost:51527/swagger/docs/v1
api_key
Explore
Show/Hide | List Operations | Expand Operations

Cisco Process Orchestrator REST API Service

GlobalVariables

GET	/api/v1/GlobalVariables	Get all global variables
DELETE	/api/v1/GlobalVariables/{nameOrId}	Delete a global variable
GET	/api/v1/GlobalVariables/{nameOrId}	Get a global variable by its name or ID
PATCH	/api/v1/GlobalVariables/{nameOrId}	Update global variable
POST	/api/v1/GlobalVariables/{nameOrId}	Create global variable
PATCH	/api/v1/GlobalVariables/{nameOrId} -- Target Reference	Update global variable of type 'Target Reference'
POST	/api/v1/GlobalVariables/{nameOrId} -- Target Reference	Create global variable of type 'Target Reference'

Processes

ProcessInstances

RuntimeUsers

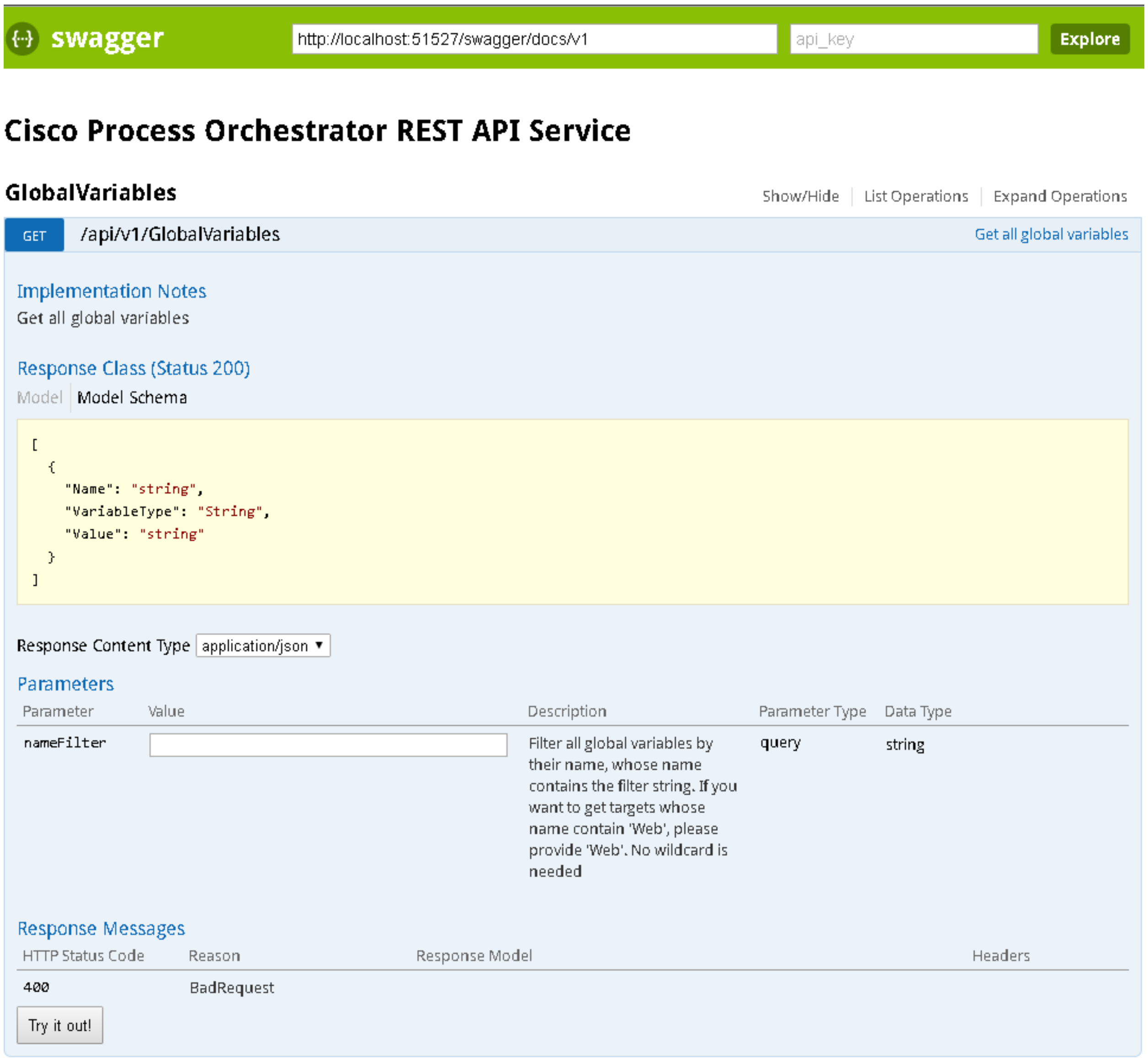
TargetProperties

Targets

Tasks

[BASE URL: , API VERSION: V1]

If all required parameters are provided, each API can be called directly from this platform. It will have details including input parameters users need to provide, what response data will look like, and if the API is called from this platform, what is the response received.



Cisco Process Orchestrator REST API Service

GlobalVariables

GET /api/v1/GlobalVariables — Get all global variables

Implementation Notes

Get all global variables

Response Class (Status 200)

```
[
  {
    "Name": "string",
    "VariableType": "String",
    "Value": "string"
  }
]
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
nameFilter		Filter all global variables by their name, whose name contains the filter string. If you want to get targets whose name contain 'Web', please provide 'Web'. No wildcard is needed	query	string

Response Messages

HTTP Status Code	Reason	Response Model	Headers
400	BadRequest		