Rules API

This chapter describes the Rules API.

Using the Rules API

The Rules API provides basic functionality to add, remove, activate, deactivate, and find rules. Each rule is identified by a name and a user. A user cannot have two rules with the same name, but different users can have rules with the same name.

When creating new rules, specify the rule name. To create a new rule, use the same syntax used in the Search API.

After creating a rule, the IoT FND Rules API returns a rule ID, which you can use to refer to the rule for removing, activating, and deactivating, operations. By default, IoT FND activates rules when created with actions.

In your IoT FND NB API client application, use this IoT FND server URL to access the Rules API WSDL:

http://<*server_address*> /nbapi/rules?wsdl

Example (Python-SUDS Client)

```
import sys
from suds.client import Client
import logging
logging.basicConfig(level=logging.INFO)
logging.getLogger('suds.client').setLevel(logging.DEBUG)
url = "http://localhost/nbapi/rules?wsdl"
cl = Client(url, username='root', password='Tree123!')
#print cl
#parse the properties of a rule and return a dictionary struct
def parseRule(r):
    rule = {}
    rule['id'] = r.id
    if (r.properties != None and r.properties != ""):
      #print r.properties
      for p in r.properties.entry:
        if (p.key == "literal"):
   rule['rule']=p.value
```

```
 else:
   rule[p.key]=p.value
    for a in r.actions:
      rule['action'] = a.type
      rule['label'] = a.parameter
    #print rule
    return rule
# output a CSV list of rules
users = ['root']
properties = ['username','status', 'lastUpdate', 'name', 'literal']
print "username,rulename,id,status,lastUpdate,rule,action,label"
for u in users:
  #print u
  result = cl.service.findRulesByUsername(u)
  #print result
  for r in result:
    rule = parseRule(r)
    print rule['username']+",",
    print rule['name']+",",
    print str(rule['id'])+",",
    print rule['status']+",",
    print rule['lastUpdate']+",",
    print rule['rule']+",",
    print rule['action']+",",
    print rule['label']
# create two new rules
names = ['My New Rule 1', 'My New Rule 2']
myAction1 = cl.factory.create('ruleAction')
myAction1.parameter = 'My Label'
myAction1.type = 'ADD_LABEL'
myAction2 = cl.factory.create('ruleAction')
myAction2.parameter = 'My Label'
myAction2.type = 'REMOVE_LABEL'
myActions = [myAction1,myAction2]
myRule1 = cl.factory.create("createRule")
myRule1.name = names[0]
myRule1.username = users[0]
myRule1.literal = 'deviceType:cgmesh status:down meshRssi<-90 meshHops>5'
myRule1.actions = myAction1
print myRule1
myRule2 = cl.factory.create("createRule")
myRule2.name = names[1]
myRule2.username = users[0]
myRule2.literal = 'label:"My Label" deviceType:cgmesh status:up'
myRule2.actions = myAction2
print myRule2
r = cl.service.createRule(myRule1)
myr1 = parseRule(r)
print myr1
r = cl.service.createRule(myRule2)
myr2 = parseRule(r)
print myr2
# toggle rules by name
for n in names:
  #print n
  result = cl.service.findRulesByName(n)
  if result != None:
    #print result
    for r in result:
      rule = parseRule(r)
      if (rule['status'] == "ACTIVATED"):
   r = cl.service.deactivateRule(rule['id'])
      else:
   r = cl.service.activateRule(rule['id'])
```

```
      rule = parseRule(r)
      print rule
sys.exit()
# delete the rules we created
r = dropRule(myr1["id"])
r = dropRule(myr2["id"])
```

Rules API Method Calls

activateRule

This call activates a deactivated rule. By default, when you create a rule with actions, IoT FND activates it. Only activated rules receive trigger events on rule matching.

Prototype

```
 <rul:activateRule
>
  <id
>?</id>
 </rul:activateRule>
```

Parameters

Table 1: activateRule Parameters

Parameter	Type	Description
id	long	Rule ID.

Results

This call returns the rule object if activation is successful.

Table 2: activateRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

activateRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
```

```
      <rul:activateRule>
         <!--Optional:-->
         <id>?</id>
      </rul:activateRule>
   </soapenv:Body>
</soapenv:Envelope>
```

createRule

This call creates a new rule with the specified action and name. The rule language is the same as the search language. The action can be one of the following: create label, remove label, or add event.

Prototype

```
 <rul:createRule
>
  <name
>?</name>
  <username
>?</username>
  <literal
>?</literal>
  <actions
>
  <parameter
>?</parameter>
  <type
>create_label
</type>
  </actions>
 </rul:createRule>
```

Parameters

Table 3: createRule Parameters

Parameter	Type	Description
name	string	The name of the rule.
username	string	The IoT FND username of the rule creator.
literal	string	The rule syntax.
actions	ruleAction	A list of rule actions. For every action, you must provide two strings: • parameter—Action label. • type—Action type. Valid options are: • **create_label** • **remove_label** • **add_event**

Results

Table 4: createRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

createRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
            <!--Optional:-->
            <type>?</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample createRule Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
    <soapenv:Header/>
    <soapenv:Body>
       <rul:createRule>
          <!--Optional:-->
          <name>2016-03-09-addEvent-3</name>
          <!--Optional:-->
          <username>root</username>
          <!--Optional:-->
          <literal>deviceCategory:router issue:lowMemory</literal>
          <!--Zero or more repetitions:-->
          <actions xmlns=''>

<parameter>{"eventName":"testINFO","eventSeverity":"INFO","eventMsg":"test_INFO"}</parameter>

             <type>add_event</type>
             </actions>
              <actions xmlns=''>
             <parameter>AndersonLabel</parameter>
```

```
            <type>add_label</type>
            </actions>
             <actions xmlns=''>
            <parameter>AndersonLabel</parameter>
            <type>remove_label</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

Sample createRule Response

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"/>
   <soap:Body>
      <ns2:createRuleResponse xmlns:ns2="http://rules.nbapi.cgms.cisco.com/">
         <rule>
            <actions>

<parameter>{"eventName":"testINFO","eventSeverity":"INFO","eventMsg":"test_INFO"}</parameter>

               <type>ADD_EVENT</type>
            </actions>
            <actions>
               <parameter>AndersonLabel</parameter>
               <type>ADD_LABEL</type>
            </actions>
            <actions>
               <parameter>AndersonLabel</parameter>
               <type>REMOVE_LABEL</type>
            </actions>
            <id>20001141</id>
            <properties>
               <entry>
                  <key>username</key>
                  <value>root</value>
               </entry>
               <entry>
                  <key>status</key>
                  <value>DEACTIVATED</value>
               </entry>
               <entry>
                  <key>lastUpdate</key>
                  <value>2016-03-09 17:28:25.771</value>
               </entry>
               <entry>
                  <key>name</key>
                  <value>2016-03-09-addEvent-3</value>
               </entry>
               <entry>
                  <key>literal</key>
                  <value>deviceCategory:router issue:lowMemory</value>
               </entry>
            </properties>
         </rule>
      </ns2:createRuleResponse>
   </soap:Body>
</soap:Envelope>
```

deactiveRule

This call deactivates the specified rule and returns the rule object if deactivation is successful.

Prototype

```
 <rul:deactiveRule
>
  <id
>?</id>
 </rul:deactiveRule>
```

Parameters

Table 5: deactivateRule Parameters

Parameter	Type	Description
id	long	Rule ID.

Results

This call returns the rule object if deactivation is successful.

Table 6: deactivateRule Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

deactivateRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
            <!--Optional:-->
            <type>?</type>
         </actions>
```

```
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
```

dropRule

This call removes the rule specified in the *id* parameter.

Prototype

```
 <rul:dropRule
>
  <id
>?</id>
 </rul:dropRule>
```

Parameters

Table 7: dropRule Parameters

Parameter	Type	Description
id	long	ID of rule to remove.

Results

Table 8: dropRule Results

Field	Type	Description
id	long	ID of removed rule.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

dropRule SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:createRule>
         <!--Optional:-->
         <name>?</name>
         <!--Optional:-->
         <username>?</username>
         <!--Optional:-->
         <literal>?</literal>
         <!--Zero or more repetitions:-->
         <actions>
            <!--Optional:-->
            <parameter>?</parameter>
```

```
            <!--Optional:-->
            <type>?</type>
         </actions>
      </rul:createRule>
   </soapenv:Body>
</soapenv:Envelope>
?
dropRule SOAP XML request format
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:dropRule>
         <!--Optional:-->
         <id>?</id>
      </rul:dropRule>
   </soapenv:Body>
</soapenv:Envelope>
```

findRulesByName

This call returns all rules specified in the *name* parameter.

Prototype

```
 <rul:findRulesByName
>
  <name
>?</name>
 </rul:findRulesByName>
```

Parameters

Table 9: findRulesByName Parameters

Parameter	Type	Description
name	string	Rule name.

Results

This call returns a list of rules matching the name string.

Table 10: findRulesByName Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

findRulesByName SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:findRulesByName>
         <!--Optional:-->
         <name>?</name>
      </rul:findRulesByName>
   </soapenv:Body>
</soapenv:Envelope>
```

findRulesByUsername

This call returns a list of rules created by the user defined in *username* .

Prototype

```
 <rul:findRulesByUsername
>
  <username
>?</username>
 </rul:findRulesByUsername>
```

Parameters

Table 11: findRulesByUsername Parameters

Parameter	Type	Description
username	string	The IoT FND username of the creator of the rules.

Results

Table 12: findRulesByUsername Results

Field	Type	Description
id	long	Rule ID.
properties	Map<String,String>	Rule properties such as name, username, status, and lastUpdate.
actions	List<RuleAction>	List of actions the rule performs.

findRulesByUsername SOAP XML Request Format

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rul="http://rules.nbapi.cgms.cisco.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <rul:findRulesByUsername>
```

```
         <!--Optional:-->
         <username>?</username>
      </rul:findRulesByUsername>
   </soapenv:Body>
</soapenv:Envelope>
```