



## トラブルシューティング用のクエリー API

この章では、個々の Cisco Prime Network Control System (NCS) REST API コールの使用法について例をあげながら説明します。

### Cisco Prime NCS API コール

Cisco Prime NCS API コールはノードのバージョンおよびタイプ、障害の理由、認証ステータスとアカウント ステータスを含むターゲット Cisco Monitoring ISE ノードのセッションに関する主要なトラブルシューティング情報を取得するためのメカニズムを提供します。

### クエリー API を使用した Cisco ISE のトラブルシューティング

Cisco Prime NCS トラブルシューティング API コールは、Cisco ISE 展開のターゲット Cisco Monitoring ISE ノードにステータス要求を送信し、次の診断関連情報を取得します。

- ノードのバージョンおよびタイプ (Version API コールを使用)
- 障害理由 (FailureReasons API コールを使用)
- 認証ステータス (AuthStatus API コールを使用)
- アカウンティング ステータス (AcctStatus API コールを使用)

### ノードのバージョンおよびタイプの API コール

各ノードの REST Programmatic インターフェイス (PI) サービスとクレデンシャルをテストするには Version API コールを使用できます。ここでは、スキーマ ファイルの出力例、この API コールを呼び出すことにより、Cisco ISE ソフトウェアのバージョンおよびノード タイプを要求する手順、この API コール発行後に返されるノードのバージョンとタイプのサンプルについて説明します。

ノード タイプは次のいずれかになります。

- STANDALONE\_MNT\_NODE = 0
- ACTIVE\_MNT\_NODE = 1
- BACKUP\_MNT\_NODE = 2
- NOT\_AN\_MNT\_NODE = 3

## Version API の出力スキーマ


このサンプル スキーマ ファイルは、ターゲット Cisco Monitoring ISE ノードへの送信後の、バージョン API コールの出力です。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="product" type="product"/>

  <xs:complexType name="product">
    <xs:sequence>
      <xs:element name="version" type="xs:string" minOccurs="0"/>
      <xs:element name="type_of_node" type="xs:int"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

## バージョン API コールの呼び出し

- 
- ステップ 1** Cisco ISE URL をブラウザのアドレス バーに入力します(たとえば `https://<ise hostname or ip address>/admin/`)。
- ステップ 2** ユーザ名と、Cisco ISE の初期セットアップで指定および設定した大文字と小文字が区別されるパスワードを入力します。
- ステップ 3** [ログイン(Login)] をクリックするか、**Enter** を押します。
- ログインが失敗した場合は、[ログイン時の問題(Problem logging in?)] [ステップ 2](#) に従ってください。
- たとえば、ホスト名が `acme123` の Cisco Monitoring ISE ノードに最初にログインする場合、このノードの URL アドレスが次のように表示されます。
- ```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```
- ステップ 4** 「/admin/」コンポーネントを API コールのコンポーネント (`/admin/API/mnt/<specific-api-call>`) に置き換えて、ターゲット ノードの URL アドレス フィールドに Version API コールを入力します。
- ```
https://acme123/admin/API/mnt/Version
```
- 
-  **(注)** これらのコールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、Cisco Monitoring ISE ノードを表します。
- 
- ステップ 5** **Enter** キーを押して API コールを発行します。
- 

### 関連項目

- [モニタリング ノードの確認\(1-2 ページ\)](#)

## Version API コールから返されるサンプルデータ

次に、ターゲット Cisco Monitoring ISE ノードで Version API コールを呼び出すときに返されるデータを示します。この API コールでは、ターゲット ノードについて次の 2 種類の値が返されます。

- ノードのバージョン(この例では、1.0.3.032 を表示します)。
- Cisco Monitoring ISE ノードのタイプ(この例では、アクティブな Cisco Monitoring ISE ノードが 1 つであることを意味する「1」を表示します)。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-
<product name="Cisco Identity Services Engine">
<version>1.0.3.032</version>
<type_of_node>1</type_of_node>
</product>
```

## 障害理由 API コール

ターゲット ノードで行われた認証ステータスのチェックで返された障害理由のリストを返すために FailureReasons API コールを使用できます。ここでは、スキーマファイルの出力例、この API コールを呼び出すことにより、Cisco Monitoring ISE ノードで記録される障害理由のリストを要求する手順、この API コール発行後に返される障害理由のサンプルについて説明します。返される障害理由は、それぞれ表 3-1 に示す次の要素で構成されます。



(注)

Cisco ISE Failure Reasons Editor を使用して障害理由の完全なリストにアクセスする方法に関する詳細については、[Cisco ISE 障害理由レポート \(A-1 ページ\)](#) を参照してください。

表 3-1 Cisco Identity Services Engine の製品マニュアル

| 障害理由の要素 | 例   |
|---------|---|
| 障害理由 ID | <failureReason id="11011">  |
| コード     | <11011 RADIUS listener failed>  |
| 原因      | <Could not open one or more of the ports used to receive RADIUS requests>                             |
| 解像度     | <Ensure that the ports 1812, 1813, 1645 and 1646 are not being used by another process on the system> |



(注)

Cisco ISE ユーザーインターフェイスを使用して([モニタ (Monitor)]>[レポート (Reports)]>[カタログ (Catalog)]>[障害理由 (Failure Reasons)]) をクリックして障害理由レポートがあるかどうかを確認します。障害理由レポートが表示されます。

## FailureReasons API の出力スキーマ

このサンプル スキーマ ファイルは、ターゲット Cisco Monitoring ISE ノードへの要求の送信後の、FailureReasons API コールの出力です。


```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="failureReasonList" type="failureReasonList"/>

  <xs:complexType name="failureReasonList">
    <xs:sequence>
      <xs:element name="failureReason" type="failureReason" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="failureReason">
    <xs:sequence>
      <xs:element name="code" type="xs:string" minOccurs="0"/>
      <xs:element name="cause" type="xs:string" minOccurs="0"/>
      <xs:element name="resolution" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

## FailureReasons API コールの呼び出し

- 
- ステップ 1** Cisco ISE URL をブラウザのアドレス バーに入力します(たとえば `https://<ise hostname or ip address>/admin/`)。
- ステップ 2** ユーザ名と、Cisco ISE の初期セットアップで指定および設定した大文字と小文字が区別されるパスワードを入力します。
- ステップ 3** [ログイン(Login)] をクリックするか、**Enter** を押します。
- ログインが失敗した場合は、[ログイン時の問題(Problem logging in?)] [ステップ 2](#) に従ってください。
- たとえば、ホスト名が `acme123` の Cisco Monitoring ISE ノードに最初にログインする場合、このノードの URL アドレスが次のように表示されます。
- ```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```
- ステップ 4** 「/admin/」コンポーネントを API コールのコンポーネント (`/admin/API/mnt/<specific-api-call>`) に置き換えて、ターゲット ノードの URL アドレス フィールドに FailureReasons API コールを入力します。
- ```
https://acme123/admin/API/mnt/FailureReasons
```
-  **(注)** これらのコールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、Cisco Monitoring ISE ノードを表します。
- 
- ステップ 5** **Enter** キーを押して API コールを発行します。
-

## 関連項目

- [モニタリング ノードの確認\(1-2 ページ\)](#)

## FailureReasons API コールから返されるサンプルデータ

次に、ターゲット Cisco Monitoring ISE ノードで FailureReasons API コールを呼び出すときに返されるデータを示します。この API コールは、ターゲット ノードから障害理由のリストを返します。障害理由は、それぞれ、障害 ID、障害コード、原因、対処法(既知の場合)によって定義されます。



(注) 次の FailureReasons API コールの例は、返されるデータの小規模なサンプルを表示しています。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-
<failureReasonList>
-
<failureReason id="100001">
-
<code>
100001 AUTHMGR-5-FAIL Authorization failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100002">
-
<code>
100002 AUTHMGR-5-SECURITY_VIOLATION Security violation on the interface
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100003">
-
<code>
100003 AUTHMGR-5-UNAUTHORIZED Interface unauthorized
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100004">
-
<code>
100004 DOT1X-5-FAIL Authentication failed for client
</code>
```

```

<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100005">
<code>100005 MAB-5-FAIL Authentication failed for client</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100006">
-
<code>
100006 RADIUS-4-RADIUS_DEAD RADIUS server is not responding
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100007">
-
<code>
100007 EPM-6-POLICY_APP_FAILURE Interface ACL not configured
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>

```

#### 関連項目

- [モニタリング ノードの確認\(1-2 ページ\)](#)
- [付録 A「Cisco ISE 障害理由レポート」](#)

## 認証ステータス API コール

ターゲット ノードのセッションの認証ステータスをチェックするために **AuthStatus API** 呼び出しを使用できます。この API コールに関連付けられたクエリーには、一致の検索対象である **MAC アドレス** が少なくとも 1 つ必要です。指定の **MAC アドレス** が返されるように、最新レコードに、ユーザ設定が可能な制限を付けます。

ここでは、スキーマ ファイルの出力例、この API コールを呼び出すことにより、ターゲットのモニタリング モードでセッション認証のステータスを検索する要求を送信する手順、この API コール発行後に返されるデータのサンプルについて説明します。

AuthStatus API コールにより、次の検索関連パラメータを設定できるようになります。

- 期間: 指定された MAC アドレスに関連付けられた認証ステータス レコードの検索と取得が試行される秒数を定義します。ユーザが設定可能な値の有効範囲は 1 ~ 864000 秒(10 日)です。0 秒の値を入力した場合は、デフォルト期間の 10 日を指定します。
- レコード: MAC アドレスごとに検索するセッションのレコード数を定義します。ユーザが設定可能な値の有効範囲は 1 ~ 500 レコードです。0 を入力した場合は、デフォルト設定の 200 レコードを指定します。



(注) 期間およびレコード パラメータの両方に値 0 を指定すると、この API コールは、指定された MAC アドレスに関連付けられている最新の認証セッション レコードのみを返します。

ここに、期間とレコードの属性を指定した URL の一般的な形式の例を示します。

`https://10.10.10.10/admin/API/mnt/AuthStatus/MACAddress/01:23:45:67:89:98/900000/2/All`

- 属性: AuthStatus API コールを使用して認証ステータスの検索で返された認証ステータスのテーブルの属性数を定義します。有効な値は 0(デフォルト)、All、または `user_name+acs_timestamp` です (AuthStatus スキーマの例 ([AcctStatus API の出力スキーマ \(3-13 ページ\)](#)) を参照)。
  - 「0」を入力すると、表 3-2 で定義された属性が返されます。これらは出力スキーマの `restAuthStatus` のセクションに記載されています。
  - 「All」を入力すると、より詳しい属性セットが返されます。これらは出力スキーマの `fullRESTAuthStatus` のセクションに記載されています。
  - `user_name+acs_timestamp` のスキーマに示されている値を入力すると、それらの属性だけが返されます。`user_name` 属性と `acs_timestamp` 属性は、出力スキーマ `restAuthStatus` のセクションに記載されています。

表 3-2 認証ステータス テーブルの属性

| 属性  | 説明   |
|---|--|
| <code>name="passed"</code> または <code>name="failed"</code> | 認証ステータスの結果: <ul style="list-style-type: none"> <li>• パス (Passed)</li> <li>• 失敗しました (Failed)</li> </ul> |
| <code>name="user_name"</code>                             | ユーザ名   |
| <code>name="nas_ip_address"</code>                        | ネットワーク アクセス デバイスの IP アドレス/ホスト名   |
| <code>name="nas_ipv6_address"</code>                      | ネットワーク アクセス デバイスの IPv6 アドレス/ホスト名   |
| <code>name="failure_reason"</code>                        | セッション認証に失敗した原因   |
| <code>name="calling_station_id"</code>                    | ソース IP アドレス  |
| <code>name="nas_port"</code>                              | ネットワーク アクセス サーバ ポート  |
| <code>name="identity_group"</code>                        | 関連するユーザとホストで構成される論理グループ  |
| <code>name="network_device_name"</code>                   | ネットワーク デバイスの名前   |
| <code>name="acs_server"</code>                            | Cisco ISE アプライアンスの名前   |
| <code>name="eap_authentication"</code>                    | 認証要求に使用する拡張認証プロトコル (EAP) 方式  |
| <code>name="framed_ip_address"</code>                     | 特定のユーザに設定されたアドレス   |

表 3-2 認証ステータス テーブルの属性(続き)

| 属性                            | 説明                                    |
|-------------------------------|---------------------------------------|
| name="framed_ipv6_address"    | 特定のユーザに設定されたアドレス                      |
| network_device_groups"        | 関連するネットワーク デバイスで構成される論理グループ           |
| name="access_service"         | アプリケーションアクセス サービス                     |
| name="acs_timestamp"          | Cisco ISE 認証要求に関連付けられたタイム スタンプ        |
| name="authentication_method"  | 認証で使用される方式を指定します                      |
| name="execution_steps"        | 要求の処理中にログに記録された各診断メッセージのメッセージ コードのリスト |
| name="radius_response"        | RADIUS 応答のタイプ(例: VLAN、ACL)            |
| name="audit_session_id"       | 認証セッションの ID                           |
| name="nas_identifier"         | 特定のリソースに関連付けられているネットワーク アクセス サーバ(NAS) |
| name="nas_port_id"            | 使用される NAS ポート ID                      |
| name="nac_policy_compliance"  | ポスチャ ステータスを示します(準拠または非準拠)             |
| name="selected_azn_profiles"  | 認証に使用されるプロファイルを指定します                  |
| name="service_type"           | フレーム ユーザを示します                         |
| name="eap_tunnel"             | EAP 認証に使用されるトンネルまたは外部方式               |
| name="message_code"           | 処理された要求の結果を定義する監査メッセージの ID            |
| name="destination_ip_address" | 宛先 IP アドレスを指定します                      |

## AuthStatus API の出力スキーマ

このサンプル スキーマ ファイルは、ターゲット Cisco Monitoring ISE ノードでの指定されたセッションへの送信後の、AuthStatus API コールの出力です。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authStatusOutputList" type="fullRESTAuthStatusOutputList"/>

  <xs:complexType name="fullRESTAuthStatusOutputList">
    <xs:sequence>
      <xs:element name="authStatusList" type="fullRESTAuthStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatusList">
    <xs:sequence>
      <xs:element name="authStatusElements" type="fullRESTAuthStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="key" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatus">
    <xs:complexContent>
      <xs:extension base="restAuthStatus">
        <xs:sequence>
```



```

<xs:element name="id" type="xs:long" minOccurs="0"/>
<xs:element name="acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
<xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
<xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
<xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
<xs:element name="identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="response" type="xs:string" minOccurs="0"/>
<xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
<xs:element name="use_case" type="xs:string" minOccurs="0"/>
<xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
<xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
<xs:element name="acs_username" type="xs:string" minOccurs="0"/>
<xs:element name="radius_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_role" type="xs:string" minOccurs="0"/>
<xs:element name="nac_username" type="xs:string" minOccurs="0"/>
<xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
<xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
<xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
<xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
<xs:element name="authentication_identity_store" type="xs:string"
minOccurs="0"/>
  <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
  <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
  <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
  <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
  <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
  <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
  <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
  <xs:element name="selected_query_identity_stores" type="xs:string"
minOccurs="0"/>
  <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
  <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
  <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
  <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
  <xs:element name="response_time" type="xs:long" minOccurs="0"/>
  <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="restAuthStatus">
  <xs:sequence>
    <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
    <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
    <xs:element name="user_name" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
    <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
    <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
    <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
    <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
    <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
    <xs:element name="eap_authentication" type="xs:string" minOccurs="0"/>
    <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
    <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
    <xs:element name="access_service" type="xs:string" minOccurs="0"/>
    <xs:element name="acs_timestamp" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
    <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
    <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
    <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
    <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
  
```

```

<xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
<xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
<xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
<xs:element name="service_type" type="xs:string" minOccurs="0"/>
<xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
<xs:element name="message_code" type="xs:string" minOccurs="0"/>
<xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

<xs:element name="nas_ipv6_address" type="xs:string"/>
<xs:complexType name="framed_ipv6_address_list">
  <xs:sequence minOccurs="0" maxOccurs="8"><xs:element name="ipv6_address"
type="xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:element name="framed_ipv6_address" type="framed_ipv6_address_list" minOccurs="1"
maxOccurs="1"/>
</xs:schema>

```

## AuthStatus API コールの呼び出し

- ステップ 1** Cisco ISE URL をブラウザのアドレス バーに入力します(たとえば `https://<ise hostname or ip address>/admin/`)。
- ステップ 2** ユーザ名と、Cisco ISE の初期セットアップで指定および設定した大文字と小文字が区別されるパスワードを入力します。
- ステップ 3** [ログイン(Login)] をクリックするか、**Enter** を押します。
- ログインが失敗した場合は、[ログイン時の問題(Problem logging in?)] [ステップ 2](#) に従ってください。

たとえば、ホスト名が `acme123` の Cisco Monitoring ISE ノードに最初にログインする場合、このノードの URL アドレスが次のように表示されます。

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

- ステップ 4** 「/admin/」コンポーネントを API コールのコンポーネント (`/admin/API/mnt/<specific-api-call>/MACAddress/<macaddress>/<seconds>/<numberofrecordspermacaddress>/All`) に置き換えて、ターゲット ノードの URL アドレス フィールドに AuthStatus API コールを入力します。
- ```
https://acme123/admin/API/mnt/AuthStatus/MACAddress/00:50:56:10:13:02/120/100/All
```



(注) REST API コールは大文字と小文字を区別します。API コール規則での「mnt」の使用は、Cisco Monitoring ISE ノードを表します。

- ステップ 5** **Enter** キーを押して API コールを発行します。

### 関連項目

- [モニタリング ノードの確認\(1-2 ページ\)](#)

## AuthStatus API コールから返されるサンプル データ

次に、ターゲット Cisco Monitoring ISE ノードで AuthStatus API コールを呼び出すときに返されるデータを示します。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-
<authStatusOutputList>
-
<authStatusList key="00:0C:29:46:F3:B8"><authStatusElements>
-
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>suser77</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>User Identity Groups:Guest</identity_group>
<acs_server>guest-240</acs_server>
<acs_timestamp>2012-10-05T10:50:56.515Z</acs_timestamp>
<execution_steps>5231</execution_steps>
<message_code>5231</message_code>
<id>1349422277270561</id>
<acsview_timestamp>2012-10-05T10:50:56.517Z</acsview_timestamp>
<identity_store>Internal Users</identity_store>
<response_time>146</response_time>
<other_attributes>ConfigVersionId=81,EndPointMACAddress=00-0C-29-46-F3-B8,PortalName=DefaultGuestPortal,
CPMSessionID=0A4D98D1000001F26F0C04D9,CiscoAVPair=</other_attributes>
</authStatusElements>
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>00:0C:29:46:F3:B8</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<nas_ipv6_address>2001:cdba::3257:9652</nas_ipv6_address>
<framed_ipv6_address>
<ipv6_address>2001:cdba:0000:0000:0000:0000:3257:9652</ipv6_address>
<ipv6_address> 2001:cdba:0:0:0:0:3257:9652</ipv6_address>
<ipv6_address>2001:cdba::3257:9652</ipv6_address>
</framed_ipv6_address>
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>Guest_IDG</identity_group>
<network_device_name>switch</network_device_name>
<acs_server>guest-240</acs_server>
<authentication_method>mab</authentication_method>
<authentication_protocol>Lookup</authentication_protocol>
<acs_timestamp>2012-10-05T10:49:47.915Z</acs_timestamp>
<execution_steps>11001,11017,11027,15049,15008,15048,15048,15004,15041,15006,15013,24209,2421
1,22037,15036,15048,15004,15016,11022,11002</execution_steps>
<response>{UserName
=00:0C:29:46:F3:B8; User-Name=00-0C-29-46-F3-B8;
State=ReauthSession:0A4D98D1000001F26F0C04D9;
Class=CACS:0A4D98D1000001F26F0C04D9:guest-240/138796808/76;
Termination-Action=RADIUS-Request; Tunnel-Type=(tag=1) VLAN;
Tunnel-Medium-Type=(tag=1) 802; Tunnel-Private-Group-ID=(tag=1) 2;
cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://guest-240.cisco.com:8443/guestportal/gateway?
sessionId=0A4D98D1000001F26F0C04D9&action=cwa;
```

```

cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-pre-posture-506e980a;
cisco-av-pair=profile-name=WindowsXP-Workstation;}</response
><audit_session_id>0A4D98D1000001F26F0C04D9</audit_session_id><nas_po
rt_id>GigabitEthernet1/0/17</nas_port_id><posture_status>Pending</posture_status>
<selected_azn_profiles>CWA_Redirect</selected_azn_profiles>
<service_type>Call Check</service_type>
<message_code>5200</message_code>
<nac_policy_compliance>Pending</nac_policy_compliance>
<id>1349422277270556</id>
<acsview_timestamp>2012-10-05T10:49:47.915Z</acsview_timestamp>
<identity_store>Internal Endpoints</identity_store>
<response_time>13</response_time>
<other_attributes>ConfigVersionId=81, DestinationPort=1812, Protocol=Radius, AuthorizationPol
icyMatchedRule=CWA_Redirect,
NAS-Port=50117, Framed-MTU=1500, NAS-Port-Type=Ethernet, EAP-Key-N
ame=, cisco-nas-port=GigabitEthernet1/0/17, AcsSessionID=guest-240/138796808/76, Us
eCase=Host Lookup, SelectedAuthenticationIdentityStores=Internal
Endpoints, ServiceSelectionMatchedRule=MAB, IdentityPolicyMatchedRule=Default, CPMS
essionID=0A4D98D1000001F26F0C04D9, EndPointMACAddress=00-0C-29-46-F3-B8, EndPointM
atchedProfile=WindowsXP-Workstation, ISEPolicySetName=Default, HostIdentityGroup=E
ndpoint Identity Groups:Guest_IDG, Device Type=Device Type#All Device
Types, Location=Location#All Locations, Device IP
Address=10.77.152.209, Called-Station-ID=00:24:F7:73:9A:91, CiscoAVPair=audit-sess
ion-id=0A4D98D1000001F26F0C04D9</other_attributes>
-
</authStatusElements>
-
</authStatusList>
-
</authStatusOutputList>

```

## アカウントステータス API コール

ターゲット ノードの最新のデバイスおよびセッションのアカウント情報を取得するために AcctStatus API コールを使用できます。ここでは、スキーマ ファイルの出力例、この API コールを呼び出すことにより、最新のデバイスおよびセッション情報の要求を送信する手順、この API コール発行後に返されるデータのサンプルについて説明します。AcctStatus API コールにより、時間関連パラメータを設定できるようになります。

- 期間: 指定された MAC アドレスに関連付けられた最新アカウントのデバイス レコードの検索と取得が試行される秒数を定義します。ユーザが設定可能な値の有効範囲は 1 ~ 432000 秒 (5 日) です。次に例を示します。
  - 2400 秒 (40 分) の値を入力した場合は、過去 40 分間に使用可能な指定 MAC アドレスの最新アカウントのデバイス レコードが必要であることを意味します。
  - 0 秒の値を入力した場合は、デフォルト期間の 15 分 (900 秒) を指定します。これは、この時間内に使用可能な指定 MAC アドレスの最新アカウントのデバイス レコードが必要であることを意味します。

AcctList API コールは、API 出力として、次のアカウント ステータスのデータ フィールドを提供します (表 3-3 を参照)。

表 3-3 アカウンティング ステータスのデータ フィールド

| データ フィールド              | 説明               |
|------------------------|------------------|
| MAC アドレス (MAC address) | クライアントの MAC アドレス |
| 監査セッション ID             | 監査セッション ID       |

表 3-3 アカウンティングステータスのデータ フィールド(続き)

| データ フィールド    | 説明           |
|--------------|--------------|
| Packets in   | 受信したパケットの合計数 |
| Packets out  | 送信したパケットの合計数 |
| Bytes in     | 受信したバイトの合計数  |
| Bytes out    | 送信したバイトの合計数  |
| Session time | 現在のセッションの期間  |

## AcctStatus API の出力スキーマ

このサンプルスキーマ ファイルは、ターゲット Cisco Monitoring ISE ノードでの指定されたセッションへの送信後の、AcctStatus API コールの出力です。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">


  <xs:element name="acctStatusOutputList" type="restAcctStatusOutputList"/>

  <xs:complexType name="restAcctStatusOutputList">
    <xs:sequence>
      <xs:element name="acctStatusList" type="restAcctStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="restAcctStatusList">
    <xs:sequence>
      <xs:element name="acctStatusElements" type="restAcctStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="username" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="restAcctStatus">
    <xs:sequence>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="paks_in" type="xs:long" minOccurs="0"/>
      <xs:element name="paks_out" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_in" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_out" type="xs:long" minOccurs="0"/>
      <xs:element name="session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="username" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

## AcctStatus API コールの呼び出し

- ステップ 1** Cisco ISE URL をブラウザのアドレス バーに入力します(たとえば `https://<ise hostname or ip address>/admin/`)。
- ステップ 2** ユーザ名と、Cisco ISE の初期セットアップで指定および設定した大文字と小文字が区別されるパスワードを入力します。
- ステップ 3** [ログイン(Login)] をクリックするか、**Enter** を押します。  
ログインが失敗した場合は、[ログイン時の問題(Problem logging in?)] [ステップ 2](#) に従ってください。  
たとえば、ホスト名が `acme123` の Cisco Monitoring ISE ノードに最初にログインする場合、このノードの URL アドレスが次のように表示されます。  
`https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash`
- ステップ 4** 「/admin/」コンポーネントを API コールのコンポーネント (`/admin/API/mnt/<specific-api-call>/MACAddress/<macaddress>/<durationofcurrenttime>`) に置き換えて、ターゲット ノードの URL アドレス フィールドに AcctStatus API コールを入力します。  
`https://acme123/admin/API/mnt/AcctStatus/MACAddress/00:26:82:7B:D2:51/1200`
-  (注) これらのコールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、Cisco Monitoring ISE ノードを表します。
- ステップ 5** **Enter** キーを押して API コールを発行します。

### 関連項目

- [モニタリング ノードの確認\(1-2 ページ\)](#)

## AcctStatus API コールから返されるサンプルデータ

次に、ターゲット Cisco Monitoring ISE ノードで AcctStatus API コールを呼び出すときに返されるデータを示します。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
-
<acctStatusOutputList>
-
<acctStatusList macAddress="00:25:9C:A3:7D:48">
-
<acctStatusElements>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<audit_session_id>0acb6b0b0000000B4D0C0DBD</audit_session_id>
<paks_in>0</paks_in>
<paks_out>0</paks_out>
<bytes_in>0</bytes_in>
<bytes_out>0</bytes_out>
<session_time>240243</session_time>
<server>HAREESH-R6-1-PDP1</server>
</acctStatusElements>
</acctStatusList>
</acctStatusOutputList>
```