# Websocket Subscription



# WebSocket Subscription

Cisco NX-OS provides an interface capability to enable the switch to push notifications to interested subscribers. Through the NX-API WebSocket interface, programs and end-users can receive notifications about various state changes on the switch, eliminating the need for periodic polling.

When you perform an API query using the Cisco NX-API REST interface, you have the option to create a subscription to any future changes in the results of a given query. When any management object (MO) is created, changed, or deleted, because of a user-initiated or system-initiated action, an event is generated. If the received event changes the results of a subscribed query, the switch generates a push notification to the API client that created the subscription.

-

### Opening a WebSocket

The API subscription feature uses the WebSocket protocol (RFC 6455) to implement a two-way connection with the API client. This way, the API can send unsolicited notification messages to the client itself. To establish the notification channel, you must first open a WebSocket connection with the respective API. Only a single WebSocket connection is needed to support multiple query subscriptions within each switch. The WebSocket connection is dependent on your API session connection (via token validation), and closes when your API session ends.

There are many ways to open a WebSocket connection. You can write python client as following:

```
from websocket import create_connection

connection_string = "ws:// 10.1.2.3/socket{0}".format(token)

ws = create_connection(connection_string, sslopt={"check_hostname": False})
```

In the URI, the token is the current API session token (cookie). This example shows the URI with a token:

```
ws://10.1.2.3/socketGkZl5NLRZJl5+jqChouaZ9CYjgE58W/pMccR+LeXmdO0obG9NB
Iwo1VBo7+YC1oiJL9mS6I9qh62BkX+Xddhe0JYrTmSG4JcKZ4t3bcP2Mxy3VBmgoJjwZ76ZOuf9V9AD6X
l83lyoR4bLBzqbSSU1R2NIgUotCGWjZt5JX6CJF0=
```

### Creating a Subscription

To create a subscription to a query, perform the query with the option “?subscription=yes”. This example creates a subscription to a query of the sys/intf/phys-[eth1/1] in the JSON format:

```
GET http://10.1.1.1/api/mo/sys/intf/phys-[eth1/1].json?subscription=yes
```

The query response contains a subscription identifier, subscriptionId, that you can use to refresh the subscription and identify future notifications from the given subscription.

```
{"totalCount":"0","subscriptionId":"18374686685813276673","imdata":[]}
```

### Receiving Notifications

An event notification from the subscription delivers a data structure that contains the subscription ID and the MO description. In this JSON example, sys/intf/phys-[eth1/1] description is changed to “test”.

{"subscriptionId":["18374686685813276673"],"imdata":[{"l1PhysIf": {"attributes": {"childAction": "","descr": "test","dn": "sys/intf/phys-[eth1/1]","modTs": "2019-10-18T19:42:29.446+00:00","rn": "","status": "modified"}}}]}

As multiple active subscriptions can exist for a given query, a notification can contain multiple subscription IDs; similar as shown in the example above. Notifications are supported in either JSON or XML format.

### Refreshing the Subscription

In order to continue receiving event notifications, you must periodically refresh each subscription during your API session. To refresh a subscription, send an HTTP GET message to the API method `subscriptionRefresh` with the parameter `id` equal to the `subscriptionId` shown in the example:

```
GET http://10.1.1.1/api/subscriptonRefresh.json?id=18374686685813276673
```

The API returns an empty response to the refresh message unless the subscription has expired.

**Note** The timeout period for a WebSocket subscription is 90 seconds by default. To prevent loss of notifications, you must send a subscription refresh message at least once every 90 seconds.

In summary, WebSocket provides a powerful tool for allowing publisher-subscriber communication for event subscription within the NX-OS REST API.