その他の例

この章の内容は、次のとおりです。

API の例に関する情報

この例では、JSON および XML 構造が読みやすいように改行、スペース、インデントで展開されています。

例：JSON API を使用したトップ レベルのシステム要素の取得

次に、存在するシステム デバイスを判断するために APIC を照会する例を示します。

システム要素（APIC、スパイン、およびリーフ）に関する一般情報は、クラス top:System のオブジェクトに含まれています。

次に、API クエリー メッセージの例を示します。

```
GET http://192.0.20.123/api/class/topSystem.json
```

正常な動作時には次のような応答本文が返されます。

```
{
  "imdata" :
  [{
```

```
      "topSystem" : {
        "attributes" : {
          "instanceId" : "0:0",
          "address" : "10.0.0.32",
          "childAction" : "",
          "currentTime" : "2013-06-14T04:13:05.584",
          "currentTimeZone" : "",
          "dn" : "topology/pod-1/node-17/sys",
          "fabricId" : "0",
          "id" : "17",
          "inbMgmtAddr" : "0.0.0.0",
          "lcOwn" : "local",
          "mode" : "unspecified",
          "name" : "leaf0",
          "nodeId" : "0",
          "oobMgmtAddr" : "0.0.0.0",
          "podId" : "1",
          "replTs" : "never",
          "role" : "leaf",
          "serial" : "FOX-270308",
          "status" : "",
          "systemUpTime" : "00:00:02:03"
        }
      }
    }, {
      "topSystem" : {
        "attributes" : {
          "instanceId" : "0:0",
          "address" : "10.0.0.1",
          "childAction" : "",
          "currentTime" : "2013-06-14T04:13:29.301",
          "currentTimeZone" : "PDT",
          "dn" : "topology/pod-1/node-1/sys",
          "fabricId" : "0",
          "id" : "1",
          "inbMgmtAddr" : "0.0.0.0",
          "lcOwn" : "local",
          "mode" : "unspecified",
          "name" : "apic0",
          "nodeId" : "0",
          "oobMgmtAddr" : "0.0.0.0",
          "podId" : "0",
          "replTs" : "never",
          "role" : "apic",
          "serial" : "",
          "status" : "",
          "systemUpTime" : "00:00:02:26"
        }
      }
    }
  ]
}
```


この応答は、システムが 1 つの APIC（ノード 1）と 1 つのリーフ（ノード 17）で構成されることを示します。

例：JSON API を使用したノードに関する情報の取得

次に、システムのノードにアクセスするために APIC を照会する例を示します。

ファブリックの特定のノード デバイスに API 操作を指示するために、リソースのパスは `/mo/topology/pod-number/node-number/sys/` とその後に続くノード コンポーネントで構成されま

す。 たとえば、この URI はノード 1 のシャーシ スロット B のボード センサー 3 にアクセスします。

```
GET http://192.0.20.123/api/mo/topology/pod-1/node-1/sys/ch/bslot/board/sensor-3.json
```

正常な動作時には次のような応答本文が返されます。

```
{
  "imdata" :
  [{
      "eqptSensor" : {
        "attributes" : {
          "instanceId" : "0:0",
          "childAction" : "",
          "dn" : "topology/pod-1/node-1/sys/ch/bslot/board/sensor-3",
          "id" : "3",
          "majorThresh" : "0",
          "mfgTm" : "not-applicable",
          "minorThresh" : "0",
          "model" : "",
          "monPolDn" : "",
          "rev" : "0",
          "ser" : "",
          "status" : "",
          "type" : "dimm",
          "vendor" : "Cisco Systems, Inc."
        }
      }
    }
  ]
}
```

例：JSON API を使用した実行中のファームウェアの取得

次に、実行中のファームウェア イメージを判断するために APIC を照会する例を示します。

実行中のファームウェアの詳細情報は、APIC ファームウェアのステータス コンテナ クラス firmware:IfcFwStatusCont の子クラス（サブツリー）であるクラスのオブジェクト firmware:IfcRunning に含まれています。 実行中のファームウェア インスタンスが複数存在する場合があるので（APIC インスタンス 1 つにつき 1 個）、コンテナ クラスを照会し、実行中のファームウェア オブジェクトのサブツリーの応答をフィルタリングできます。

次に、API クエリー メッセージの例を示します。

```
GET http://192.0.20.123/api/class/firmwareIfcFwStatusCont.json?
        query-target=subtree
        &
        target-subtree-class=firmwareIfcRunning
```

正常な動作時には次のような応答本文が返されます。

```
{
  "imdata" : [{
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "3",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-3",
          "lcOwn" : "local",
```

```
          "replTs" : "never",
          "rn" : "",
          "status" : "",
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }, {
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "1",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-1",
          "lcOwn" : "local",
          "replTs" : "never",
          "rn" : "",
          "status" : "",
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }, {
      "firmwareIfcRunning" : {
        "attributes" : {
          "instanceId" : "0:0",
          "applId" : "2",
          "childAction" : "",
          "dn" : "ifcfwstatuscont/ifcrunning-2",
          "lcOwn" : "local",
          "replTs" : "never",
          "rn" : "",
          "status" : "",
          "ts" : "2012-12-31T16:00:00.000",
          "type" : "ifc",
          "version" : "1.1"
        }
      }
    }
  ]
}
```


この応答は、APIC ファームウェア バージョン 1.1 の 3 つの実行中のインスタンスについて記述しています。

例：JSONAPIを使用したリーフポートセレクタ プロファイルの追加

次に、リーフ ポート セレクタ プロファイルを追加する例を示します。

『*Cisco APIC Management Information Model Reference*』に示すように、このクラスの階層はリーフ ポート セレクタ プロファイルを形成します。

- fabric:LePortP：リーフ ポート プロファイルは、このクラスの管理対象オブジェクト（MO）で表され、`uni/fabric/leportp-[name]` の識別名（DN）形式（`leportp-[name]` は相対名（RN））を持ちます。リーフ ポート プロファイル オブジェクトは、子オブジェクトとしてリーフ ポート セレクタを含むことができるテンプレートです。

- fabric:LFPortS：リーフ ポート セレクタは、このクラスの MO で表され、`lefabports-[name]-typ-[type]` の RN 形式を持ちます。 リーフ ポート セレクタ オブジェクトには、子オブジェクトとして 1 つ以上のポートまたはポートの範囲を含めることができます。
  - fabric:PortBlk：リーフ ポートまたはリーフ ポートの範囲は、このクラスの MO で表され、`portblk-[name]` の RN 形式を持ちます。

新しいリーフ ポート セレクタ プロファイル MO を作成する API コマンドは、子 MO を作成および設定することもできます。

次の例では、「MyLPSelectorProf」という名前でリーフ ポート セレクタ プロファイルを作成します。この例のプロファイルには、リーフ スイッチ 1 上でリーフ ポート 1 を、リーフ スイッチ 1 上でリーフ ポート 3 ～ 5 を選択する「MySelectorName」という名前のセレクタが含まれます。 新しいプロファイルを作成および設定するには、この HTTP POST メッセージを送信します。

```
POST http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json
```

```
{
  "fabricLePortP" : {
    "attributes" : {
      "descr" : "Selects leaf ports 1/1 and 1/3-5"
    },
    "children" : [{
        "fabricLFPortS" : {
          "attributes" : {
            "name" : "MySelectorName",
            "type" : "range"
          },
          "children" : [{
              "fabricPortBlk" : {
                "attributes" : {
                  "fromCard" : "1",
                  "toCard" : "1",
                  "fromPort" : "1",
                  "toPort" : "1",
                  "name" : "block2"
                }
              }
            }, {
              "fabricPortBlk" : {
                "attributes" : {
                  "fromCard" : "1",
                  "toCard" : "1",
                  "fromPort" : "3",
                  "toPort" : "5",
                  "name" : "block3"
                }
              }
            }
          ]
        }
      }
    ]
  }
}
```

正常な動作時には次の応答本文が返されます。

```
{
  "imdata" : [{
      "fabricLePortP" : {
```

```
            "attributes" : {
              "instanceId" : "0:0",
              "childAction" : "deleteNonPresent",
              "descr" : "Select leaf ports 1/1 and 1/3-5",
              "dn" : "uni/fabric/leportp-MyLPSelectorProf",
              "lcOwn" : "local",
              "name" : "MyLPSelectorProf",
              "replTs" : "never",
              "rn" : "",
              "status" : "created"
            },
            "children" : [{
                "fabricLFPortS" : {
                  "attributes" : {
                    "instanceId" : "0:0",
                    "childAction" : "deleteNonPresent",
                    "dn" : "",
                    "lcOwn" : "local",
                    "name" : "MySelectorName",
                    "replTs" : "never",
                    "rn" : "lefabports-MySelectorName-typ-range",
                    "status" : "created",
                    "type" : "range"
                  },
                  "children" : [{
                      "fabricPortBlk" : {
                        "attributes" : {
                          "instanceId" : "0:0",
                          "childAction" : "deleteNonPresent",
                          "dn" : "",
                          "fromCard" : "1",
                          "fromPort" : "3",
                          "lcOwn" : "local",
                          "name" : "block3",
                          "replTs" : "never",
                          "rn" : "portblk-block3",
                          "status" : "created",
                          "toCard" : "1",
                          "toPort" : "5"
                        }
                      }
                    }, {
                      "fabricPortBlk" : {
                        "attributes" : {
                          "instanceId" : "0:0",
                          "childAction" : "deleteNonPresent",
                          "dn" : "",
                          "fromCard" : "1",
                          "fromPort" : "1",
                          "lcOwn" : "local",
                          "name" : "block2",
                          "replTs" : "never",
                          "rn" : "portblk-block2",
                          "status" : "created",
                          "toCard" : "1",
                          "toPort" : "1"
                        }
                      }
                    }
                  ]
                }
              }
            ]
          }
        }
      ]
    }
```

新しいプロファイルを削除するには、次の例のように、"status":"deleted" の fabricLePortP 属性を持つ HTTP POST メッセージを送信します。

```
POST http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json

{
  "fabricLePortP" : {
    "attributes" : {
      "status" : "deleted"
    }
  }
}
```

または、この HTTP DELETE メッセージを送信できます。

```
DELETE http://192.0.20.123/api/mo/uni/fabric/leportp-MyLPSelectorProf.json
```