Cisco Process Orchestrator REST Web Services Guide

Release 3.5
August 2017

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

Cisco Process Orchestrator REST Northbound Web Services Guide

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 services into the Cisco Process Orchestrator server. The REST Web services are disabled by default. Users can enabled either secure web service (HTTPS) or non-secure web service (HTTPS), on the port of their choosing.

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

Configuring Web Services Global Settings

The ***Web Services*** setting can be configured via the **File > Environment Properties > Web Services**

Users can configure the global settings for both secure (HTTPS) & non-secure (HTTP) REST web services. The settings will be applied to all Process Orchestrator servers in the same Process Orchestrator environment.

Securing the Cisco Process Orchestrator REST Web service

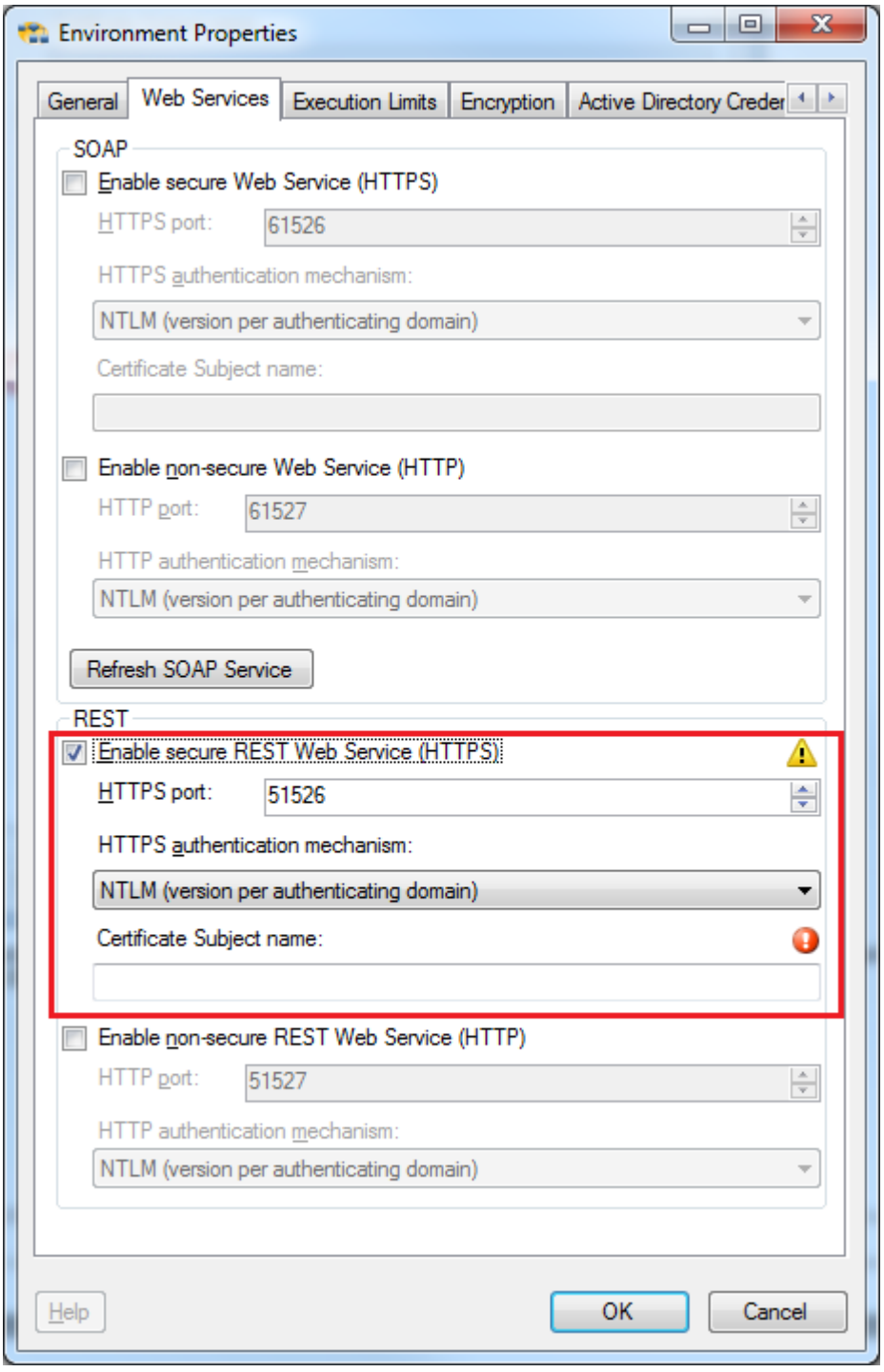
Cisco Process Orchestrator allows users the ability to modify the authentication for the HTTPs endpoints. Use the following steps to secure the Cisco Process Orchestrator REST Web service.

To enable the HTTPs REST Web service:

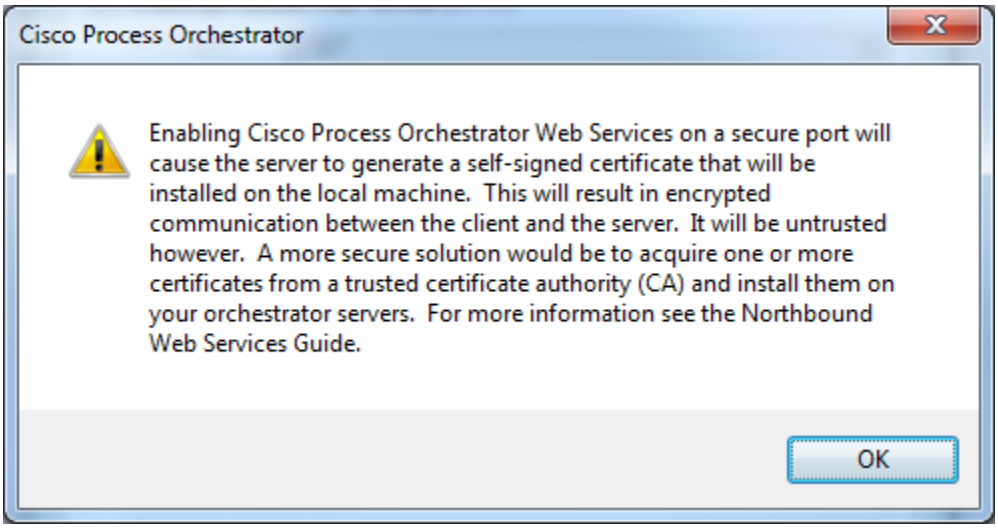
Step 1 On the Cisco Process Orchestrator Console, choose **File > Environment Properties**. The Environment Properties dialog box displays.

Step 2 Click the **Web services** tab

Figure 1: Server Properties Dialog Box, Web service Tab



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



Step 4 Click **OK** to continue.

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.
Certificate Subject name	Specify the subject name of a valid certificate to be used by all Cisco Process Orchestrator servers in this environment. The certificate must be installed in the certificate store of the computers where Cisco Process Orchestrator server is installed.

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

Enabling a Non-Encrypted Endpoint of the REST Web service

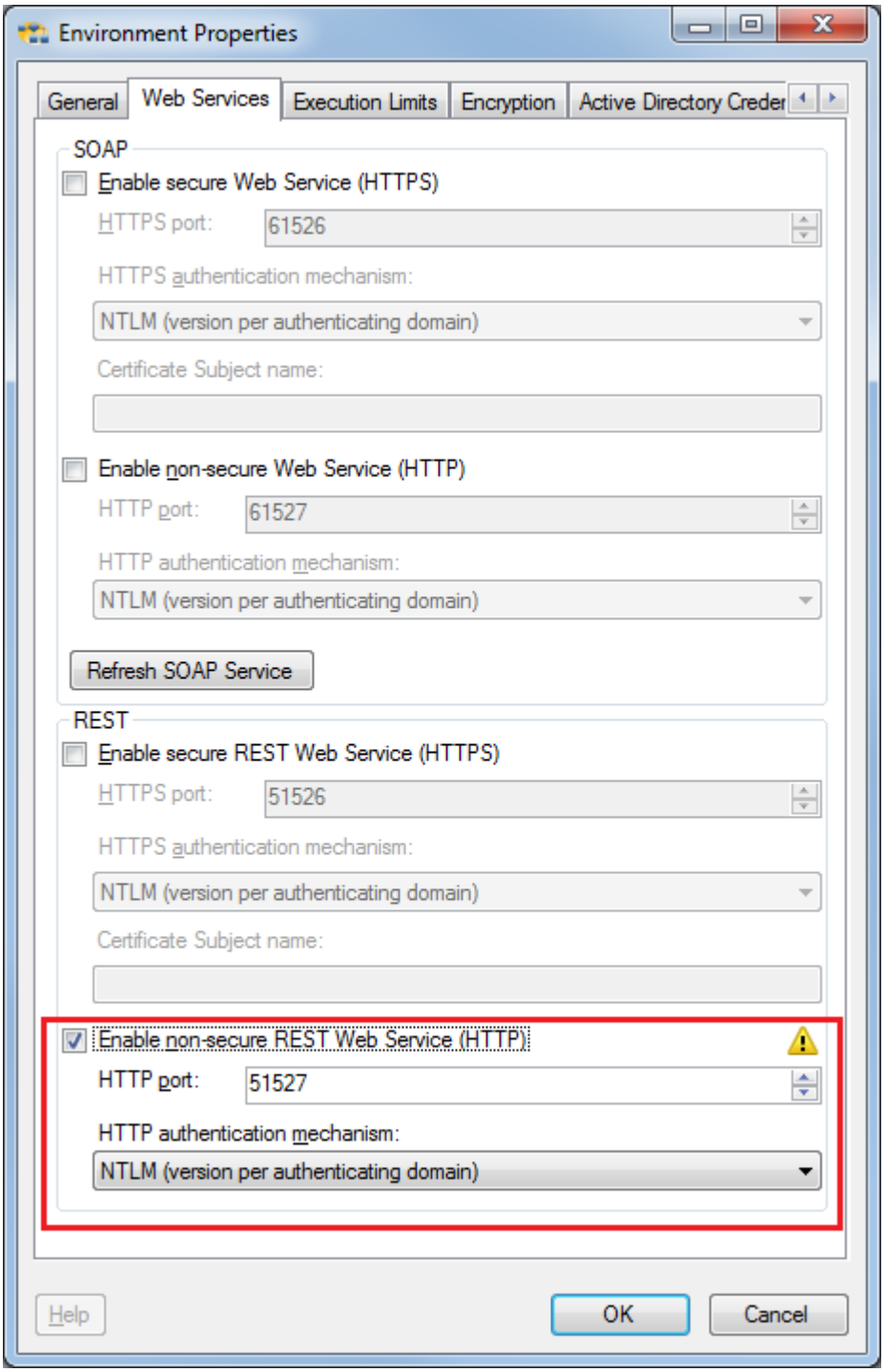
Use the following steps to open a non-encrypted endpoint of the Cisco Process Orchestrator REST Web service.

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 Services** tab

Figure 2: Environment Properties Dialog Box—Web Service Tab



Step 3 Check the **Enable non-secure REST Web service (HTTP)** check box to enable the non-secure REST Web Service.

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.

Override Web Services Settings for Individual Process Orchestrator Server

Users can overwrite the web services global settings for individual Process Orchestrator Server via the server config file (Tidal.Automation.Server.exe.config). This file can be found in the Cisco Process Orchestrator install directory.

Note that the ***Cisco Process Orchestrator Server*** service (*Service Name: Orchestrator Server*) needs to be restarted for the server to pick up the changes done to the config file.

The following properties can be overwritten.

Non-secure REST Web Service Port

- Locate the following block of XML in the config file. Specify a new port number in the value tag

```
<setting
   name="NonsecuredRESTPort"
   serializeAs="String">
   <value>-1</value>
</setting>
```

Secure REST Web Service Port

- Locate the following block of XML in the config file. Specify a new port number in the value tag

```
<setting
   name="SecuredRESTPort"
   serializeAs="String">
   <value>-1</value>
</setting>
```

Secure REST Web Service Certificate

- Locate the following block of XML in the config file. Specify a new certificate subject in the value tag

```
<setting
   name="ServerOverrideRESTCertSubject"
   serializeAs="String">
   <value></value>
</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>
```