Cisco Identity Services Engine API リファレンス ガイド リリース 1.3

２０１５年３月５日

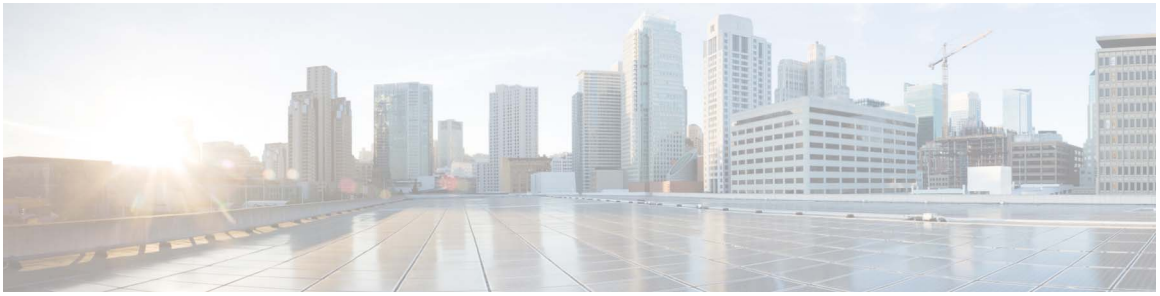
Cisco Systems, Inc.
www.cisco.com

シスコは世界各国 200 箇所にオフィスを開設しています。
所在地、電話番号、FAX 番号は以下の
シスコ Web サイトをご覧ください。
www.cisco.com/go/offices

Text Part Number:OL-26134-01-J

Cisco Identity Services Engine API リファレンス ガイド リリース 1.3

目 次



はじめに

改訂日：15/3/5、OL-26134-01-J

ここでは、『*Cisco Identity Services Engine API リファレンス ガイド リリース 1.3*』の目的、対象読者、および構成について説明します。また、指示を記述する表記法について説明し、次のセクションでは他の種類の情報を説明します。

Cisco Identity Services Engine の概要

Cisco Identity Services Engine（ISE）は、企業でのコンプライアンスの順守、インフラストラクチャのセキュリティの強化、サービス オペレーションの合理化を実現する、次世代のアイデンティティおよびアクセス コントロール ポリシーのプラットフォームです。Cisco ISE の固有のアーキテクチャにより、企業は、アクセス スイッチ、Wireless LAN Controller（WLC）、バーチャル プライベート ネットワーク（VPN）ゲートウェイ、およびデータセンター スイッチなど、さまざまなネットワーク要素に ID を結びつけることで予防的な管理を決定するために、ネットワーク、ユーザ、およびデバイスからリアルタイムのコンテキスト情報を収集することができるようになります。

Cisco ISE は Cisco Security Group Access Solution のキー コンポーネントです。Cisco ISE は、統合されたポリシーベースのアクセス コントロール ソリューションで以下を実現します。

- 認証、承認、アカウント（AAA）、ポスチャ、プロファイラ、ゲスト管理サービスを 1 つのアプライアンスに結合します。
- 802.1X 環境を含むネットワークにアクセスしているすべてのエンドポイントのデバイス ポスチャをチェックすることでエンドポイント コンプライアンスを徹底します。
- ネットワーク上のエンドポイント デバイスの検出、プロファイリング、ポリシーベースの配置、モニタリングのサポートを提供します。

- 集中型展開および分散型展開においてポリシーの一貫性が維持され、サービスを必要な場所に配信できるようになります。
- Security Group Tags（SGT）および Security Group（SG）Access Control List（ACL）によって Security Group Access（SGA）などの高度な強化機能を使用します。
- 小さな事務所から大企業までさまざまな環境の展開シナリオに対応するスケーラビリティをサポートします。

Cisco ISE のアーキテクチャは、集中型ポータルからネットワークを設定して管理できるように、スタンドアロンの導入と分散型の導入をサポートします。Cisco ISE の機能の詳細については、『*Cisco Identity Services Engine Admin Guide, Release 1.3*』を参照してください。

目的

このアプリケーション プログラミング インターフェイス（API）リファレンス ガイドは、サポート対象の API が提供する機能の概要だけを説明します。この API リファレンス ガイドの目的は、Cisco ISE 展開内で概説された API を使用するための基本的な注意事項を、開発者、システム管理者やネットワーク管理者、またはシステム インテグレータに提供することです。

REST API コールは、次の種類のデータを確認するためにクエリーを使用します。

- アクティブ セッションの数
- アクティブ セッションのタイプ
- アクティブ セッションの認証ステータス
- 使用中の MAC アドレス
- 使用中の NAS の IP アドレス
- ノードのバージョンとタイプ
- ノードのセッション障害の理由

外部 RESTful サービス API および関連 API コールは、Cisco ISE リソースに対して CRUD（作成、読み取り、更新、削除）操作を実行するために使用できます。外部 RESTful サービスは HTTP プロトコルおよび REST 方法論に基づいています。

コメント Cisco ISE ネットワークとそのノードおよびペルソナ、動作または用途の概念、Cisco ISE ユーザ インターフェイスの使用法の詳細については、『*Cisco Identity Services Engine Admin Guide, Release 1.3*』を参照してください。

対象読者

この API リファレンス ガイドは、ネットワーク環境内で Cisco ISE アプライアンスを管理する経験豊富なシステム管理者、API を利用するシステム インテグレータ、Cisco ISE 導入の管理やトラブルシューティングの役割を持つサードパーティ製パートナーを対象としています。この API リファレンスガイドを使用する前提条件として、トラブルシューティングと診断方法についてと、API コールの作成および解釈方法について、基礎を理解しておく必要があります。

ガイドの構成

このマニュアルの構成は、次のとおりです。

- Part 1：Cisco ISE Monitoring REST API
  - 第 1 章「Monitoring REST API の概要」
  - 第 2 章「セッション管理クエリー API」
  - 第 3 章「トラブルシューティング用のクエリー API」
  - 第 4 章「認可変更 REST API」
- Part 2：Cisco ISE 外部 RESTful API サービス API
  - 第 5 章「ERS API の概要」
  - 第 7 章「外部 RESTful サービス API の操作」
- Part 3：Cisco pxGrid API
  - 付録 A「Cisco ISE 障害理由レポート」

ドキュメントの表記法

ここでは、このマニュアル全体で使用されている表記法について説明します。

注意 「*要注意*」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記載されています。

コメント 「*注釈*」です。役立つ情報や、このマニュアル以外の参照資料などを紹介しています。

この API リファレンス ガイドは次の表記法を使用して、指示と情報を伝送します。

項目	表記法
手順で選択する必要があるコマンド、キーワード、特殊な用語、およびオプション	**太字**
ユーザが値を指定する変数、および新しい用語や重要な用語	*イタリック体*
表示されるセッション情報、システム情報、パス、およびファイル名	screen フォント
ユーザが入力する情報	**太字の screen フォント**
ユーザが入力する変数	*イタリック体の screen フォント*
メニュー項目およびボタン名	**太字**
選択する順序で並べられたメニュー項目	**[オプション（Option）] > [ネットワーク設定（Network Preferences）]**

マニュアルの更新

表 1 に、このマニュアルの初版と、更新の履歴、および最新の更新が示されます。

表 1　Cisco Identity Services Engine API リファレンス ガイド リリース 1.3 の更新

日付	説明
8-25-2014	Cisco Identity Services Engine（ISE）Release 1.3

製品マニュアル

コメント　初版発行後、印刷物または電子マニュアルのアップデートを行う場合があります。マニュアルの更新については、http://cisco.com で確認してください。

表 2 に、www.cisco.com で入手可能な Cisco ISE Release 1.3 の関連製品のマニュアルを示します。www.cisco.com ですべての製品のエンド ユーザ マニュアルを検索するには、次のサイトにアクセスしてください。

http://www.cisco.com/go/techdocs

関連資料

ここでは、このリリースのマニュアルと、このプラットフォームのマニュアルの情報を提供します。

このリリースのマニュアル

表 2 に Cisco ISE Release で利用可能な製品マニュアルを示します。Cisco ISE の全般的な製品情報は http://www.cisco.com/go/ise で確認できます。エンドユーザ マニュアルは、Cisco.com の http://www.cisco.com/en/US/products/ps11640/tsd_products_support_series_home.html から入手できます。

表 2　Cisco Identity Services Engine の製品マニュアル

参照先	場所
『Release Notes for the Cisco Identity Services Engine, Release 1.3』	http://www.cisco.com/en/US/products/ps11640/prod_release_notes_list.html
『Cisco Identity Services Engine Network Component Compatibility, Release 1.3』	http://www.cisco.com/en/US/products/ps11640/products_device_support_tables_list.html
『Cisco Identity Services Engine Admin Guide, Release 1.3』	http://www.cisco.com/en/US/products/ps11640/products_user_guide_list.html
『Cisco Identity Services Engine Hardware Installation Guide, Release 1.3』	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html

表 2　Cisco Identity Services Engine の製品マニュアル （続き）

参照先	場所
『*Cisco Identity Services Engine Migration Guide for Cisco Secure ACS 5.5 to Release 1.3*』	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
『*Cisco Identity Services Engine Sponsor Portal User Guide, Release 1.3*』	http://www.cisco.com/en/US/products/ps11640/products_user_guide_list.html
『*Cisco Identity Services Engine CLI Reference Guide, Release 1.3*』	http://www.cisco.com/en/US/products/ps11640/prod_command_reference_list.html
『*Cisco Identity Services Engine API リファレンス ガイド リリース 1.3*』	http://www.cisco.com/en/US/products/ps11640/prod_command_reference_list.html
『*Cisco Identity Services Engine Troubleshooting Guide, Release 1.3*』	http://www.cisco.com/en/US/products/ps11640/prod_troubleshooting_guides_list.html
『*Regulatory Compliance and Safety Information for Cisco Identity Services Engine, Cisco 1121 Secure Access Control System, Cisco NAC Appliance, Cisco NAC Guest Server, and Cisco NAC Profiler*』	http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
『*Cisco Identity Services Engine In-Box Documentation and China RoHS Pointer Card*』	http://www.cisco.com/en/US/products/ps11640/products_documentation_roadmaps_list.html

プラットフォーム別のマニュアル

ポリシー管理ビジネス ユニットのマニュアルは http://www.cisco.com の次の場所で入手できます。

- Cisco ISE
  http://www.cisco.com/en/US/products/ps11640/prod_installation_guides_list.html
- Cisco Secure ACS
  http://www.cisco.com/en/US/products/ps9911/tsd_products_support_series_home.html
- Cisco NAC Appliance
  http://www.cisco.com/en/US/products/ps6128/tsd_products_support_series_home.html
- Cisco NAC Profiler
  http://www.cisco.com/en/US/products/ps8464/tsd_products_support_series_home.html
- Cisco NAC ゲスト サーバ
  http://www.cisco.com/en/US/products/ps10160/tsd_products_support_series_home.html

マニュアルの入手方法およびテクニカル サポート

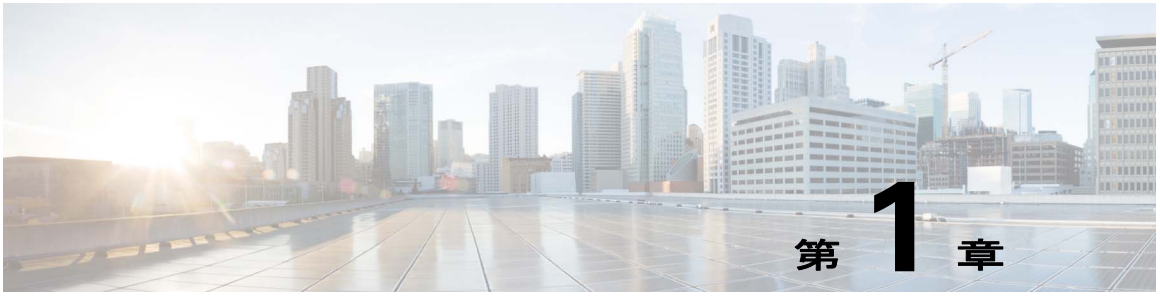
マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

PART 1

Cisco ISE Monitoring REST API



Monitoring REST API の概要

Monitoring REST API では、ネットワークでモニタリング ノードを使用して、セッションおよびノード固有の情報を収集することができます。セッションは、目的のノードにアクセスしてから情報の収集に必要な操作を完了するまでの期間として定義されます。

次の Monitoring REST API カテゴリが Cisco ISE Release 1.3 でサポートされます。

- セッション管理
- トラブルシューティング
- 認可変更（CoA）

（注） Monitoring ペルソナによって監視されているエンドポイントに関する情報を収集するために、サポート対象のカテゴリだけを使用できます。Monitoring は、ノード タイプが Cisco ISE Release 1.3 の導入で実行できる、サポート対象の 3 つのペルソナの 1 つです。このガイドの残りの部分では、Cisco ISE ノードの Monitoring ペルソナを説明するため、「モニタリング ノード」を使用します。

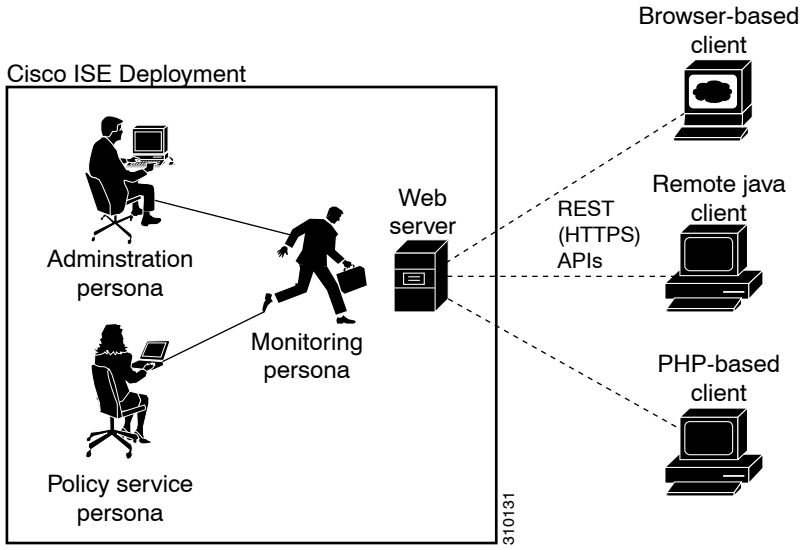
これらのカテゴリを Cisco ISE アプライアンスの Policy service ペルソナに関する情報の収集に使用しようとすると、エラーが発生します。Cisco ISE ノードおよびペルソナに関する詳細については、『*Cisco Identity Services Engine Admin Guide, Release 1.3*』を参照してください。

Monitoring REST API コールを使用すると、ネットワークで、個々のエンドポイントに格納されている重要なリアルタイムのセッション ベースの情報を検索、監視、収集することができます。モニタリング ノードを通じてこの情報にアクセスできます。

収集するリアルタイムのセッション ベースの情報は、Cisco ISE 操作を理解するのに役立ち、状態や問題の診断を支援することができます。また、モニタリング動作に影響を及ぼす可能性のあるエラー条件、またはアクティビティや動作をトラブルシュートするために使用できます。図 1-1 に示すように、Monitoring REST API コールは、モニタリング ノードにアクセスして Cisco ISE 導入のエンドポイントに格納されている重要なセッション ベースの情報を取得する目的で使用されます。

（注） Monitoring REST API for Cisco ISE リリース 1.2 は廃止され、すべてのサービス コールの URL パスが変更されました。Monitoring REST API for Cisco ISE リリース 1.3 を使用してください。

図 1-1　分散展開での Monitoring REST API コール



モニタリング ノードの確認

はじめる前に

API コールをモニタリング ノードで正常に呼び出す前に、監視するノードが有効なノードであることを確認しておく必要があります。

（注） パブリック Monitoring REST API を使用できるようにするには、最初に有効なクレデンシャルを使用して Cisco ISE で認証を受ける必要があります。

ステップ 1　有効なログインクレデンシャル（ユーザ名とパスワード）を [Cisco ISE ログイン（Cisco ISE Login）] ウィンドウに入力し、**[ログイン（Login）]** をクリックします。

Cisco ISE ダッシュボードとユーザ インターフェイスが表示されます。

ステップ 2　**[許可（Authorization）] > [システム（System）] > [展開（Deployment）]** の順に選択します。

展開されたすべての設定済みノードがリストされた [展開ノード（Deployment Nodes）] ページが表示されます。

ステップ 3　[展開ノード（Deployment Nodes）] ページの [ロール（Roles）] カラムで、モニタするターゲット ノードのロールがモニタリング ノードとしてリストされていることを確認します。

サポートされる API コール

次の表で、さまざまな種類の API コールを説明し、API コールの形式の例を示します。

- 表 1-1（P.1-3）：セッション管理用の API コールを定義します。
- 表 1-2（P.1-6）：トラブルシューティング用の API コールを定義します。
- 表 1-3（P.1-8）：CoA API コールを定義します。

Cisco ISE でサポートされる Monitoring REST API を使用して認証を受けるため、汎用プログラマチック インターフェイスを使用する計画の場合、Cisco ISE と使用するツールを接続する REST ベースのクライアントを最初に作成する必要があります。次に、この REST クライアントを使用して Cisco ISE Monitoring REST API で認証を受け、API 要求を変換してモニタリング ノードに送信します。そして、API 応答を再変換し、指定されたツールに引き渡します。

表 1-1　　Cisco ISE セッション管理 API コール

API コール カテゴリ	説明と例
セッション カウンタ	
ActiveCount	アクティブなセッションの数をリストします。 https://<ISEhost>/admin/API/mnt/Session/ActiveCount
PostureCount	ポスチャされたエンドポイントの数をリストします。 https://<ISEhost>/admin/API/mnt/Session/PostureCount **（注）** ポスチャとは、Cisco ISE ネットワークに接続しているすべてのエンドポイントの状態（またはポスチャ）の確認を支援するサービスです。Cisco ISE は、デバイスのポスチャ コンプライアンスを確認するために NAC Agent を使用します。
ProfilerCount	アクティブなプロファイラ サービス セッションの数をリストします。 https://<ISEhost>/admin/API/mnt/Session/ProfilerCount **（注）** プロファイラとは、Cisco ISE ネットワークにあるすべての接続エンドポイントの機能の識別、検索、確認を支援するサービスです。

表 1-1　Cisco ISE セッション管理 API コール（続き）

API コール カテゴリ	説明と例
セッション リスト **（注）** セッション リストには、MAC アドレス、ネットワーク アクセス デバイス（NAD）の IP アドレス、ユーザ名、セッションに関連付けられているセッション ID 情報が含まれます。	
ActiveList	すべてのアクティブなセッションをリストします。 https://<ISEhost>/admin/API/mnt/Session/ActiveList **（注）** Cisco ISE のこのリリースでは、アクティブな認証済みエンドポイント セッションの表示可能な最大数は、250,000 に制限されています。
AuthList	現在アクティブなすべての認証済みセッションをリストします。 https://<ISEhost>/admin/API/mnt/Session/AuthList/<parameteroptions> 異なる値を返す次のパラメータ オプションを指定できます。 • null/null：すべてのアクティブな認証済みセッションをリストします。 • null/endtime：指定された終了時刻の後にアクティブなすべての認証済みセッションがリストされます。 • starttime/null：指定された開始時刻の前にアクティブなすべての認証済みセッションがリストされます。 • starttime/endtime：指定された開始時刻と終了時刻の間で認証されたすべてのアクティブなセッションがリストされます。 次の形式で、開始時刻と終了時刻の日付と時刻を入力します。 YYYY-MM-DD hh:mm:ss.s 引数の説明 • YYYY：4 桁の年 • MM：2 桁の月（01 = 1 月など） • DD：2 桁の日（01 ～ 31） • hh：2 桁の時刻（00 ～ 23）（a.m.と p.m. は 使用できません） • mm：2 桁の分（00 ～ 59） • ss：2 桁の秒（00 ～ 59） • s：秒の小数を表す 1 桁以上の値 **（注）** すべての Cisco ISE ノードは、タイム ゾーンを使用して設定されます。推奨されるタイム ゾーンは UTC です。 4 つのパラメータ オプションをすべて示すサンプルについては、「null/null オプションを使用した AuthList API コールから返されるサンプル データ」（P.2-9）を参照してください。

表 1-1　Cisco ISE セッション管理 API コール（続き）

API コール カテゴリ	説明と例
セッション属性 **（注）** これは、指定された検索属性を含む最新のセッションのタイムスタンプに基づいた検索です。	
MACAddress	指定した MAC アドレスを含む最新のセッションについてデータベースを検索します。 https://<ISEhost>/admin/API/mnt/Session/MACAddress/<macaddress> **（注）** XX:XX:XX:XX:XX:XX は MAC アドレス形式です。大文字と小文字は区別されません（例：0a: 0B: 0c: 0D: 0e: 0F）。 **（注）** MAC アドレスは、監視対象の正しいセッションを検索する唯一の一意のキーとして機能します。MAC アドレスの検索のベースとすることが可能なアクティブなすべてのセッションと MAC アドレスをリストするには ActiveList API コールを使用します。
UserName	指定したユーザ名を含む最新のセッションについてデータベースを検索します。 https://<ISEhost>/admin/API/mnt/Session/UserName/<username> **（注）** ユーザ名は、ネットワーク ユーザ名に使用しているのと同じ Cisco ISE パスワード ポリシーに準拠している必要があります。Monitoring REST API の唯一の無効な文字はバックスラッシュ（\）文字です。詳細については、『*Cisco Identity Services Engine User Guide, Release 1.1*』の「User Password Policy」を参照してください。
IPAddress	指定した NAS IP アドレスを含む最新のセッションについてデータベースを検索します。 https://<ISEhost>/admin/API/mnt/Session/IPAddress/<nasipaddress> **（注）** xxx.xxx.xxx.xxx は NAS IP アドレス形式（例：10.10.10.10）です。

セッション管理用の Cisco ISE API コールの詳細については、第 2 章「セッション管理クエリー API」を参照してください。

表 1-2　Cisco ISE トラブルシューティング API コール - トラブルシューティング

API コール	説明と例
Version	ノードのバージョンおよびタイプをリストします。 https://<ISEhost>/admin/API/mnt/Version ノードのタイプは、次の値（0 ～ 3）のいずれかです。 0：STAND_ALONE_MNT_NODE 1：ACTIVE_MNT_NODE 2：STAND_BY_MNT_NODE 3：NOT_AN_MNT_NODE **（注）** STAND_ALONE_MNT_NODE は、分散展開で機能しないモニタリング ノードであることを意味します。 ACTIVE_MNT_NODE は、分散展開におけるプライマリ - セカンダリ関係のプライマリ ノードであることを意味します。 STAND_BY_MNT_NODE は、分散展開におけるプライマリ - セカンダリ ペアのセカンダリ ノードであることを意味します。 NOT_AN_MNT_NODE は、モニタリング ノードではないことを意味します。サポート対象の ISE ノードおよびペルソナの詳細については、『*Cisco Identity Services Engine User Guide, Release 1.1*』を参照してください。
FailureReasons	障害の理由をリストします。 https://<ISEhost>/admin/API/mnt/FailureReasons 各障害理由は、次の例に示すように、エラーコード（failureReason id）、簡単な説明（code）、障害理由（cause）、および可能な対処（resolution）を表示します。 <failureReason id="100009"> <code> 100009 WEBAUTH_FAIL <cause> This may or may not be indicating a violation. <resolution> Please review and resolve this issue according to your organization's policy. **（注）** FailureReasons API コールは、モニタリング ノードから情報を収集するために一度だけ呼び出されます。使用しているファイル システムまたはデータベースに、返された障害理由の内容を保存する必要があります。これらの API コールの返信内容はあくまでも参照用に使用することを目的としています。認証中に問題が発生した場合、認証応答で提供される障害理由コードと、ユーザのファイル システムまたはデータベースに保存した障害理由のリストを比較する必要があります。 Cisco ISE 障害理由の完全なリストについては、付録 A「Cisco ISE 障害理由レポート」を参照してください。

表 1-2　Cisco ISE トラブルシューティング API コール - トラブルシューティング（続き）

API コール	説明と例
AuthStatus	すべてのセッションの認証ステータスをリストします。 https://<ISEhost>/admin/API/mnt/AuthStatus/MACAddress/<macaddress>/<numberofseconds>/<numberofrecordspermacaddress>/All **（注）** seconds パラメータ <numberofseconds> は、0 秒から 432000 秒（5 日）の範囲でユーザが設定できます。
セッション アカウンティング ステータスの取得	
AcctStatus	特定の期間内のすべてのセッションのアカウンティング ステータスを示します。 https://<ISEhost>/admin/API/mnt/Session/AcctStatusTT/MACAddress/<macaddress>/<numberof seconds> **（注）** seconds パラメータ <numberofseconds> は、0 秒から 432000 秒（5 日）の範囲でユーザが設定できます。

トラブルシューティング用の Cisco ISE API コールの詳細については、第 2 章「セッション管理クエリー API」を参照してください。

表 1-3　Cisco ISE 認可変更 API コール

API コール	説明と例
Reauth	セッション再認証コマンドとタイプを送信します。 https://<ISEhost>/admin/API/mnt/CoA/Reauth/<serverhostname>/<macaddress>/<reauthtype>/<nasipaddress>/<destinationipaddress> ここで、*<ISEhost>* は ISE ホストの IP アドレスを示し、*<serverhostname>* は ISE サーバの名前を示し、*<nasipaddress>* は NAS の識別 IP アドレスを示し、*<destinationipaddress>* は宛先の IP アドレスを示します。 再認証タイプは次の値（0 ～ 2）のいずれかです。 0：REAUTH_TYPE_DEFAULT 1：REAUTH_TYPE_LAST 2：REAUTH_TYPE_RERUN **（注）** NAS IP アドレスが不明な場合は、この時点までに必要な値を入力できます。API はこれらの値を検索クエリーに使用します。ただし、この API コールを実行するには、MAC アドレスを知っている必要がありますが、NAS IP アドレスで始まる他のパラメータはヌルにしたままにできます。NAS IP アドレスを指定する場合、宛先 IP アドレスも指定する必要があります。 この API コールは、CoA をリモートで実行する要求を送信する Monitoring ISE ノードでしか実行できません。Administration ISE ノードは、これらの CoA API コールの実行には関係ないか、必要がありません。
セッション切断	
切断	セッション切断コマンドおよびポート オプション タイプを送信します。 https://<ISEhost>/admin/API/mnt/CoA/Disconnect/<serverhostname>/<macaddress>/<disconnecttype>/<nasipaddress>/<destinationipaddress> ポート オプション タイプは次の値（0 ～ 2）のいずれかです。 0：DYNAMIC_AUTHZ_PORT_DEFAULT 1：DYNAMIC_AUTHZ_PORT_BOUNCE 2：DYNAMIC_AUTHZ_PORT_SHUTDOWN **（注）** NAS IP アドレスが不明な場合は、この時点までに必要な値を入力します。API はこれらの値を検索クエリーに使用します。ただし、この API コールを実行するには、MAC アドレスを知っている必要がありますが、他のパラメータはヌルにしたままにできます。

Cisco ISE 認可変更 API コールに関する詳細については、第 4 章「認可変更 REST API」を参照してください。

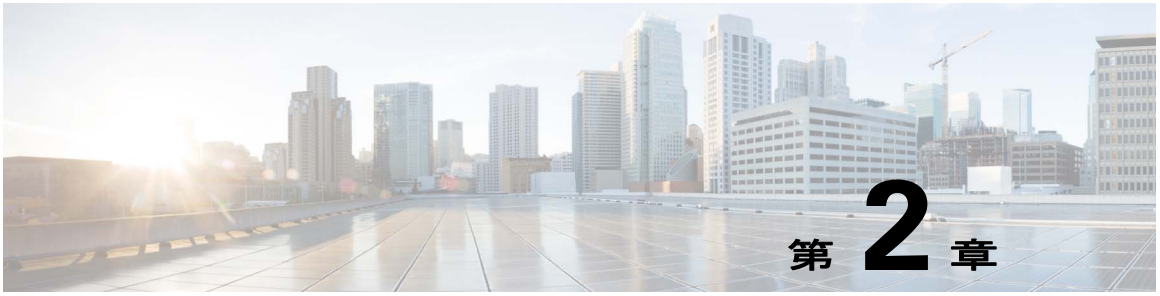
HTTP PUT API コール

表 1-2 の AuthStatus API コールと同様に、クライアントがアカウント ステータスを取得できるようにする API コールの HTTP PUT バージョンがあります。Monitoring REST API は、HTTP GET コールについて記述したこのマニュアルの例で示すように、HTTP PUT と HTTP GET の両方のコールをサポートします。HTTP PUT は、パラメータの入力が必要なコールの必要性に対処します。次のスキーマ ファイルの例は、アカウント ステータスの要求です。

```
<?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="acctRequest" type="mnTRESTAcctRequest"/>

  <xs:complexType name="mnTRESTAcctRequest">
    <xs:complexContent>
      <xs:extension base="mnTRESTRequest">
        <xs:sequence>
          <xs:element name="duration" type="xs:string" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="mnTRESTRequest" abstract="true">
    <xs:sequence>
      <xs:element name="valueList">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="value" type="xs:string" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="searchCriteria" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```



セッション管理クエリー API

この章では、Cisco ISE 展開において、Cisco Monitoring ISE ノード内から重要なセッション関連の情報を取得する手段を提供するセッション管理 API コールについて説明します。

セッション カウンタ API コール

次のセッション カウンタ API コールによって、Cisco ISE 展開におけるターゲット Cisco Monitoring ISE ノードのセッション関連情報の現在のカウントをすぐに収集できるようになります。

- アクティブ セッション（ActiveCount）：アクティブ セッションは、ネットワークで認証されるセッションの 1 つです。
- ポスチャ セッション（PostureCount）：ポスチャが結論付けられる（準拠/非準拠）と、ポスチャ状態がアサートされます。ポスチャはオプションで、IP 電話やプリンタなどはポスチャ状態になりません。ポスチャ後、アカウンティングの開始が設定されると開始済み状態になるため、ポスチャ状態は短期間の一時的な状態です。
- プロファイル セッション（ProfilerCount）

いずれかのフェーズでエンドポイントが停止した場合、これらのさまざまな状態はトラブルシューティングが必要であることを示します。

アクティブ セッション カウンタ

現在アクティブなすべてのセッション カウントを取得するために ActiveCount 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、ActiveCount API コールを呼び出すことにより、すべてのアクティブ セッションをカウントする手順、この API コール発行後に返されるアクティブ セッション データのサンプルについて説明します。

ActiveCount API の出力スキーマ

このサンプル スキーマ ファイルは、ISE のノードのターゲット Monitoring ペルソナでアクティブ セッションのカウントを取得するための ActiveCount 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="sessionCount" type="activeCount"/>
  <xs:complexType name="activeCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

ActiveCount 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して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。

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 ActiveCount API コールを入力します。

```
https://acme123/admin/API/mnt/Session/ActiveCount
```

（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、ターゲット 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」（P.1-2）

ActiveCount 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 ActiveCount API コールを呼び出すときに返されるデータ（アクティブ セッション数）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.
-
<sessionCount>
<count>5</count>
</sessionCount>
```

ポスチャ セッション カウンタ

現在アクティブなすべてのポスチャ セッションの現在のカウントを取得するために PostureCount API コールを使用できます。

PostureCount 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 ノードで現在アクティブなポスチャ セッションのカウントを取得するための PostureCount 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="sessionCount" type="postureCount"/>

  <xs:complexType name="postureCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

PostureCount 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して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。

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/Session/<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 PostureCount API コールを入力します。

```
https://acme123/admin/API/mnt/Session/PostureCount
```

（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、ターゲット 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」（P.1-2）

PostureCount 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 PostureCount API コールを呼び出すときに返されるデータ（現在アクティブなポスチャ セッション数）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.

-
<sessionCount>
<count>3</count>
</sessionCount>
```

プロファイラ セッション カウンタ

現在アクティブなすべてのプロファイラ セッション カウントを取得するために ProfilerCount API コールを使用できます。

ProfilerCount 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 ノードで現在アクティブなプロファイラ セッションのカウントを取得するための ProfilerCount 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="sessionCount" type="profilerCount"/>

  <xs:complexType name="profilerCount">
    <xs:sequence>
      <xs:element name="count" type="xs:int"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

ProfilerCount 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して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。

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/Session/<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 ProfilerCount API コールを入力します。

```
https://acme123/admin/API/mnt/Session/ProfilerCount
```

（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」（P.1-2）

ProfilerCount 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 ProfilerCount API コールを呼び出すときに返されるデータ（現在アクティブなプロファイラ セッション数）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.

-
<sessionCount>
<count>1</count>
</sessionCount>
```

単純なセッション リスト API コール

次の単純なセッション リスト API コールによって、Cisco ISE 展開におけるターゲット Cisco Monitoring ISE ノードの現在のアクティブ セッションに関連付けられた MAC アドレス、ネットワーク アクセス デバイス（NAD）の IP アドレス、ユーザ名、セッション ID などのセッション関連の情報をすぐに収集できるようになります。

- アクティブなセッション リスト（ActiveList）
- 認証セッション リスト（AuthList）

アクティブなセッション リスト

現在アクティブなすべてのセッションをリストするには ActiveList API 呼び出しを使用できます。

（注）　アクティブな認証済みエンドポイント セッションの表示可能な最大数は、100,000 に制限されています。

ActiveList 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 ノードで現在アクティブなセッション（およびセッション関連情報）のリストを取得するための ActiveList 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="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

ActiveList 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して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。

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/Session/<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 ActiveList API コールを入力します。

```
https://acme123/admin/API/mnt/Session/ActiveList
```

（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 呼び出し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」（P.1-2）

ActiveList 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 ActiveList API コールを呼び出すときにアクティブ セッションのリストから返されるセッション関連データを示します。

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

-
<activeSessionList noOfActiveSession="5">
-
<activeSession>
<calling_station_id>00:0C:29:FA:EF:0A</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<calling_station_id>70:5A:B6:68:F7:CC</calling_station_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000032</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>0000002C</acct_session_id>
<audit_session_id>0ACB6BA10000002A165FD0C8</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>ipepvpnuser</user_name>
<calling_station_id>172.23.130.89</calling_station_id>
<nas_ip_address>10.203.107.45</nas_ip_address>
<acct_session_id>A2000070</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

認証セッション リスト

現在アクティブなすべての認証セッションのリストを取得するために AuthList API コールを使用できます。

（注）　アクティブな認証済みエンドポイント セッションの表示可能な最大数は、100,000 に制限されています。

AuthList 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、指定した期間内（または「null/null」パラメータを使用して期間を指定しない場合）現在アクティブなすべての認証セッションのリストを取得するための AuthList 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="activeSessionList" type="simpleActiveSessionList"/>

<xs:complexType name="simpleActiveSessionList">
    <xs:sequence>
      <xs:element name="activeSession" type="simpleActiveSession" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="noOfActiveSession" type="xs:int" use="required"/>
  </xs:complexType>

  <xs:complexType name="simpleActiveSession">
    <xs:sequence>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

AuthList 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して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。

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/Session/<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 AuthList API コールを入力します。

（注）　次の 2 種類の例では、定義済みの開始時刻パラメータおよび null パラメータを使用し、開始時刻以降に認証された現在アクティブなセッションのリストを表示します。2 番目の例は、現在アクティブなすべての認証済みセッションのリストを表示する「null/null」パラメータを使用します。この API コールに対する 4 種類のパラメータ設定の例については、「null/null オプションを使用した AuthList API コールから返されるサンプル データ」（P.2-9）を参照してください。

```
https://acme123/admin/API/mnt/Session/AuthList/2010-12-14 15:33:15/null

https://acme123/admin/API/mnt/Session/AuthList/null/null
```

（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」（P.1-2）

null/null オプションを使用した AuthList API コールから返されるサンプル データ

次に、null/null オプションを使用して AuthList API コールを呼び出した場合に返される現在アクティブな認証済みセッションのリストの例を示します。

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

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c000000174D07F487</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
```

```
</activeSessionList>
```

endtime/null オプションを使用した AuthList API コールから返されるサンプル データ

次に、endtime/null オプションを使用して AuthList API コールを呼び出した場合に返される現在アクティブな認証済みセッションのリストの例を示します。

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

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

null/starttime オプションを使用した AuthList API コールから返されるサンプル データ

次に、null/starttime オプションを使用して AuthList API コールを呼び出した場合に返される現在アクティブな認証済みセッションのリストの例を示します。

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

-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>bob_ludlum</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
```

```
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>tom_wolfe</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

starttime/endtime オプションを使用した AuthList API コールから返されるサンプル データ

次に、starttime/endtime オプションを使用して AuthList API コールを呼び出した場合に返される現在アクティブな認証済みセッションのリストの例を示します。

```
This XML file does not appear to have any style information associated with it.The
document tree is shown below.
-
<activeSessionList noOfActiveSession="3">
-
<activeSession>
<user_name>ipepwlcuser</user_name>
<calling_station_id>00:26:82:7B:D2:51</calling_station_id>
<nas_ip_address>10.203.107.10</nas_ip_address>
<audit_session_id>0acb6b0c0000001F4D08085A</audit_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>graham_hancock</user_name>
<calling_station_id>00:50:56:8E:28:BD</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000035</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
-
<activeSession>
<user_name>hunter_thompson</user_name>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_ip_address>10.203.107.161</nas_ip_address>
<acct_session_id>00000033</acct_session_id>
<server>HAREESH-R6-1-PDP2</server>
</activeSession>
</activeSessionList>
```

詳細なセッション属性 API コール

次の詳細なセッション属性 API コールによって、次のようなキー情報の最新のセッションをすぐに検索することができるようになります。

- MAC アドレス セッションの検索（MACAddress）
- ユーザ名のセッションの検索（UserName）
- NAS IP アドレス セッションの検索（ターゲット Monitoring ISE ノードに関連付けられた IP アドレス）

MAC アドレス セッションの検索

現在のアクティブなセッションから指定された MAC アドレスを取得するために MACAddress API コールを使用できます。この API コールは、ノード データベース テーブルから供給されるさまざまセッション関連の情報をリストします。

MACAddress API の出力スキーマ

このサンプル スキーマ ファイルは、現在アクティブなセッションから指定された MAC アドレスを取得するための MACAddress 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="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <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="authen_protocol" 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="auth_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="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" 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="service_type" 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="selected_azn_profiles" 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="eap_tunnel" 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:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
        <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
        <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
        <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
        <xs:element name="security_group" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
        <xs:element name="started" type="xs:anyType" minOccurs="0"/>
        <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
        <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
        <xs:element name="type" type="xs:long" minOccurs="0"/>
        <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="vlan" type="xs:string" minOccurs="0"/>
        <xs:element name="dacl" type="xs:string" minOccurs="0"/>
        <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
        <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
        <xs:element name="reason" type="xs:string" minOccurs="0"/>
        <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
```

MACAddress 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して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。

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>）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 MACAddress API コールを入力します。

```
https://acme123/admin/API/mnt/Session/MACAddress/0A:0B:0C:0D:0E:0F
```

（注）　XX:XX:XX:XX:XX:XX 形式を使用して MAC アドレスを指定していることを確認します。

（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」（P.1-2）

MACAddress API コールから返されるサンプル データ

次に、ActiveList API コールを呼び出すときにアクティブ セッションのリストから返されるセッション関連データの例を示します。

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

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>hunter_thompson</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
-
<network_device_groups>
```

```
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=00-14-BF-5A-0C-03; User-Name=00-14-BF-5A-0C-03;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>00:14:BF:5A:0C:03</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
&lt;nas_port_type&gt;Ethernet&lt;/nas_port_type&gt;
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
```

```
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

ユーザ名のセッションの検索

現在のアクティブなセッションから指定されたユーザ名を取得するために UserName API コールを使用できます。この API は、ノード データベース テーブルから供給されるさまざまセッション関連の情報をリストします。

UserName API の出力スキーマ

このサンプル スキーマ ファイルは、現在アクティブなセッションから指定されたユーザ名を取得するための UserName 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="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <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="authen_protocol" 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="auth_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="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" 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="service_type" 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="selected_azn_profiles" 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="eap_tunnel" 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:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
        <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
        <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
        <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
        <xs:element name="security_group" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
        <xs:element name="started" type="xs:anyType" minOccurs="0"/>
        <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
        <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
        <xs:element name="type" type="xs:long" minOccurs="0"/>
        <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="vlan" type="xs:string" minOccurs="0"/>
        <xs:element name="dacl" type="xs:string" minOccurs="0"/>
        <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
        <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
        <xs:element name="reason" type="xs:string" minOccurs="0"/>
        <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
```

UserName 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して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。

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>/<username>）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 UserName API コールを入力します。

```
https://acme123/admin/API/mnt/Session/UserName/graham_hancock
```

（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」（P.1-2）

UserName API コールから返されるサンプル データ

次に、UserName API コールを呼び出すときにアクティブ セッションのリストから返されるセッション関連データの例を示します。

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

-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>graham_hancock</user_name>
<nas_ip_address>10.203.107.161</nas_ip_address>
<calling_station_id>00:14:BF:5A:0C:03</calling_station_id>
<nas_port>50115</nas_port>
<identity_group>Profiled</identity_group>
<network_device_name>Core-Switch</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>Lookup</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T02:11:12.359Z</auth_acs_timestamp>
```

```
<authentication_method>mab</authentication_method>
-
<execution_steps>
11001,11017,11027,15008,15048,15004,15041,15004,15013,24209,24211,22037,15036,15048,15048,
15004,15016,11022,11002
</execution_steps>
<audit_session_id>0ACB6BA1000000351BBFBF8B</audit_session_id>
<nas_port_id>GigabitEthernet1/0/15</nas_port_id>
<nac_policy_compliance>Pending</nac_policy_compliance>
<auth_id>1291240762077361</auth_id>
<auth_acsview_timestamp>2010-12-15T02:11:12.360Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/681</acs_session_id>
<service_selection_policy>MAB</service_selection_policy>
<identity_store>Internal Hosts</identity_store>
-
<response>
{UserName=graham_hancock; User-Name=graham_hancock;
State=ReauthSession:0ACB6BA1000000351BBFBF8B;
Class=CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681;
Termination-Action=RADIUS-Request; cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://HAREESH-R6-1-PDP2.cisco.com:8443/guestportal/gateway?se
ssionId=0ACB6BA1000000351BBFBF8B&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-ACL-DENY-4ced8390; }
</response>
<service_type>Call Check</service_type>
<use_case>Host Lookup</use_case>
<cisco_av_pair>audit-session-id=0ACB6BA1000000351BBFBF8B</cisco_av_pair>
<acs_username>graham_hancock</acs_username>
<radius_username>00:14:BF:5A:0C:03</radius_username>
<selected_identity_store>Internal Hosts</selected_identity_store>
<authentication_identity_store>Internal Hosts</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
&lt;nas_port_type&gt;Ethernet&lt;/nas_port_type&gt;
<selected_azn_profiles>CWA</selected_azn_profiles>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-MTU=1500,EAP-Key-Name=,CPMSessionID=0ACB6BA1000000351BBFBF8B,CPMSessionID=0ACB6BA
1000000351BBFBF8B,EndPointMACAddress=00-14-BF-5A-0C-03,HostIdentityGroup=Endpoint Identity
Groups:Profiled,Device Type=Device Type#All Device Types,Location=Location#All
Locations,Model Name=Unknown,Software Version=Unknown,Device IP
Address=10.203.107.161,Called-Station-ID=04:FE:7F:7F:C0:8F
</other_attributes>
<response_time>77</response_time>
<acct_id>1291240762077386</acct_id>
<acct_acs_timestamp>2010-12-15T02:12:30.779Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T02:12:30.780Z</acct_acsview_timestamp>
<acct_session_id>00000038</acct_session_id>
<acct_status_type>Interim-Update</acct_status_type>
<acct_session_time>78</acct_session_time>
<acct_input_octets>13742</acct_input_octets>
<acct_output_octets>6277</acct_output_octets>
<acct_input_packets>108</acct_input_packets>
<acct_output_packets>66</acct_output_packets>
-
<acct_class>
CACS:0ACB6BA1000000351BBFBF8B:HAREESH-R6-1-PDP2/81148292/681
</acct_class>
<acct_delay_time>0</acct_delay_time>
<started xsi:type="xs:boolean">false</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

NAS IP アドレス セッションの検索

現在のセッションから指定された NAS IP アドレスのデータを取得するために IPAddress API コールを使用できます。この API は、ノード データベース テーブルから供給されるさまざまセッション関連の情報をリストします。

IPAddress API の出力スキーマ

このサンプル スキーマ ファイルは、現在アクティブなセッションから指定された NAS IP アドレスを取得するための IPAddress 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="sessionParameters" type="restsdStatus"/>

  <xs:complexType name="restsdStatus">
    <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="authen_protocol" 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="auth_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="auth_id" type="xs:long" minOccurs="0"/>
      <xs:element name="auth_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" 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="service_type" 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="selected_azn_profiles" 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="eap_tunnel" 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:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_id" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_acs_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="acct_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_status_type" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_session_time" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_input_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_output_octets" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_input_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_output_packets" type="xs:long" minOccurs="0"/>
        <xs:element name="acct_class" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_terminate_cause" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_multi_session_id" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_authentic" type="xs:string" minOccurs="0"/>
        <xs:element name="termination_action" type="xs:string" minOccurs="0"/>
        <xs:element name="session_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="idle_timeout" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_interim_interval" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_delay_time" type="xs:string" minOccurs="0"/>
        <xs:element name="event_timestamp" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_connection" type="xs:string" minOccurs="0"/>
        <xs:element name="acct_tunnel_packet_lost" type="xs:string" minOccurs="0"/>
        <xs:element name="security_group" type="xs:string" minOccurs="0"/>
        <xs:element name="cisco_h323_setup_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_connect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="cisco_h323_disconnect_time" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="framed_protocol" type="xs:string" minOccurs="0"/>
        <xs:element name="started" type="xs:anyType" minOccurs="0"/>
        <xs:element name="stopped" type="xs:anyType" minOccurs="0"/>
        <xs:element name="ckpt_id" type="xs:long" minOccurs="0"/>
        <xs:element name="type" type="xs:long" minOccurs="0"/>
        <xs:element name="nad_acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
        <xs:element name="vlan" type="xs:string" minOccurs="0"/>
        <xs:element name="dacl" type="xs:string" minOccurs="0"/>
        <xs:element name="authentication_type" type="xs:string" minOccurs="0"/>
        <xs:element name="interface_name" type="xs:string" minOccurs="0"/>
        <xs:element name="reason" type="xs:string" minOccurs="0"/>
        <xs:element name="endpoint_policy" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:schema>
```

NAS IPAddress 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して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。

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>/<nasipaddress>）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 IPAddress API コールを入力します。

```
https://acme123/admin/API/mnt/Session/IPAddress/10.10.10.10
```

（注）　xxx.xxx.xxx.xxx の形式を使用して NAS IP アドレスを指定していることを確認します。

（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」（P.1-2）

IPAddress API コールから返されるサンプル データ

次に、IPAddress API コールを呼び出すときにアクティブ セッションのリストから返されるセッション関連データの例を示します。

```
This XML file does not appear to have any style information associated with it.The
document tree is shown below.
-
<sessionParameters>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>ipepvpnuser</user_name>
<nas_ip_address>10.10.10.10</nas_ip_address>
<calling_station_id>172.23.130.90</calling_station_id>
<nas_port>1015</nas_port>
<identity_group>iPEP-VPN-Group</identity_group>
<network_device_name>iPEP-HA-Routed</network_device_name>
<acs_server>HAREESH-R6-1-PDP2</acs_server>
<authen_protocol>PAP_ASCII</authen_protocol>
-
<network_device_groups>
Device Type#All Device Types,Location#All Locations
```

```
</network_device_groups>
<access_service>RADIUS</access_service>
<auth_acs_timestamp>2010-12-15T19:57:29.885Z</auth_acs_timestamp>
<authentication_method>PAP_ASCII</authentication_method>
-
<execution_steps>
11001,11017,15008,15048,15048,15004,15041,15004,15013,24210,24212,22037,15036,15048,15048,
15004,15016,11002
</execution_steps>
<audit_session_id>0acb6be4000000044D091DA9</audit_session_id>
<nac_policy_compliance>NotApplicable</nac_policy_compliance>
<auth_id>1291240762083580</auth_id>
<auth_acsview_timestamp>2010-12-15T19:57:29.887Z</auth_acsview_timestamp>
<message_code>5200</message_code>
<acs_session_id>HAREESH-R6-1-PDP2/81148292/693</acs_session_id>
<service_selection_policy>iPEP-VPN</service_selection_policy>
<identity_store>Internal Users</identity_store>
-
<response>
{User-Name=ipepvpnuser; State=ReauthSession:0acb6be4000000044D091DA9;
Class=CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693;
Termination-Action=RADIUS-Request; }
</response>
<service_type>Framed</service_type>
-
<cisco_av_pair>
audit-session-id=0acb6be4000000044D091DA9,ipep-proxy=true
</cisco_av_pair>
<acs_username>ipepvpnuser</acs_username>
<radius_username>ipepvpnuser</radius_username>
<selected_identity_store>Internal Users</selected_identity_store>
<authentication_identity_store>Internal Users</authentication_identity_store>
<identity_policy_matched_rule>Default</identity_policy_matched_rule>
<nas_port_type>Virtual</nas_port_type>
<selected_azn_profiles>iPEP-Unknown-Auth-Profile</selected_azn_profiles>
<tunnel_details>Tunnel-Client-Endpoint=(tag=0) 172.23.130.90</tunnel_details>
-
<other_attributes>
ConfigVersionId=44,DestinationIPAddress=10.203.107.162,DestinationPort=1812,Protocol=Radiu
s,Framed-Protocol=PPP,Proxy-State=Cisco Secure
ACS9e733142-070a-11e0-c000-000000000000-2906094480-3222,CPMSessionID=0acb6be4000000044D091
DA9,CPMSessionID=0acb6be4000000044D091DA9,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Model Name=Unknown,Software Version=Unknown,Device
IP Address=10.203.107.228,Called-Station-ID=172.23.130.94
</other_attributes>
<response_time>20</response_time>
<acct_id>1291240762083582</acct_id>
<acct_acs_timestamp>2010-12-15T19:57:30.281Z</acct_acs_timestamp>
<acct_acsview_timestamp>2010-12-15T19:57:30.283Z</acct_acsview_timestamp>
<acct_session_id>F1800007</acct_session_id>
<acct_status_type>Start</acct_status_type>
-
<acct_class>
CACS:0acb6be4000000044D091DA9:HAREESH-R6-1-PDP2/81148292/693
</acct_class>
<acct_delay_time>0</acct_delay_time>
<framed_protocol>PPP</framed_protocol>
<started xsi:type="xs:boolean">true</started>
<stopped xsi:type="xs:boolean">false</stopped>
</sessionParameters>
```

古いセッション

一部のデバイスでは、Wireless LAN Controller（WLC）など、古いセッションを保持できるようにする場合があります。このような場合、手動で非アクティブなセッションを削除するには、HTTP **DELETE** API コールを使用できます。これを行うには、URL（HTTP、HTTPS）構文のデータを転送するための無償のサードパーティ製のコマンドライン ツールである **cURL** を使用します。

ISE は、これらのセッションを追跡しません。これは、ISE が長期間ネットワークに接続できなくなり、WLC/NAD から多数のアカウンティングを停止できなくなった場合に問題を軽減するためです。この API を使用して ISE からこのような古い情報をクリアすることができます。

（注） HTTP および HTTPS を使用してファイルを取得するための無償ユーティリティである GNU Wget は、HTTP **DELETE** API コールをサポートしません。

古いセッションの削除

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して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。

（注） API コールは大文字と小文字が区別され、慎重に入力する必要があります。変数 <mntnode> は Cisco Monitoring ISE ノードを表します。

ステップ 4　手動で MAC アドレスの古いセッションを削除するには、コマンドラインで次の API コールを発行します。

```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/MACAddress/<madaddress>
```

ステップ 5　手動でセッション ID の古いセッションを削除するには、コマンドラインで次の API コールを発行します。

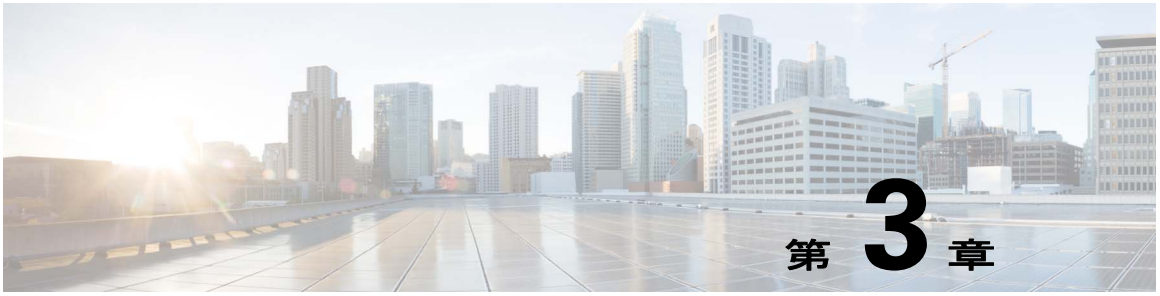
```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/SessionID/<sid#>
```

ステップ 6　手動でモニタリング ノードのすべてのセッションを削除するには、コマンドラインで次の API コールを発行します。

```
curl -X DELETE https://<mntnode>/admin/API/mnt/Session/Delete/All
```

関連項目

- 「モニタリング ノードの確認」（P.1-2）



第 3 章

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

バージョン 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して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（Login）] ページの **[ログイン時の問題（Problem logging in?）]** リンクをクリックして、手順 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」（P.1-2）

バージョン 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 ノードでバージョン 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 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」（P.A-1）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>
解決策	<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 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して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（Login）] ページの **[ログイン時の問題（Problem logging in?）]** リンクをクリックして、手順 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」（P.1-2）

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」（P.1-2）
- 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」（P.3-12）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
name=”passed” または name=”failed”	認証ステータスの結果： • パス • 失敗
name=”user_name”	ユーザ名
name=”nas_ip_address”	ネットワーク アクセス デバイスの IP アドレス/ホスト名
name=”failure_reason”	セッション認証に失敗した原因
name=”calling_station_id”	ソース IP アドレス
name=”nas_port”	ネットワーク アクセス サーバ ポート
name=”identity_group”	関連するユーザとホストで構成される論理グループ
name=”network_device_name”	ネットワーク デバイスの名前
name=”acs_server”	Cisco ISE アプライアンスの名前
name=”eap_authentication”	認証要求に使用する拡張認証プロトコル（EAP）方式
name=”framed_ip_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

表 3-2　　認証ステータス テーブルの属性（続き）

属性	説明
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”	フレームド 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: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して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（Login）] ページの **[ログイン時の問題（Problem logging in?）]** リンクをクリックして、手順 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」（P.1-2）

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>
<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=Defau
ltGuestPortal,
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>
<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>11001,11017,11027,15049,15008,15048,15048,15004,15041,15006,15013,24209,2
421
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 address	クライアントの MAC アドレス
audit-session-id	認証セッション ID
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して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（Login）] ページの **[ログイン時の問題（Problem logging in?）]** リンクをクリックして、手順 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

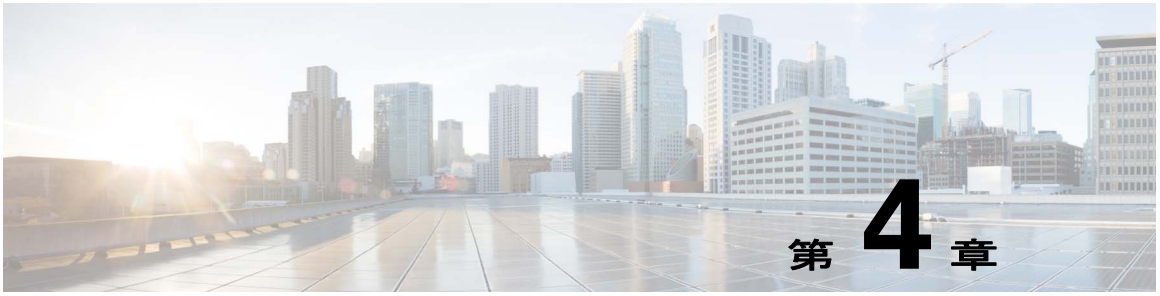
- 「モニタリング ノードの確認」（P.1-2）

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>
```



認可変更 REST API

この章では、Cisco Identity Services Engine のこのリリースでサポートされている次の個々の認可変更（CoA）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。

はじめに

CoA API コールは、Cisco ISE 導入で指定された Cisco Monitoring ISE ノードセッションに認証コマンドおよびセッション切断コマンドを送信する方法を提供します。

CoA セッション管理 API コール

CoA セッション管理 API コールにより、Cisco ISE 導入において、ターゲット Cisco Monitoring ISE ノードの指定セッションに再認証コマンドおよび切断コマンドを送信できるようにします。

- セッション再認証（Reauth）
- セッション切断（Disconnect）

セッション再認証 API コール

セッション再認証 API コールは次のタイプを構成します。

- REAUTH_TYPE_DEFAULT = 0
- REAUTH_TYPE_LAST = 1
- REAUTH_TYPE_RERUN = 2

Reauth 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 ノードで指定セッションへの送信後の Reauth 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="remoteCoA" type="coAResult"/>
<xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Reauth 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して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。

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/CoA/<specific-api-call>/<macaddress>/<reauthtype>）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 Reauth API コールを入力します。

```
https://acme123/admin/API/mnt/CoA/Reauth/server12/00:26:82:7B:D2:51/1
```

（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」（P.1-2）

Reauth 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 Reauth API コールを呼び出すときに返されるデータを示します。このコマンドの呼び出しから、次の 2 種類の結果が返されます。

- 「True」はコマンドが正常に実行されたことを示します。
- 「False」は（さまざまな条件により）コマンドが実行されなかったことを意味します。

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

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

セッション切断 API コール

セッション切断 API コールは、次の接続解除のポート オプション タイプを構成します。

- DYNAMIC_AUTHZ_PORT_DEFAULT = 0
- DYNAMIC_AUTHZ_PORT_BOUNCE = 1
- DYNAMIC_AUTHZ_PORT_SHUTDOWN = 2

Disconnect 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 ノードで指定セッションへの送信後の Disconnect 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="remoteCoA" type="coAResult"/>

  <xs:complexType name="coAResult">
    <xs:sequence>
      <xs:element name="results" type="xs:boolean" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="requestType" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Disconnect 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して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。

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/CoA/<Disconnect>/<serverhostname>/<macaddress>/<portoptiontype>/<nasipaddress>/<destinationipaddress>）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 Disconnect API コールを入力します。

```
https://acme123/admin/API/mnt/CoA/Disconnect/server12/00:26:82:7B:D2:51/2/10.10.10.10/192.168.1.1
```

（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」（P.1-2）

Disconnect 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 Disconnect API コールを呼び出すときに返されるデータを示します。このコマンドの呼び出しから、次の 2 種類の結果が返されます。

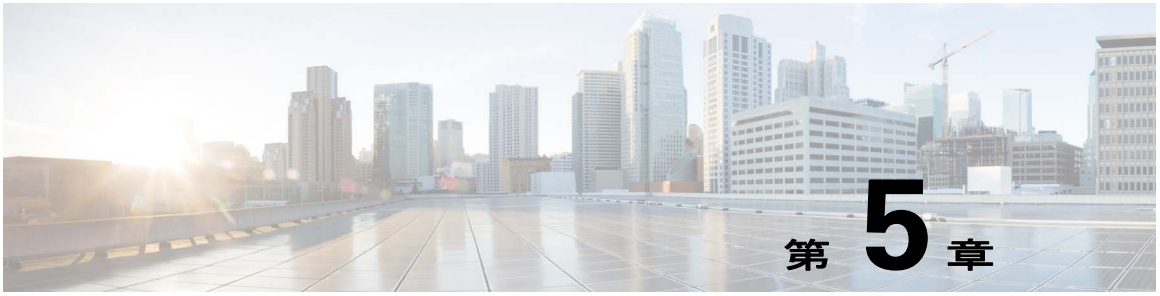
- 「True」はコマンドが正常に実行されたことを示します。
- 「False」は（さまざまな条件により）コマンドが実行されなかったことを意味します。

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

-
<remoteCoA requestType="reauth">
<results>true</results>
</remoteCoA>
```

PART 2

Cisco ISE 外部 RESTful サービス API



ERS API の概要

概要

この章では、サポートされている外部 RESTful サービス API および関連 API コールを使用する際のガイドラインおよび例を示します。これらのコールを使用すると、Cisco ISE リソースに対して CRUD（作成、読み取り、更新、削除）操作を実行できます。

サポートされる Cisco ISE リソース

Cisco ISE 外部 RESTful サービスを使用すると、次のタイプの ISE リソースに対して操作を実行できます。

- エンドポイント
- エンドポイント ID グループ
- ゲスト ユーザ
- ID グループ
- 内部ユーザ
- ポータル
- プロファイラのポリシー
- ネットワーク デバイス
- ネットワーク デバイス グループ
- セキュリティ グループ

すべての要求および応答の例については、第 7 章「外部 RESTful サービス API の操作」を参照してください。

外部 RESTful サービス API の認証および承認

外部 RESTful サービス API は HTTPS プロトコルおよび REST 方法論に基づいており、ポート 9060 を使用します。

外部 RESTful サービス API は、基本認証をサポートしています。認証クレデンシャルは、暗号化され、要求ヘッダーの一部となっています。

ISE 管理者は、外部 RESTful サービス API を使用して操作を実行するための特権をユーザに割り当てる必要があります。ISE 管理者は、外部 RESTful サービス API を使用して操作を実行する次の 2 種類のロールを割り当てることができます。

- 外部 RESTful サービス管理者 : すべての ERS API へのフルアクセス（GET、POST、DELETE、PUT）。
- 外部 RESTful サービス オペレータ：読み取り専用アクセス（GET 要求のみ）。

必要な権限がない場合に外部 RESTful サービス API を使用して操作を実行しようとすると、エラー応答を受信します。

関連項目

- 新しい Cisco ISE 管理者の作成
- 「スポンサーの認証および認可」（P.6-1）

GUI からの外部 RESTful サービス API のイネーブル化

Cisco ISE REST API 用に開発されたアプリケーションから Cisco ISE にアクセスできるようにするには、Cisco ISE REST API をイネーブルにする必要があります。Cisco REST API は HTTPS ポート 9060 を使用します。このポートはデフォルトでは閉じられています。Cisco ISE REST API が Cisco ISE 管理用サーバでイネーブルになっていない場合、クライアント アプリケーションは、サーバから Guest REST API 要求に対するタイムアウト エラーを受信します。

すべてのタイプの外部 RESTful サービス要求はプライマリ ISE ノードに限り有効です。セカンダリ ノードは読み取りアクセス（GET 要求）に対応します。

手順

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　ユーザ名および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。

ステップ 4　**[管理（Administration）] > [設定（Settings）] > [ERS 設定（ERS Settings）]** の順に選択します。

ステップ 5　プライマリ管理ノードの **[読み取り/書き込み用に ERS をイネーブル化（Enable ERS for Read/Write）]** を選択します。

ステップ 6　セカンダリ ノードがある場合は、その他すべてのノードの **[読み取り用に ERS をイネーブル化（Enable ERS for Read）]** を選択します。

ステップ 7　**[送信（Submit）]** をクリックします。

（注）　すべての REST 操作が監査され、ログがシステム ログに記録されます。

関連項目

- 「外部 RESTful サービス API コールを使用するための前提条件」（P.7-1）

外部 RESTful サービス API のステータス

GUI の **[管理（Administration）] > [設定（Settings）] > [ERS 設定（ERS Settings）]** ページで、外部 RESTful サービス API がプライマリおよびセカンダリ ノードに対してイネーブルになっているかどうかを確認できます。

外部 RESTful サービス API は、デフォルトではイネーブルになっていません。それらをイネーブルにする前に外部 RESTful サービス API コールを呼び出そうとすると、エラー応答を受信します。

外部 RESTful サービス API にはデバッグ ロギング カテゴリがあります。このカテゴリは、Cisco ISE GUI のデバッグ ログ ページからイネーブルにすることができます。

（注）　詳細については、『*Cisco Identity Services Engine Admin Guide, Release 1.3*』の「Debug Log Configuration Options」セクションを参照してください。

データの検証

サーバに送信された CRUD データは、Cisco ISE が GUI に使用するのと同じ検証ルールで検証されます。すべての検証は、検証レイヤに一元化されます。ポストされたすべての XML データはスキーマと照合して検証されます。

2 種類の検証（データ検証と構造検証）が実行されます。データ検証は、必須フィールド、フィールド長、タイプなどのデータが ISE に準拠していることを検証します。一方、構造検証は、フィールド順序、名前などのスキーマに対して行われます。

名前空間

次のようにリソースの名前と URI 内に、厳密な名前空間を維持する必要があります。

- 内部ユーザ ID、エンドポイント、エンドポイント グループ、および ID グループ
- SGT の SGA
- 応答メッセージに表示される検索結果など、その他すべてのオブジェクト リソースの外部 RESTful サービス（クライアントは要求内の ERS の名前空間を送信する必要があります）

Accept/Content-Type ヘッダーには、次の名前空間を含める必要があります。

```
application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml
```

例：`application/vnd.com.cisco.ise.identity.internaluser.1.0+xml`

要求 XML には、次のように名前空間の定義が含まれている必要があります。

```
identity.ers.ise.cisco.com

sga.ers.ise.cisco.com
```

例：

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns:endpoint xmlns:ns="identity.ers.ise.cisco.com" id="id">
<group>Profiled</group>
…
</ns:endpoint>
```

外部 RESTful サービス SDK

外部 RESTful サービス SDK を使用して、独自ツールの構築を開始できます。次の URL から外部 RESTful サービス SDK にアクセスできます。`https://<ISE-ADMIN-NODE>:9060/ers/sdk`

外部 RESTful サービス SDK には、外部 RESTful サービス管理ユーザのみがアクセスできます。SDK は、次のコンポーネントで構成されています。

- クイック リファレンス API マニュアル
- すべての利用可能な API 操作のリスト
- ダウンロード可能なスキーマ ファイル
- ダウンロード可能な Java のサンプル アプリケーション
- cURL スクリプト形式の使用例
- Chrome POSTMAN の使用方法

外部 RESTful サービス スキーマ ファイル

外部 RESTful サービス SDK には、ISE ERS インターフェイスでサポートされるオブジェクトの構造を説明する 4 つの XSD スキーマ ファイルが付属しています。

次の 4 つの XSD ファイルがあります。

- ers.xsd
- identity.xsd
- sga.xsd
- network.xsd

JAXB などの使用可能なツールとともにスキーマを使用して、スキーマ クラスを生成できます。

HTTP クライアントを開発し、またはサード パーティ HTTP クライアント コードを使用して、XSD ファイルから生成されたスキーマ クラスと統合できます。

（注） コンテンツで送信される XML は、スキーマに対して検証されます。したがって、XML のフィールド順序と構文はスキーマに表示されるものと同じである必要があります。そうでない場合、Bad Request ステータス コードを受信します。

スキーマ ファイルのダウンロード

手順

ステップ 1 外部 RESTful サービス SDK ページにログインするには、ブラウザのアドレス バーに次の URL を入力してください。`https://<ISE-ADMIN-NODE>:9060/ers/sdk`

ステップ 2 外部 RESTful サービス管理者のユーザ名と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。

ステップ 4 **[ダウンロード（Downloads）]** カテゴリで、**[Schema Jar (ers-schema-1.3-iteration-01-SNAPSHOT.jar)]** をクリックします。

ステップ 5 ファイルをローカル マシンに保存します。

関連項目

- 「外部 RESTful サービス API の認証および承認」（P.5-2）

外部 RESTful サービス要求と応答

ここでは、要求および応答ヘッダー内の情報と、ERS から返されるステータス コードについて説明します。

外部 RESTful サービス要求ヘッダー

表 5-1　ERS 要求ヘッダー

ヘッダー	サポートされる値	用途	必須
Accept	Guest REST API リソースのメディア タイプ	このクライアントが受け入れようとしているメディア タイプを、リソースのバージョンを含めて、サーバに示します。	GET/GET ALL/DELETE/GET VERSION 操作（これらには、メッセージ本文は含まれません）で、必須。
Authorization	「基本」とユーザ名およびパスワード（RFC 2617 に準拠）	この要求を実行する許可ユーザを識別します。	すべての要求で、必須。
Content-Length	Guest REST API リソースのメディア タイプ	このクライアントが受け入れようとしているメディア タイプを、リソースのバージョンを含めて、サーバに示します。	メッセージ本文を含む要求で、必須。
Content-Type	要求メッセージ本文を示すメディア タイプ	要求メッセージ本文の表現と構文について示します。	メッセージ本文を含む要求で、必須。

外部 RESTful サービス応答ヘッダー

表 5-2　ERS の必須応答ヘッダー

ヘッダー	サポートされる値	用途	必須
Content-Length	応答メッセージ本文の長さ（バイト単位）。	メッセージ本文のサイズを示します。	メッセージ本文を含む応答で、必須。
Content-Type	応答メッセージ本文を示すメディア タイプ。	応答メッセージ本文の表現と構文について示します。	メッセージ本文を含む応答で、必須。
Location	新しく作成されたリソースの正規の URI。	新しく作成されたリソースの表現を要求するために使用できる新しい URI を返します。	URI でアクセスできる新しいサーバ側リソースを作成する要求への応答で、必須。

共通の外部 RESTful サービスの HTTP ステータス コード

表 5-3　ERS によって返される HTTP 応答コードの説明

HTTP ステータス	説明
200 OK	要求は正常に完了しました。この要求により URI でアドレス指定可能な新しいリソースが作成され、新しいリソースの表現を含む応答本文が返される場合は、新規作成されたリソースの正規の URI を含む Location ヘッダーとともに 200 ステータスが返されます。
201 Created	新しいリソースを作成する要求は完了しましたが、新しいリソースの表現を含む応答本文は返されていません。新しく作成されたリソースの正規の URI を含む Location ヘッダーも返す必要があります。
202 Accepted	要求が受け入れられ処理されていますが、処理が完了していません。HTTP/1.1 仕様に従って、返されるエンティティ（ある場合）は、要求の現在のステータスの説明、およびステータス モニタへのポインタまたは要求の実行が期待できる時間の概算へのポインタを含む必要があります。
204 No Content	サーバは要求を満たしましたが、応答メッセージ本文を返す必要はありません。
400 Bad Request	欠落した情報または無効な情報が含まれるため（入力フィールドの検証エラーまたは必要な値の欠落など）、要求を処理できませんでした。
401 Unauthorized	この要求に含まれる認証クレデンシャルが欠落しているか、または無効です。
403 Forbidden	サーバはクレデンシャルを認識しましたが、この要求を実行する許可を所有していません。
404 Not Found	要求は、存在しないリソースの URI を指定しました。
405 Method Not Allowed	要求に指定された HTTP 動詞（DELETE、GET、HEAD、POST、PUT）は、この要求 URI でサポートされていません。
406 Not Acceptable	この要求によって識別されたリソースは、要求の Accept ヘッダーのメディア タイプの 1 つに対応する表現を生成できません。
409 Conflict	サーバによってサポートされるリソースの現在の状態に競合が生じるため、作成または更新要求を完了できませんでした（たとえば、既存のリソースに割り当てられている一意の ID で新しいリソースを作成する試行）。
415 Unsupported Media Type	Accept ヘッダーに指定されたメディア タイプは、サーバによってサポートされていません。これは、クライアント リソースのバージョンがサーバでサポートされなくなった場合の共通の応答です。
429 Too many requests	同時 ERS 要求が多すぎます。
500 Internal Server Error	サーバで、要求の処理を妨げる予期しない状態が発生しました。
501 Not Implemented	サーバは（現在）要求を処理するために必要な機能をサポートしていません。
503 Service Unavailable	サーバの一時的な過負荷またはメンテナンスのため、サーバは現在要求を処理できません。

（注） 応答ヘッダーで返されるステータス コードに加えて、各要求には、要求の性質に応じて、追加の XML コンテンツがある場合があります。

外部 RESTful サービス API のバージョン管理

外部 RESTful サービス API は、以前の Cisco ISE バージョンとの下位互換性を提供します。外部 RESTful サービス API には、API バージョン管理のバージョン設定メカニズムがあります。すべての非ゲスト リソースはバージョン 1.0 であり、下位互換性は必要ありません。

それぞれの RESTful リソースにはモデル バージョン（major.minor）があります。バージョンは、次のような構文を持つ要求ヘッダーの一部である必要があります。

```
application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml
```

たとえば、内部ユーザ リソース バージョン 1.0 を取得するには、次の要求を渡します。

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

要求の認証と承認後、バージョン一致のチェックが実行され、次の表に示される一致結果になります。

バージョンの一致	結果
送信されたバージョンなし	サーバは、ステータス 415「Unsupported Media Type」を返します
クライアントのバージョンとサーバのバージョンは等しい	サーバは、要求の処理を続行します。
クライアントのマイナー バージョンとサーバのマイナー バージョンは等しくない	サーバは、バージョン ギャップを示す応答の警告メッセージを追加し、要求の処理を続行します。
クライアントとサーバのメジャー バージョンが一致しない	サーバは、ステータス 415 および対応するエラー メッセージを返します。

（注）　各リソースには、サーバでサポートされているバージョンのリストを取得する API があります。

検索およびフィルタリング

すべてのフィルタリングおよび検索操作は、フィルタリングを使用して行われます。

GET 要求をリソース URI に送信して、リソースを検索します。デフォルトで、結果はデフォルト サイズが 20 の最初のページ（ページ インデックス = 0）に表示されます。フィルタ、ソート、およびページングのパラメータ（次のセクションで説明）を URI に追加することによって、クライアントはこの検索を制御できます。

ページング、フィルタ、またはソート要求で生じるリソースは、<resources> コレクションでバンドルされます。このコレクションには、各リソースの名前、ID、説明、および完全な表現へのリンクが含まれます。これによって、クライアントは容易にリソースにドリルダウンできるようになります。

外部 RESTful サービス API のフィルタリング パラメータ

フィルタのクエリー文字列パラメータを使用して簡単なフィルタ操作を実行できます。複数のフィルタを送信できます。すべてのフィルタ基準に共通の論理演算子はデフォルトで AND です。「filtertype=or」クエリー文字列パラメータを使用して、これを変更できます。

各リソース データ モデルの説明には、属性がフィルタ処理されたフィールドかどうか指定する必要があります。

たとえば、名が「a」で始まり、「Finance」ID グループに属する内部ユーザを取得するには、次の要求を渡します。

```
GET
/ers/config/internaluser/?page=0&size=20&sortacs=name&filter=name.STARTSW.a&filter=identit
yGroup.EQ.Finance
```

次の表に、使用可能なフィルタ演算子を示します。

パラメータ	説明
EQ	等しい
GT	次の値より大きい
LT	次の値より小さい
STARTSW	次で始まる
ENDSW	次で終了
CONTAINS	次の文字列を含む

次の表に、各リソースのフィルタリング可能な属性のリストを示します。

リソース	フィルタリング可能な属性
エンドポイント	mac、portalUser、profile、profileId、staticGroupAssignment、staticProfileAssignment
内部ユーザ	name
エンドポイント ID グループ	name
ID グループ	name
SGT	name

外部 RESTful サービス API 用のページング パラメータ

すべての外部 RESTful サービスの検索結果はページングされます。ページング パラメータは、クエリー パラメータを使用して URI で渡されます。

たとえば、「name」フィールドで昇順でソートされた内部ユーザの最初から 20 のレコードを取得するには、次の要求を渡します。

```
GET /ers/config/internaluser?page=0&size=20&sortasc=name
```

次の表に、使用可能なページング パラメータを示します。

パラメータ	説明	デフォルト値
page	ページの開始インデックス	0
size	ページ サイズ	20（最大 = 100）
sortasc	ソートに使用可能な一連のフィールドの中で、昇順のソート フィールド。（英文字の場合、A から Z の順序でソートされます）。	name
sortdsc	ソートに使用可能な一連のフィールドの中で、降順のソート フィールド。（英文字の場合、Z から A の順序でソートされます）。	name

外部 RESTful サービス システム フロー

共通の外部 RESTful サービス フローは、クライアントから送信された HTTPS 要求とサーバからの HTTPS 応答で構成されます。フローは要求タイプ、URI、要求ヘッダー、応答ヘッダー、および応答内容によって異なります。

図 5-1　ERS の成功フロー シーケンス

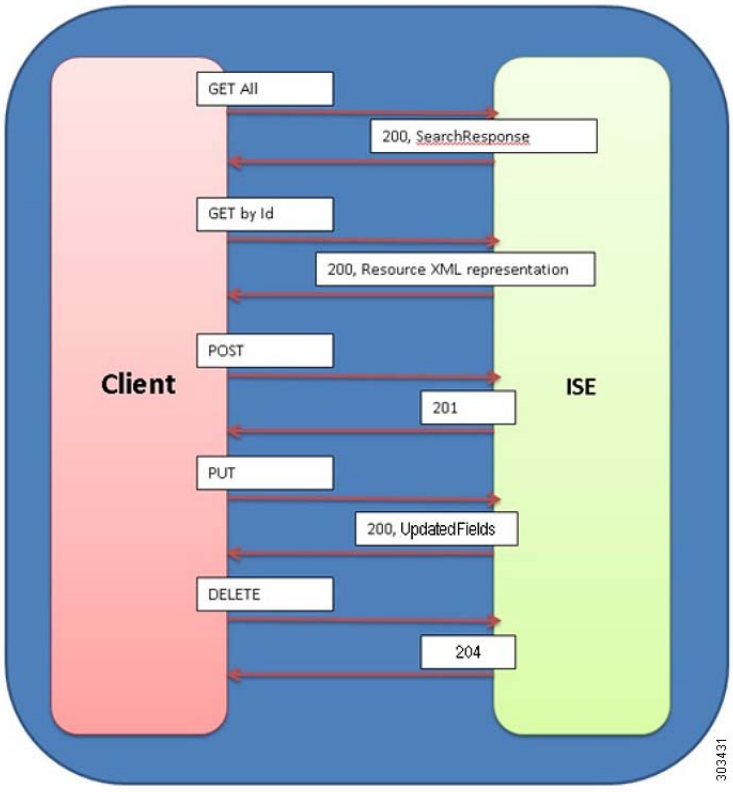
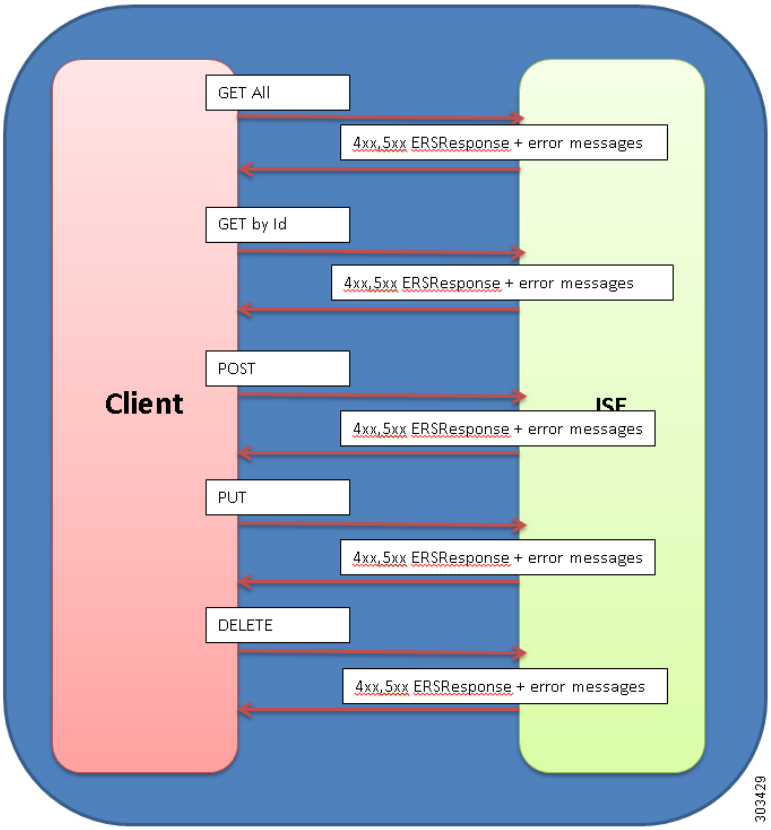


図 5-2　　ERS の失敗フロー シーケンス



ハイパーリンク

XML 表現内でのハイパーリンクの使用は、アプリケーション ステート（HATEOS）エンジンとしての Hypermedia の最も強力な特性の 1 つです。ハイパーリンクは、リソースに特定 URI の表現があるクライアントへの伝送に主に使用されます。

外部 RESTful サービス リンクは、次の名前空間で宣言されるリンクの Atom 定義に合わせて設定されます。

http://www.w3.org/2005/Atom namespace

次の表に、外部 RESTful サービスに必要なリンク属性を示します。

属性	説明
Href	リンクの URI が含まれます。
Rel	元々「relation」を意味するこの属性は、次のリンク タイプを示します。 • self：現在の表現を更新するリンク • next：次のページを取得するための集合 • info：リソースの詳細の取得
Type	リンクの URI に対してサーバが返すことがある表現のメディア タイプに対するヒントです。通常は「application/xml」です。

検索結果内のリンクの例

次に、検索結果内のリンクの例を示します。

```
<ns2:searchResult xmlns:ns2="ers.ise.cisco.com"   total="1163">
<link rel="self"
href="http://cisco.com/ers/config/internaluser?page=0&size=20"
type="application/xml"/>
<link rel="next" href="http://cisco.com/ers/config/internaluser?page=20&size=20"
type="application/xml"/>
<resources>
<link rel="john doe" href="http://cisco.com/ers/config/internaluser/333"
type="application/xml"/>
<link rel="jeff smit" href="http://cisco.com/ers/config/internaluser/444"
type="application/xml"/>
.
.
.
</resources>
</ns2:searchResult>
```

バルク操作

バルク要求によって、1 つの要求で最大 500 の操作（ID 別に 5000 の操作）を送信できます。すべてのリソースの作成、更新、および削除操作と、いくつかのリソース固有の操作（エンドポイントの登録など）を実行できます。

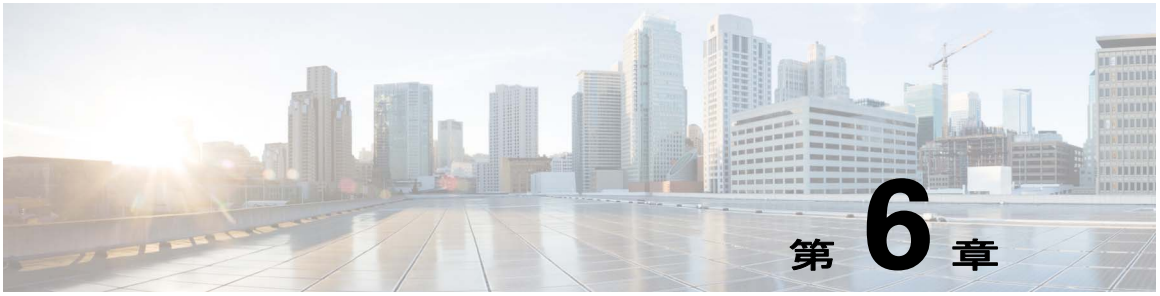
要求内の操作はすべて同じタイプである必要があります。さまざまなタイプが混在したリソース要求を送信することはできません。

各リソースには独自のトランザクションがあります。マルチスレッドで実行されるため、トランザクションの順序は保証されません。

Cisco ISE サーバは、要求を解析し、その構造を検証します。要求が有効で、その他のバルク操作が進行中でなければ、実行が開始され、サーバはステータス コード `202 (ACCEPTED)` と LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID によって、後でバルク ステータスの取得操作を使用する際にバルク ステータスを追跡することができます。ステータス レポートは、操作の開始時刻から 2 時間以上経ってから使用できます。

リソースでの実行中に障害が発生すると、障害がステータス レポートに記録され、実行は次のリソースに進みます。

一度に可能なバルク実行は 1 つのみです。別のバルク実行の実行中にバルク要求がポストされると、サーバは応答ステータス `503 (Service Unavailable)` とともにクライアントに後で再試行するように促すメッセージを返します。



第 6 章

Guest REST API

ゲスト ユーザ リソースの API

Cisco Guest API は、REST（Representational State Transfer）ベースの操作セットで、シスコのゲスト ユーザに安全で認証済みのアクセスを提供し、シスコのゲスト ユーザを管理します。この API を使用することにより、ゲスト ユーザを作成、読み取り、更新、削除、および検索することができます。

この API をコールする場合は、ゲスト ユーザを管理するための Cisco ISE スポンサー ポータルを使用するスポンサーと同様に、API をコールすることになります。この API を使用するには、まずスポンサー ポータルへのアクセスをイネーブルにしてから、ISE でスポンサー認証を設定する必要があります。

すべての要求および応答の例については、「ゲスト ユーザの外部 RESTful サービス API」（P.7-24）を参照してください。

スポンサーの認証および認可

スポンサーは、スポンサー ポータルを使用してゲスト ユーザを作成および管理できる特別なタイプの Cisco ISE ユーザです。Guest REST API は、スポンサー ポータルと同じ機能を備えています。Guest REST API の認証は、スポンサーを認証するプロセスとほぼ同じです。スポンサー グループのポリシーに指定された権限が、Guest REST API に適用されます。

スポンサーおよびスポンサー グループの詳細については、『Cisco ISE User Guide』を参照してください。

はじめる前に

他の ISE ユーザと同様に、Cisco ISE では、ローカル データベースあるいは外部の Lightweight Directory Access Protocol（LDAP）または Microsoft Active Directory ID ストアによりスポンサーを認証します。外部 ID ストアを使用していない場合は、Cisco ISE でユーザを作成する必要があります（**[管理（Administration）] > [ID 管理（Identity Management）] > [ID（Identities）] > [ユーザ（Users）]**）。

手順

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　ユーザ名および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。

ステップ 4　適切な ID グループにユーザを割り当てます。

a. [管理（Administration）] > [アイデンティティの管理（Identity Management）] > [グループ（Groups）] > [ID グループ（Identity Groups）] を選択します。

b. 新規グループを作成するか、既存グループを編集します。Cisco ISE には、次のスポンサー ユーザ ID グループがデフォルトで含まれています。

- SponsorAllAccount
- SponsorGroupAccounts
- SponsorOwnAccounts

c. メンバーのリストにユーザを追加します。

d. [保存（Save）] をクリックします。

ステップ 5　ユーザの ID グループをスポンサー グループに追加します。

a. **[ゲスト アクセス（Guest Access）] > [設定（Configure）] > [スポンサー グループ（Sponsor Groups）]** を選択します。

b. 新規スポンサー グループを作成するか、既存スポンサー グループを編集します。

c. **[メンバー（Members）]** をクリックします。

d. ユーザの ID グループをスポンサー グループ メンバーのリストに追加し、**[OK]** をクリックします。

e. ユーザがスポンサーになることができるゲスト タイプおよび場所を追加します。

f. [プログラマチック インターフェイスを介した ISE ゲスト アカウントへのアクセス（REST API）（**Access ISE guest accounts via the programatic interface (REST API)**）] チェックボックスをオンにします。

g. **[保存（Save）]** をクリックします。

ステップ 6　Sponsor_Portal_Sequence からユーザの ID ソースにアクセスできることを確認します。

a. **[管理（Administration）] > [ID 管理（Identity Management）] > [ID ソース順序（Identity Source Sequences）] > [Sponsor_Portal_Sequence]** を選択します。

b. ユーザの ID ソース データベースが認証検索リストに選択されていない場合は、これを選択します。

c. **[保存（Save）]** をクリックします。

関連項目

スポンサーおよびスポンサー グループの詳細については、『Cisco ISE User Guide』を参照してください。

Guest REST API 要求

API に対する要求には次の特性があります。

- ユーザ要求は、HTTPS で Cisco ISE サーバに送信され、応答が返されます。
- すべての API にアクセスする URI は、プライマリのドメイン名になります。

```
https://<ISE-ADMIN-NODE>:9060/ers/config
```

- API 要求は、大文字と小文字が区別され、このマニュアルで示すように入力する必要があります。
- 各ゲストの要求では、Content-Type を次のように設定する必要があります。

```
application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

- 各ゲストの要求では、Accept を次のように設定する必要があります。

```
application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

- 各ゲストのバルク要求では、Content-Type を次のように設定する必要があります。

```
application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml
```

- 各ゲストのバルク要求では、Accept を次のように設定する必要があります。

```
application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml
```

- また、応答ペイロードを圧縮する `Accept-Encoding:gzip` ヘッダーをオプションで含めることもできます。

要求構造

次の表に、Cisco Guest REST API と使用できる要求操作タイプを示します。

表 6-1　要求メソッド タイプ

要求タイプ	操作
GET	すべてのリソースを取得（検索）し、ID によってリソースを取得し、リソースのバージョン情報を取得します。
POST	新規ゲスト ユーザを作成します。
PUT	ゲスト ユーザを更新します。
DELETE	ゲスト ユーザを削除します

次の表に、要求の必須ヘッダーを示します。

表 6-2　必須要求ヘッダー

ヘッダー	値	説明
ACCEPT	Guest REST API リソースのメディア タイプ	このクライアントが受け入れようとしているメディア タイプを、リソースのバージョンを含めて、サーバに示します。
AUTHORIZATION	「基本」とユーザ名およびパスワード（RFC 2617 に準拠）	この要求を実行する許可ユーザを識別します。
CONTENT-TYPE	要求メッセージ本文を示すメディア タイプ	要求メッセージ本文の表現と構文について示します。

関連項目

「Content-Type ヘッダーおよび Accept ヘッダー」（P.7-24）

要求コンテンツ

以下の要求の例に、POST（作成）操作を使用してゲスト ユーザ アカウントを作成するために必要な XML コンテンツを示します。

（注）　JSON は、Guest REST API には使用されません。

POST 要求の例

```
POST https://<ISE-Admin-Node>:9060/ers/config/guestuser/

Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Content-Type: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization: Basic xxxxxxxxxxxxxxxx

<?xml version="1.0" encoding="UTF-8"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
    <guestAccessInfo>
        <fromDate>14/08/06 23:22</fromDate>
        <toDate>14/08/07 23:22</toDate>
        <validDays>1</validDays>
    </guestAccessInfo>
    <guestInfo>
        <company>New Company</company>
        <emailAddress>john@example.com</emailAddress>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <notificationLanguage>English</notificationLanguage>
        <password>12345</password>
        <phoneNumber>9999998877</phoneNumber>
        <smsServiceProvider>Global Default</smsServiceProvider>
        <userName>autoguestuser1</userName>
    </guestInfo>
    <guestType>Daily</guestType>
    <personBeingVisited>sponsor</personBeingVisited>
    <portalId>portal101</portalId>
    <reasonForVisit>interview</reasonForVisit>
</ns2:guestuser>
```

バルク実行

バルク要求を使用すると、1 つの要求で最大 500 の操作、または ID に基づいて 5000 の操作を送信できます。

要求内の操作はすべて同じタイプである必要があります。さまざまなタイプが混在したリソース要求を送信することはできません。

各リソースには独自のトランザクションがあります。マルチスレッドで実行されるため、トランザクションの順序は保証されません。

Cisco ISE サーバは、要求を解析し、その構造を検証します。要求が有効で、その他のバルク操作が進行中でなければ、実行が開始され、サーバはステータス コード `202 (ACCEPTED)` と LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID によって、後でバルク ステータスの取得操作を使用する際にバルク ステータスを追跡することができます。ステータス レポートは、操作の開始時刻から 2 時間以上経ってから使用できます。

リソースでの実行中に障害が発生すると、障害がステータス レポートに記録され、実行は次のリソースに進みます。

一度に可能なバルク実行は 1 つのみです。別のバルク実行の実行中にバルク要求がポストされると、サーバは応答ステータス `503 (Service Unavailable)` とともにクライアントに後で再試行するように促すメッセージを返します。

バルク ステータスの取得などのバルク実行は、異なるヘッダーを使用します。

- Content Type：`application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`
- Accept：`application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`

関連項目

- 「ゲスト ユーザのバルク実行の開始」（P.7-39）

Guest REST API 応答

各要求の後に、標準ヘッダーのサーバ HTTPS 応答と応答コンテンツが続きます。

応答ステータス コード

応答には、HTTP ステータス コードが含まれます。Guest REST API の場合、次のコードが含まれます。

- 20x：成功した操作
- 4xx：クライアント エラー
- 5xx：サーバ エラー

次の表に、ステータス コードの詳細を示します。

表 6-3　応答コンテンツ タイプ

要求タイプ	ステータス	応答ペイロード
ID によるゲスト ユーザの取得	200	XML 表現のゲスト ユーザ
すべて	4xx（クライアント エラー）	エラーを説明するメッセージ（サポートされていないバージョン、不正な URI など）での応答
すべて	5xx（サーバ エラー）	エラーを説明するメッセージ（ランタイム例外など）での応答
ゲスト ユーザの作成 [POST]	201	情報や警告がない場合、コンテンツは返されません。新しいゲスト ユーザ ID が応答「Location」ヘッダーで送信されます。追加情報または警告がある場合は、応答でカプセル化されます。
ゲスト ユーザの更新 [PUT]	200	更新されたフィールドを返します。
ゲスト ユーザの削除 [Delete]	204	コンテンツなし

関連項目

「応答エラー メッセージ」（P.6-7）

応答構造

次の表に、応答のヘッダーを示します。

表 6-4　必須応答ヘッダー

ヘッダー	値	説明
LOCATION	新しく作成されたリソース ID	POST のみ:新しいリソース ID を（URI 表現として）保持します
CONTENT-TYPE	応答メッセージ本文を示すメディア タイプ	応答メッセージ本文の表現と構文について示します

ゲスト パスワード

ISE は、ゲストの作成時に自動的にパスワードを生成します。特定のゲスト ユーザで PUT を呼び出し、空のパスワードを提供することによって、Guest REST API でゲストのパスワードをリセットできます。これは、スポンサー ポータルからスポンサーがゲストのパスワードをリセットする方法と同じです。

GET 操作を使用して、ゲスト ユーザの情報を取得し、パスワードを表示します。Cisco ISE のゲスト パスワードは、パスワードが次のような場合、GET 操作への応答に表示されます。

1. ISE によって自動的に生成されている。

2. この API またはスポンサー ポータルでリセットされている。

いくつかのゲスト フローでは、ゲストは自身のパスワードを変更できます。ゲストが変更した Cisco ISE のゲスト パスワードは、スポンサー ポータルに表示されません。また、REST API でも表示されません。

スポンサーに、スポンサー グループ権限を介してパスワードをリセットする権限がある場合でも、パスワードをリセットできます。

関連項目

- 「ユーザ パスワードのリセットの例」（P.7-33）
- ゲスト ユーザのパスワード ポリシーの詳細については、『Cisco ISE Admin Guide』を参照してください。

応答エラー メッセージ

POST または PUT 要求を実行するときは、特定のヘッダーを使用する必要があります。そうしなければ、エラーが発生します。

操作の修正を可能にする詳細なメッセージが含まれているログファイルは、次の Cisco ISE UI からアクセスできます。

[操作（Operations）] > [ノード（Node）] > [デバッグ ログ（Debug Logs）] > [guest.log]

表 6-5　エラー コード

エラー コード	説明	HTTP ステータス
Resource version exception	要求コンテンツで送信されたリソースのバージョンが、サーバでサポートされていません。	415
Resource media type exception	ACCEPT または Content-Type ヘッダーでクライアントから送信されたメディア タイプが無効です。	415
Unsupported resource exception	URI に示されたリソースは、サーバでサポートされていません。	400
Unsupported method exception	要求メソッド タイプは、指定された URI でサポートされていません。	400
Query string validation exception	検索フィルタまたはページング パラメータが無効です。	400
Schema validation exception	API が認識しないフィールドが含まれていたり必須のフィールドが含まれていないなど、要求の XML コンテンツがリソースの API スキーマ ルールに準拠していません。	400
Application resource validation exception	API は、属性値に無効な文字を使用している要求コンテンツなど、一部の要求コンテンツを検証できません。	400
Unauthorized user	スポンサーのユーザ名とパスワードが無効です。	401

表 6-5　エラー コード

エラー コード	説明	HTTP ステータス
Internal exception	実行時に予期しない内部サーバ エラーが発生しました。	500
Conversion exception	一部の内部変換は、内部例外として処理する必要があります。	500
CRUD operation exception	CRUD 操作の実行は、内部例外として処理する必要があります。	500

サポートされないメディア タイプの例

次に、クライアントが ACCEPT ヘッダーでサポートされていないメディア タイプを送信した際に発生する応答の例を示します。

応答

```
STATUS 415 Unsupported Media Type
Cache-Control: no-cache
Content-Length: 411
Content-Type: application/vnd.com.cisco.ise.ers.ersresponse.1.0+xml
Date: Wed, 11 Dec 2013 05:27:37 GMT
Expires: Wed, 31 Dec 2015 16:00:00 PST
Pragma: No-cache

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:ersResponse
    xmlns:ns2="ers.ise.cisco.com" operation="GET-getAll-guestuser">
    <link type="application/xml"
href="https://<ISE-Admin-Node>:9060/ers/config/guestuser/" rel="related"/>
    <messages>
        <ns2:message type="ERROR" code="Resource media type exception">
            <title>Wrong media type, check Accept request header.</title>
        </ns2:message>
    </messages>
</ns2:ersResponse>
```

バージョン設定

Guest REST API for Cisco ISE リリース 1.3 は、バージョン 2.0 です。将来のバージョンと互換性があるように設計されています。

この REST API は、Guest REST API の以前のアルファ バージョンおよびベータ バージョンと互換性がありません。

それぞれの RESTful リソースにはモデル バージョン（major.minor）があります。バージョンは、次のような構文を持つ要求ヘッダーの一部である必要があります。

```
application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor
version>+xml
```

たとえば、ゲスト ユーザ リソース バージョン 2.0 を取得するには、次の要求を渡します。

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/333
Authorization: Basic xxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

要求の認証および承認後、バージョン一致のチェックが実行され、次のいずれかの一致の結果になります。

表 6-6　　バージョン一致の結果

バージョン一致	結果
送信されたバージョンなし	サーバは、ステータス 415「Unsupported Media Type」を返します。
クライアントのバージョンとサーバのバージョンは等しい	サーバは、要求の処理を続行します。
クライアントのマイナー バージョンとサーバのマイナー バージョンは等しくない	サーバは、バージョン ギャップを示す応答の警告メッセージを追加し、要求の処理を続行します。
クライアントとサーバのメジャー バージョンが一致しない	サーバは、ステータス 415 および対応するエラー メッセージを返します。

検索およびフィルタリング

すべてのフィルタリングおよび検索操作は、フィルタリングを使用して行われます。

GET 要求をリソース URI に送信して、リソースを検索します。デフォルトで、結果はデフォルト サイズが 20 の最初のページ（ページ インデックス = 0）に表示されます。フィルタ、ソート、およびページングのパラメータ（次のセクションで説明）を URI に追加することによって、クライアントはこの検索を制御できます。

ページング、フィルタ、またはソート要求で生じるリソースは、<resources> コレクションでバンドルされます。このコレクションには、各リソースの名前、ID、説明、および完全な表現へのリンクが含まれます。これによって、クライアントは容易にリソースにドリルダウンできるようになります。

フィルタリング パラメータ

フィルタリングは、フィルタのクエリー文字列パラメータを使用して実行できます。フィルタは、ドットで区切られたフィールド演算子、パラメータ、および値の 3 ビット バイトの構造です。たとえば、ユーザ名が文字「g」から始まるすべてのゲスト ユーザを検索するには、filter=name.STARTSW.g を使用します。

フィルタに複数のパラメータを使用すると、結果はそれらのパラメータの AND になります。つまり、結果として、API に送信されたパラメータのすべてに一致するユーザが得られます。各リソースの説明で、属性がフィルタ対象のフィールドかどうかを指定します。

（注） フィルタ値が無効な場合、ステータス 400 (Bad Request) と対応するメッセージが返されます。

次の表に、フィルタ クエリーで使用できるパラメータを示します。

表 6-7　フィルタリング パラメータ

パラメータ	説明
firstName	ゲストの名
lastName	ゲストの姓
emailAddress	ゲストの電子メール アドレス
userName	ゲスト アカウント ユーザ名
creationTime	ゲスト アカウントの作成日時
toDate	ゲスト アカウントの有効期限
guestType	ゲストのタイプ
company	ゲストの会社名
phoneNumber	ゲストの電話番号
groupTag	グループ
sponsor	ゲストのスポンサー
status	ゲスト アカウントのステータス
name	リソース識別子

次の表に、フィルタ クエリーで使用できる操作を示します。

表 6-8　使用可能なフィルタ操作

パラメータ	説明
EQ	等しい
GT	次の値より大きい
LT	次の値より小さい
STARTSW	次で始まる
ENDSW	次で終わる
CONTAINS	次の文字列を含む

フィルタリング例

名が「br」から始まるゲスト ユーザの GET 要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?filter=firstName.STARTSW.br
```

名が「b」から始まり、電子メール アドレスに「bob」が含まれるゲスト ユーザの GET 要求

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?filter=firstName.STARTSW.b&filter=email
Address.CONTAINS.bob
```

ステータスが「Approved」で有効期限が 10 月のゲスト ユーザの GET 要求

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?filter=status.EQ.Approved&filter=toDate
.STARTSW.10
```

関連項目

「ゲスト ユーザの取得」（P.7-25）

ページ サイズ パラメータ

デフォルトの検索結果は、ページあたり 20 リソースです。ページの番号付けはページ「0」から始まります。ページあたりの最大リソース数は 100 を超えることはできません。パラメータ値が不正な場合、ステータス `400 (Bad Request)` と対応するメッセージが返されます。

ページ サイズ パラメータは、表 6-9 で説明されているページング パラメータを使用することでデフォルトをオーバーライドできます。

次の例では、ページ サイズは 50 リソースに変更されています。

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?filter=name.STARTSW.b&filter=emailAddress.CONTAINS.bob&size=50&page=0
```

ソート パラメータ

デフォルトでは、ソート カラムは `name`、ソート方向は `sortasc` です。次の例に示すように、パラメータを使用してデフォルトをオーバーライドできます。

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?size=50&page=0&sortdsc=name
```

ソートに使用できる値のリストは、表 6-7（P.6-10）を参照してください。

表 6-9　使用可能なページ設定およびソート設定

パラメータ	説明	デフォルト値
page	ページの開始インデックス	0
size	ページ サイズ	20（最大 = 100）
sortasc	ソートに使用可能な一連のフィールドの中で、昇順のソート フィールド。（英文字の場合、A から Z の順序でソートされます）。	`name`
sortdsc	ソートに使用可能な一連のフィールドの中で、降順のソート フィールド。（英文字の場合、Z から A の順序でソートされます）。	`name`

例：最初の 20 のゲスト ユーザ レコードを取得（GET）し、姓によって昇順にソートします

要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?page=0&size=20&sortasc=lastName
```

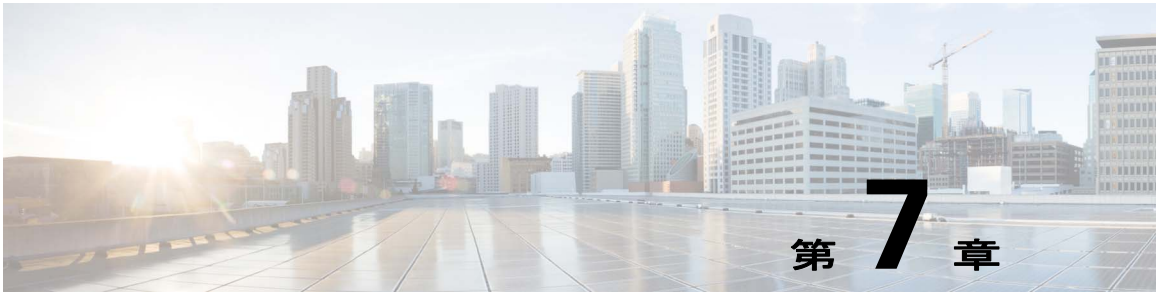
応答

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
    xmlns:ns2="ers.ise.cisco.com" total="22">
    <nextPage type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser?page=1&size=20&sortasc=lastName"
rel="next"/>
    <resources>
```

```
        <resource name="aname001" id="8e4bf290-6229-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-Admin-Node:9060/ers/config/guestuser/8e4bf290-6229-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="aname002" id="8fe86480-6229-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-Admin-Node:9060/ers/config/guestuser/8fe86480-6229-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
...
        <resource name="aname020" id="908b5b40-6229-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-Admin-Node:9060/ers/config/guestuser/908b5b40-6229-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
    </resources>
</ns2:searchResult>
```

関連項目

- 「「ilucky」で始まるユーザ名によるフィルタリングの例」（P.7-26）
- 「「ilucky」で始まるユーザ名および「J」で始まる姓によるフィルタリングの例」（P.7-27）
- 「名「John」によるフィルタリングおよびユーザ名によるソートの例」（P.7-28）



外部 RESTful サービス API の操作

概要

この章では、外部 RESTful サービス API コールの例を示し、その使用方法について説明します。外部 RESTful サービス API コールを発行する手順と、API 出力スキーマ ファイルの例、および返されるデータの例を示します。

外部 RESTful サービス API コールを使用するための前提条件

外部 RESTful サービス API コールを呼び出す前に、次の前提条件を満たす必要があります。

- GUI から外部 RESTful サービスをイネーブルにしておく必要があります。
- 外部 RESTful サービスの管理者権限が必要です。

JAVA、curl Linux コマンド、Python などの REST クライアントやその他のクライアントを使用して、外部 RESTful サービス API コールを呼び出すことができます。

関連項目

- 「GUI からの外部 RESTful サービス API のイネーブル化」（P.5-3）
- 「外部 RESTful サービス API の認証および承認」（P.5-2）

GetVersion

GetVersion 操作はすべての利用可能なリソースに共通です。これは、必要なリソースのバージョン情報を取得します。次の表に、この操作の主な特性を示します。

表 7-1　GetVersion 操作の主な特性

説明	指定したリソースのバージョン情報を取得します
概要	GET/ers/config/<resource-name>/versioninfo
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	バージョン情報
応答ステータス	200、400、401、403、404、415、500

GetVersion 操作の要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/<resource-type>/versioninfo
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.<resource-namespace>.1.0+xml
```

GetVersion 操作の応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.<resource-namespace>.versoininfo.1.0+xml
Content-Length: 122
{
<?xml version="1.0" encoding="UTF-8"?>
<ns2:versionInfo xmlns:ns2="ers.ise.cisco.com">
    <currentServerVersion>1.2</currentServerVersion>
    <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/<resource-type>/versioninfo" rel="self"/>
    <supportedVersions>1.0,1.1,1.2,1.3<supportedVersion>
</ns2:versionInfo>
}
```

内部ユーザの外部 RESTful サービス API

内部ユーザの外部 RESTful サービス API はすべての CRUD 機能をサポートしています。次の表に、内部ユーザに使用できる外部 RESTful サービス API を示します。

表 7-2　内部ユーザに使用できる外部 RESTful サービス API

オペレーション	HTTP メソッド	URL	内容	クエリー文字列
すべてのユーザの取得	GET	/ers/config/internaluser	適用対象外	Page、Size、sortacs または sortdsn、Filter
ユーザの取得	GET	/ers/config/internaluser/{internal user-id[1] }	適用対象外	
ユーザの作成	POST	/ers/config/internaluser/	internaluser	
ユーザの更新	PUT	/ers/config/internaluser/{internal user-id}	internaluser	
ユーザの削除	DELETE	/ers/config/internaluser/{internal user-id}	適用対象外	
InternalUser リソースのバージョン情報の取得	GET	/ers/config/internaluser/version	適用対象外	

1. 内部ユーザ ID は、Cisco ISE データベースに保存された UUID タイプです。

すべての内部ユーザの取得

この API コールを使用して、Cisco ISE に存在するすべての内部ユーザを取得できます。次の表に、この API コールの主な特性を示します。

表 7-3　Retrieve All Internal Users API コールの主な特性

説明	内部ユーザの集合を取得します
概要	GET /ers/config/internaluser
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、429、500

Retrieve All Internal Users API 要求の例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

Retrieve All Internal Users API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
    <resources>
        <ns2:resource description="description1" name="name1" id="id1"/>
        <ns2:resource description="description2" name="name2" id="id2"/>
    </resources>
</ns2:searchResult>
}
```

ID による内部ユーザの取得

この API コールを使用して、Cisco ISE で ID によって内部ユーザを取得できます。次の表に、この API コールの主な特性を示します。

表 7-4　Read Internal Users API コールの主な特性

説明	指定された内部ユーザを取得します
概要	GET /ers/config/internaluser/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Read Internal Users API 要求の例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.internaluser.1.0+xml
```

Read Internal Users API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
<ns3:internaluser description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <changePassword>true</changePassword>
    <customAttributes>
        <entry>
            <key>key2</key>
            <value>value3</value>
        </entry>
        <entry>
            <key>key1</key>
            <value>value1</value>
        </entry>
    </customAttributes>
    <email>email@example.com</email>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <identityGroups>identityGroups</identityGroups>
    <lastName>Doe</lastName>
    <password>12345</password>
</ns3:internaluser>
}
```

内部ユーザの作成

この API コールを使用して、Cisco ISE で内部ユーザを作成できます。外部 RESTful サービス API を使用して内部ユーザを作成する場合には、パスワードが必要です。次の表に、この API コールの主な特性を示します。

表 7-5　Create Internal Users API コールの主な特性

説明	指定された内部ユーザを作成します
概要	POST /ers/config/internaluser/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	InternalUser
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	201、400、401、403、415、429、500

Create Internal Users API 要求の例

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/
Authorization: Basic xxxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <changePassword>true</changePassword>
    <customAttributes>
        <entry>
            <key>key2</key>
            <value>value3</value>
```

```
        </entry>
        <entry>
            <key>key1</key>
            <value>value1</value>
        </entry>
    </customAttributes>
    <email>email@example.com</email>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <identityGroups>identityGroups</identityGroups>
    <lastName>Doe</lastName>
    <password>12345</password>
</ns3:internaluser>
}
```

Create Internal Users API 応答の例

```
HTTP/1.1 201 OK  (see the location header for the new user's ID)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
Location: https://<ISE-ADMIN-NODE>/ers/config/internaluser/444
```

内部ユーザの更新

この API コールを使用して、Cisco ISE で内部ユーザを更新できます。外部 RESTful サービス API を使用して内部ユーザを更新する際にパスワードが変更されない場合は、パスワードを「********」に設定する必要があります。次の表に、この API コールの主な特性を示します。

表 7-6　Update Internal Users API コールの主な特性

説明	指定された内部ユーザを更新します
概要	PUT /ers/config/internaluser/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	InternalUser
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、429、500

Update Internal Users API 要求の例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:internaluser description="description" name="name" id="333"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <changePassword>true</changePassword>
    <customAttributes>
        <entry>
            <key>key2</key>
            <value>value3</value>
```

```
        </entry>
        <entry>
            <key>key1</key>
            <value>value1</value>
        </entry>
    </customAttributes>
    <email>email@example.com</email>
    <enabled>true</enabled>
    <firstName>John</firstName>
    <identityGroups>IdentityGroups</identityGroups>
    <lastName>Doe</lastName>
    <password>********</password>
</ns3:internaluser>
}
```

Update Internal Users API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
    <updatedField field="lastname">
        <newValue>Doe</newValue>
        <oldValue>name</oldValue>
    </updatedField>
    <updatedField field="dentityGroups">
        <newValue>IdentityGroups</newValue>
        <oldValue>zzz</oldValue>
    </updatedField>
</ns2:updatedFields>
}
```

内部ユーザの削除

この API コールを使用して、Cisco ISE から内部ユーザを削除できます。次の表に、この API コールの主な特性を示します。

表 7-7　Delete Internal Users API コールの主な特性

説明	指定された内部ユーザを削除します
概要	DELETE /ers/config/internaluser/{internaluser-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	204、400、401、403、404、415、429、500

Delete Internal Users API 要求の例

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/internaluser/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.internaluser.1.0+xml
```

Delete Internal Users API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

エンドポイントの外部 RESTful サービス API

次の表に、エンドポイントの外部 RESTful サービス API を示します。

表 7-8　　エンドポイントに使用できる外部 RESTful サービス API

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのエンドポイントの取得	GET	/ers/config/endpoint	適用対象外	page、size、sortacs または sortdsn、filter
エンドポイントの取得	GET	/ers/config/endpoint/{id[1] }	適用対象外	
エンドポイントの作成	POST	/ers/config/endpoint/	endpoint	
エンドポイントの更新	PUT	/ers/config/endpoint/{id}	endpoint	
エンドポイントの削除	DELETE	/ers/config/endpoint/{id}	適用対象外	
エンドポイントの登録	PUT[2]	/ers/config/endpoint/register	endpoint	
エンドポイントの登録解除	PUT	/ers/config/endpoint/{id}/deregister	適用対象外	
エンドポイント リソースのバージョン情報の取得	GET	/ers/config/endpoint/version	適用対象外	

1. エンドポイント ID は、Cisco ISE データベースに保存された UUID タイプです。
2. エンドポイントがすでに存在する場合は、登録されます。存在しない場合は、作成後に登録されます。どちらの場合でも、リターン ステータスは 204 になります。

すべてのエンドポイントの取得

エンドポイントの Get All API は、フィルタに指定されたユーザに関連付けられたエンドポイントの取得にのみ機能します。次の表に、この API コールの主な特性を示します。

表 7-9　Get All Endpoints API コールの主な特性

説明	指定された内部ユーザに関連付けられたエンドポイントの集合を取得します
概要	GET /ers/config/endpoint/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、429、500

Get All Endpoints API 要求の例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint?filter=userid.EQ.123
Authorization: Basic xxxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Get All Endpoints API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
<resources>
<resource name=name1" id="id1">
   <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/id1" rel="self"/>
  </resource>
  <resource name="name2" id="id2">
   <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/id2" rel="self"/>
  </resource>
</resources>
</ns2:searchResult>
}
```

ID によるエンドポイントの取得

この API コールを使用して、Cisco ISE で ID によってエンドポイントを取得できます。次の表に、この API コールの主な特性を示します。

表 7-10　Read Endpoints API コールの主な特性

説明	指定されたエンドポイントを取得します
概要	GET /ers/config/endpoint/{endpoint-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Read Endpoints API 要求の例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Read Endpoints API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
    <group>group</group>
    <groupId>groupId</groupId>
    <identityStore>identityStore</identityStore>
    <identityStoreId>identityStoreId</identityStoreId>
    <mac>00:01:02:03:04:05</mac>
    <portalUser>portalUser</portalUser>
    <profile>profile</profile>
    <profileId>profileId</profileId>
    <staticGroupAssignment>true</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
}
```

エンドポイントの作成

この API コールを使用して、Cisco ISE でエンドポイントを作成できます。次の表に、この API コールの主な特性を示します。

表 7-11　Create Endpoints API コールの主な特性

説明	指定されたエンドポイントを作成します
概要	POST /ers/config/endpoint/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Create Endpoints API 要求の例

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
    <group>group</group>
    <groupId>groupId</groupId>
    <identityStore>identityStore</identityStore>
    <identityStoreId>identityStoreId</identityStoreId>
    <mac>00:01:02:03:04:05</mac>
    <portalUser>portalUser</portalUser>
    <profile>profile</profile>
    <profileId>profileId</profileId>
    <staticGroupAssignment>true</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
}
```

Create Endpoints API 応答の例

```
HTTP/1.1 201 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
Location: https://cisco.com/ers/config/endpoint/444
```

エンドポイントの更新

この API コールを使用して、Cisco ISE でエンドポイントを更新できます。次の表に、この API コールの主な特性を示します。

表 7-12　Update Endpoints API コールの主な特性

説明	指定されたエンドポイントを更新します
概要	PUT /ers/config/endpoint/{endpoint-id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Update Endpoints API 要求の例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:endpoint xmlns:ns2="identity.ers.ise.cisco.com" description="updated"
name="Endpoint">
    <mac>AA:AA:AA:AA:AA:AA</mac>
    <staticGroupAssignment>false</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
</ns2:endpoint>}
```

Update Endpoints API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
<updatedField field="mac">
  <newValue>AA:AA:AA:AA:AA:AA</newValue>
  <oldValue>BB:BB:BB:BB:BB:BB</oldValue>
 </updatedField>
 <updatedField field="staticGroupAssignment">
  <newValue>false</newValue>
  <oldValue>true</oldValue>
 </updatedField>
<updatedField field="staticProfileAssignment">
  <newValue>false</newValue>
  <oldValue>true</oldValue>
 </updatedField>
</ns2:updatedFields>
}
```

エンドポイントの削除

この API コールを使用して、Cisco ISE でエンドポイントを削除できます。次の表に、この API コールの主な特性を示します。

表 7-13　Delete Endpoints API コールの主な特性

説明	指定されたエンドポイントを削除します
概要	DELETE /ers/config/endpoint/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Delete Endpoints API 要求の例

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Delete Endpoints API 応答の例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

エンドポイントの登録

この API コールを使用して、Cisco ISE にエンドポイントを登録できます。エンドポイントが存在しない場合は作成されます。GUI 登録フローと同様に、エンドポイントは登録済みデバイスグループに静的に割り当てられ、ポータル ユーザおよび ID ストアはコンテンツで指定されているとおりに設定されます。

次の表に、この API コールの主な特性を示します。

表 7-14　Register Endpoints API コールの主な特性

説明	指定されたエンドポイントを登録します
概要	PUT /ers/config/endpoint/register
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	endpoint
応答ヘッダー	Content-Length、Content-Type

表 7-14　Register Endpoints API コールの主な特性（続き）

応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	202、400、401、403、404、415、429、500

Register Endpoints API 要求の例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/register
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpoint description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="identity.ers.ise.cisco.com">
    <group>group</group>
    <groupId>groupId</groupId>
    <identityStore>identityStore</identityStore>
    <identityStoreId>identityStoreId</identityStoreId>
    <mac>00:01:02:03:04:05</mac>
    <portalUser>portalUser</portalUser>
    <profile>profile</profile>
    <profileId>profileId</profileId>
    <staticGroupAssignment>true</staticGroupAssignment>
    <staticProfileAssignment>false</staticProfileAssignment>
</ns3:endpoint>
}
```

Register Endpoints API 応答の例

```
HTTP/1.1 204 No Content
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

エンドポイントの登録解除

この API コールを使用して、Cisco ISE でエンドポイントを登録解除できます。結果は何も示されません。次の表に、この API コールの主な特性を示します。

表 7-15　Deregister Endpoints API コールの主な特性

説明	指定されたエンドポイントを登録解除します
概要	PUT /ers/config/endpoint/{id}/deregister
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答メッセージ本文	該当なし
応答ステータス	202、400、401、403、404、415、429、500

Deregister Endpoints API コールの要求例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/123/deregister
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpoint.1.0+xml
```

Deregister Endpoints API コールの応答例

```
HTTP/1.1 204 No Content
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

エンドポイントのバルク実行の開始

バルク実行によって、1 つの要求で同じタイプの CRUD 操作を最大 500 送信できます。

要求が有効な場合、サーバはステータス コード `202 (ACCEPTED)` と LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID によって、バルク ステータスの取得操作を使用する際にバルク ステータスを追跡できます。

一度に可能なバルクは 1 つのみです。別のバルク実行の実行中にバルク要求がポストされると、サーバは応答ステータス `503 (Service Unavailable)` とともに、クライアントに後で再試行するように促すメッセージを返します。

表 7-16　Start Bulk Execution for Endpoints の主な特性

説明	実行を開始します
概要	PUT /ers/config/endpoint/bulk
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	BulkRequest
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	適用対象外
応答ステータス	202、400、401、403、404、415、500

Start Bulk Execution for Endpoints API コールの要求例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/bulk HTTP/1.1
Host: {some-ise-node-ip}
Authorization: Basic xxxxxxxxxxxxxxxxxx Content-Type:
application/vnd.com.cisco.ise.ers.bulkrequest.1.0+xml
{
<ns3:endpointBulkRequest
resourceMediaType = "vnd.com.cisco.ise.ers.identity.endpoint.1.0+xml" operationType =
"create"
xmlns:ns2 = "ers.ise.cisco.com"
xmlns:ns3 = "identity.ers.ise.cisco.com">
<resourcesList> <resource
xsi:type = "ns3:ersEndPoint"
description = "created by bulk request"
```

```
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"> <mac>11:22:33:44:55:66</mac>
<staticGroupAssignment>false</staticGroupAssignment>
<staticProfileAssignment>false</staticProfileAssignment>
</resource>
...
<resource
xsi:type = "ns3:ersEndPoint"
description = "created by bulk request"
xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"> <group>Profiled</group>
<groupId>804f5350-7808-11e2-bdd0-0050568e01f0</groupId> <identityStore></identityStore>
<identityStoreId></identityStoreId> <mac>11:22:33:44:55:77</mac>
<portalUser></portalUser>
<profile>Apple-iPod</profile> <profileId>b8128870-7808-11e2-bdd0-0050568e01f0</profileId>
<staticGroupAssignment>true</staticGroupAssignment>
<staticProfileAssignment>true</staticProfileAssignment>
</resource> </resourcesList>
</ns3:endpointBulkRequest>
}
```

Start Bulk Execution for Endpoints API コールの応答例

```
HTTP/1.1 202 ACCEPTED
        Date: Thu, 12 Jul 2012 23:59:59 GMT
        Location: https://ise-node-ip:9060/ers/config/endpoint/123443545334
```

エンドポイントのバルク ステータスの取得

バルク実行要求が有効で、その他のバルクが進行中でない場合、サーバは LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID を使用してバルク ステータスを追跡します。ステータス レポートは、操作の開始時刻から 2 時間以上経ってから使用できます。

表 7-17　Get Bulk Status の主な特性

説明	指定されたバルク実行の進捗を監視します
概要	GET /ers/config/endpoint/bulk/{bulkid}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	BulkStatus
応答ステータス	200、400、401、403、404、415、500

エンドポイントのバルク ステータスの取得の例

要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/bulk/53454354534 HTTP/1.1
    Host: {some-ise-node-ip}
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.bulkStatus.1.0+xml
```

応答

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.bulkStatus.1.0+xml Content-Length: 16347
{
<ns2:bulkStatus
           xmlns:ns2 = "ers.ise.cisco.com"
           successCount = "750"
           startTime = "Thu Mar 07 17:17:35 IST 2013"
           resourcesCount = "750"
           operationType = "create"
           mediaType =
    "vnd.com.cisco.ise.ers.identity.endpoint.1.0+xml"
           failCount = "0"
           executionStatus = "COMPLETED"
           bulkId = "1362669455284">
           <resourcesStatus>
                  <resourceStatus status = "SUCCUESS"
                         description = "created by bulk request"
                         id = "23d068d0-873a-11e2-bad4-00215edbb2a8" />
....
                  <resourceStatus
                         status = "SUCCUESS"
                         description = "created by bulk request"
                         id = "23cfa580-873a-11e2-bad4-00215edbb2a8"/>
           </resourcesStatus>
    </ns2:bulkStatus>
}
}
```

エンドポイント ID グループの外部 RESTful API

次の表に、エンドポイント ID グループの外部 RESTful サービス API を示します。

表 7-18　エンドポイント ID グループに使用できる ERS API

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのエンドポイント グループの取得	GET	/ers/config/endpointgroup	適用対象外	page、size、sortacs または sortdsn、filter
エンドポイント グループの取得	GET	/ers/config/endpointgroup/{id[1] }	適用対象外	
エンドポイント グループの作成	POST	/ers/config/endpointgroup/	Endpointgroup	

表 7-18　エンドポイント ID グループに使用できる ERS API（続き）

オペレーション	メソッド	URL	内容	クエリー文字列
エンドポイント グループの更新	PUT	/ers/config/endpointgroup/{id}	Endpointgroup	
エンドポイント グループの削除	DELETE	/ers/config/endpointgroup/{id}	適用対象外	
IdentityGroup リソースのバージョン情報の取得	GET	/ers/config/ endpointgroup /version	適用対象外	

1. エンドポイント グループ ID は、Cisco ISE データベースに保存された UUID タイプです。

すべてのエンドポイント ID グループの取得

次の表に、Get All Endpoint Identity Groups API コールの主な特性を示します。

表 7-19　Get All Endpoint Identity Groups API コールの取得の主な特性

説明	エンドポイント グループの集合を取得します
概要	GET /ers/config/endpoint
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortasc、sortdsc、filter
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	202、400、401、403、404、415、429、500

Get All Endpoint Identity Groups API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
```

Get All Endpoint Identity Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
<resources>
  <resource name="name1" id="id1" description="description1">
   <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup/id1" rel="self"/>
  </resource>
```

```
  <resource name="name2" id="id2" description="description2">
   <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/endpointgroup/id2" rel="self"/>
  </resource>
</resources>
</ns2:searchResult>
}
```

ID によるエンドポイント ID グループの取得

次の表に、Get Endpoint Identity Groups by ID API コールの主な特性を示します。

表 7-20　Read Endpoint Identity Groups API コールの取得の主な特性

説明	指定されたエンドポイント グループを取得します
概要	GET /ers/config/endpoint/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	Endpoint タイプのリソース
応答ステータス	200、400、401,403、404、415、429、500

Read Endpoint Identity Groups API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
```

Read Endpoint Identity Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <systemDefined>true</systemDefined>
</ns3:endpointgroup>
}
```

エンドポイント ID グループの作成

次の表に、Create Endpoint Identity Groups API コールの主な特性を示します。

:

表 7-21　Create Endpoint Identity Groups API コールの取得の主な特性

説明	指定されたエンドポイント グループを作成します
概要	POST /ers/config/endpoint/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	エンドポイント グループ
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	エンドポイント グループ
応答ステータス	201、400、401、403、415、429、500

Create Endpoint Identity Groups API コールの要求例

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:endpointgroup description="description" name="name" id="id"
xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com">
    <systemDefined>true</systemDefined>
</ns3:endpointgroup>
}
```

Create Endpoint Identity Groups API コールの応答例

```
HTTP/1.1 201 OK  (see the location header for the new endpoint ID)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type:
Location: https://cisco.com/ers/config/endpointgroup/444
```

エンドポイント ID グループの更新

次の表に、Update Endpoint Identity Groups API コールの主な特性を示します。

表 7-22　Update Endpoint Identity Groups API コールの取得の主な特性

説明	指定されたエンドポイント グループを更新します
概要	PUT /ers/config/endpoint/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	エンドポイント グループ
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、429、500

Update Endpoint Identity Groups API コールの要求例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/ endpoint /333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:endpointgroup xmlns:ns2="identity.ers.ise.cisco.com" description="updated" id="0"
name="Group">
    <systemDefined>false</systemDefined>
</ns2:endpointgroup>
}
```

Update Endpoint Identity Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml
Content-Length: 529
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
 <updatedField field="description">
  <newValue>updated</newValue>
  <oldValue>Group</oldValue>
 </updatedField>
</ns2:updatedFields>
}
```

エンドポイント ID グループの削除

次の表に、Delete Endpoint Identity API コールの主な特性を示します。

表 7-23　Delete Endpoint Identity Groups API コールの主な特性

説明	指定されたエンドポイント グループを削除します
概要	DELETE /ers/config/endpointgroup/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	NA
応答ステータス	204、400、401、403、404、415、429、500

Delete Endpoint Identity Groups API コールの要求例

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/endpoint/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.endpointgroup.1.0+xml
```

Delete Endpoint Identity Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

ID グループの外部 RESTful サービス API

次の表に、ID グループの外部 RESTful サービス API を示します。

表 7-24　ID グループに使用できる API

オペレーション	メソッド	URL	内容	クエリー文字列
すべての ID グループの取得	GET	/ers/config/identitygroup	適用対象外	page、size、sortacs または sortdsn、filter
IdentityGroup リソースのバージョン情報の取得	GET	/ers/config/identitygroup /version	適用対象外	

すべての ID グループの取得

この API コールを使用して、Cisco ISE のすべての ID グループを取得できます。次の表に、この API コールの主な特性を示します。

表 7-25　Retrieve All Identity Groups API コールの取得の主な特性

説明	ID グループ リソースの集合を取得します
概要	GET /ers/config/identitygroup
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、429、500

Retrieve All Identity Groups API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.identitygroup.1.0+xml
```

Retrieve All Identity Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
    <resources>
      <resource name="name1" id="id1" description="description1">
        <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup/id1" rel="self"/>
      </resource>
      <resource name="name2" id="id2" description="description2">
        <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/identitygroup/id2" rel="self"/>
      </resource>
    </resources>
</ns2:searchResult>
}
```

ゲスト ユーザの外部 RESTful サービス API

次の表に、ゲスト ユーザの外部 RESTful サービス API を示します。

表 7-26　　サポートされているシナリオ

オペレーション	メソッド	URL	内容
特定のゲスト ユーザの取得	GET	/ers/config/guestuser/{id}	適用対象外
すべてのゲスト ユーザの取得	GET	/ers/config/guestuser/	適用対象外
ゲスト ユーザの作成	POST	/ers/config/guestuser /	ゲスト ユーザ情報（XML）
ゲスト ユーザの更新	PUT	/ers/config/guestuser/{id}	部分的または完全なゲスト ユーザ情報（XML）
ゲスト ユーザの削除	DELETE	/ers/config/guestuser/{id}	適用対象外
ゲスト ユーザの一時停止	PUT	/ers/config/guestuser/suspend/{id}	理由
ゲスト ユーザの復元	PUT	/ers/config/guestuser/reinstate/{id}	適用対象外
電子メールの送信	PUT	/ers/config/guestuser/email/{id}/ portalId/{portalId}	senderEmail
SMS の送信	PUT	/ers/config/guestuser/sms/{id}/portalId /{portalId}	適用対象外
ゲスト ユーザの承認	PUT	/ers/config/guestuser/approve/{id}	適用対象外
ゲスト ユーザの拒否	PUT	/ers/config/guestuser/deny/{id}	適用対象外
バルク実行の開始	PUT	/ers/config/ guestuser/bulk	BulkRequest
バルク ステータスの取得	GET	/ers/config/ guestuser/bulk/{bulkId}	適用対象外
スポンサーのパスワードの変更	PUT	/ers/config/guestuser/changeSponsor Password/{portalId}	operationAdd itionalData
すべてのポータルの取得	GET	/ers/config/portal	適用対象外
ID によるポータルの取得	GET	/ers/config/portal/{id}	適用対象外
ゲスト API 情報の取得	GET	/ers/config/guestuser/versioninfo	適用対象外

Content-Type ヘッダーおよび Accept ヘッダー

各ゲスト アカウント要求は次のように設定する必要があります。

- Content Type：`application/vnd.com.cisco.ise.identity.guestuser.2.0+xml`
- Accept：`application/vnd.com.cisco.ise.identity.guestuser.2.0+xml`

各バルク実行要求は次のように設定する必要があります。

- Content Type：`application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`
- Accept：`application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml`

ゲスト ユーザの取得

GET 操作を使用して、ゲストのユーザ名またはデータベース レコード ID のいずれかによって ISE データベースから特定のゲスト ユーザを取得できます。

表 7-27　Get a Guest User の主な特性

説明	指定されたゲスト ユーザを取得します
概要	GET /ers/config/guestuser/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	GuestUser タイプのリソース
応答ステータス	200、400、401、403、404、415、500

ゲスト ユーザの取得の例

ID によるゲスト ユーザの取得の例

要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3333


Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxx
```

応答

```
<?xml version="1.0" encoding="UTF-8"?>
<ns3:guestuser xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"
name="fzvhervocnz" id="af827350-1f0f-11e4-b961-005056103001">
 <link type="application/xml" href="https://10.0.10.130:9060/ers/config/guestuser/3333"
rel="self"/>
 <customFields/>
 <guestAccessInfo>
  <fromDate>14/08/08 08:21</fromDate>
  <toDate>14/08/09 08:21</toDate>
  <validDays>1</validDays>
 </guestAccessInfo>
 <guestInfo>
  <company>Cisco</company>
  <creationTime>14/08/08 08:21</creationTime>
  <emailAddress>doe@example.com</emailAddress>
  <enabled>true</enabled>
```

```
  <firstName>John</firstName>
  <lastName>Doe</lastName>
  <notificationLanguage>English</notificationLanguage>
  <password>12345</password>
  <phoneNumber>9999998877</phoneNumber>
  <smsServiceProvider>ATT</smsServiceProvider>
  <userName>Guest1</userName>
 </guestInfo>
 <guestType>Daily (default)</guestType>
 <personBeingVisited>sponsor@example.com</personBeingVisited>
 <reasonForVisit>Interview</reasonForVisit>
 <sponsorUserName>SponsoredUser1</sponsorUserName>
 <status>Awaiting Initial Login</status>
</ns3:guestuser>
```

関連項目

API でのパスワード可視性の詳細については、「ゲスト パスワード」(P.2-8) を参照してください。

「ilucky」で始まるユーザ名によるフィルタリングの例

要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?filter=userName.STARTSW.ilucky
```

応答

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
    xmlns:ns2="ers.ise.cisco.com" total="8">
    <resources>
        <resource name="ilucky101" id="a0957160-6224-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/a0957160-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky102" id="e14f4460-6224-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/e14f4460-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky201" id="123581f0-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/123581f0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky301" id="154f9330-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/154f9330-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky401" id="172c6980-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/172c6980-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky501" id="19631fa0-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/19631fa0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky601" id="1b44b0e0-6227-11e3-9bc2-000c2932c73c">
```

```
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/1b44b0e0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky602" id="2e1ac600-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/2e1ac600-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
    </resources>
</ns2:searchResult>
```

「ilucky」で始まるユーザ名および「J」で始まる姓によるフィルタリングの例

要求

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?filter=userName.STARTSW.ilucky&filter=
lastName.STARTSW.j
```

応答

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
    xmlns:ns2="ers.ise.cisco.com" total="8">
    <resources>
        <resource name="ilucky101" id="a0957160-6224-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/a0957160-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky102" id="e14f4460-6224-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/e14f4460-6224-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky201" id="123581f0-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/123581f0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky301" id="154f9330-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/154f9330-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky401" id="172c6980-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/172c6980-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky501" id="19631fa0-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/19631fa0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky601" id="1b44b0e0-6227-11e3-9bc2-000c2932c73c">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/1b44b0e0-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
        <resource name="ilucky602" id="2e1ac600-6227-11e3-9bc2-000c2932c73c">
```

```
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/2e1ac600-6227-11e3-9bc2-000c2932c73
c" rel="self"/>
        </resource>
    </resources>
</ns2:searchResult>
```

名「John」によるフィルタリングおよびユーザ名によるソートの例

要求

```
GET
https://<ISE-Admin-node>:9060/ers/config/guestuser/?page=0&size=10&sortdsc=name&filter=fir
stName.eq.john
```

応答

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult
    xmlns:ns2="ers.ise.cisco.com" total="2">
    <resources>
        <resource name="jdoe0002" id="886f5b40-5ece-11e3-8faf-000c29c56fc6">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/886f5b40-5ece-11e3-8faf-000c29c56fc
6" rel="self"/>
        </resource>
        <resource name="jdoe0001" id="79e5a5a0-5df9-11e3-84f5-000c29c56fc6">
            <link type="application/xml"
href="https://ISE-ADMIN-NODE:9060/ers/config/guestuser/79e5a5a0-5df9-11e3-84f5-000c29c56fc
6" rel="self"/>
        </resource>
    </resources>
</ns2:searchResult>
```

curl を使用したゲスト ユーザの要求例および応答例

次に、ISE に送信された ID によるゲスト ユーザの取得要求、および curl Linux コマンドを使用した応答の例を示します。

curl コマンド

```
$ curl -v -k -H 'ACCEPT:application/vnd.com.cisco.ise.identity.guestuser.2.0+xml'
https://username:password@<ISE-ADMIN-NODE>:9060/ers/config/guestuser/user1
* About to connect() to <ISE-ADMIN-NODE> port 9060
*   Trying 111.11.11.111... * connected
* Connected to <ISE-ADMIN-NODE> (<ISE-ADMin-NODE-IP>) port 9060
* successfully set certificate verify locations:
*   CAfile: /usr/share/ssl/certs/ca-bundle.crt
  CApath: none
* SSL connection using DHE-RSA-AES256-SHA
* Server certificate:
*        subject: /CN=<ISE-ADMIN-NODE>
*        start date: 2013-11-26 00:56:55 GMT
*        expire date: 2014-11-26 00:56:55 GMT
*        common name: <ISE-ADMIN-NODE>
*        issuer: /CN=<ISE-ADMIN-NODE>
* Server auth using Basic with user 'username'
```

ID によるゲスト ユーザの GET 要求

```
> GET /ers/config/guestuser/444
Authorization: Basic xxxxxxxxxxxxxxxx
User-Agent: curl/7.12.1 (i386-redhat-linux-gnu) libcurl/7.12.1 OpenSSL/0.9.7a zlib/1.2.1.2
libidn/0.5.6
Host: <ISE-ADMIN-NODE>:9060
Pragma: no-cache
ACCEPT:application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

応答

```
< HTTP/1.1 200 OK
< Pragma: No-cache
< Cache-Control: no-cache
< Expires: Wed, 31 Dec 1969 16:00:00 PST
< Set-Cookie: JSESSIONIDSSO=0FCBC2621A0897193FE3105B3FBA8F16; Path=/; Secure
< Set-Cookie: JSESSIONID=5B6092B3FCCE047F7282C52592FAFC7A; Path=/ers; Secure
< Date: Thu, 02 Jan 2014 23:01:59 GMT
< Content-Type: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
< Content-Length: 1162
< Server:
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:guestuser xmlns:ns2="ers.ise.cisco.com" xmlns:ns3="identity.ers.ise.cisco.com"
name="user1" id="b4bdf2b0-73e1-11e3-8cdf-000c29c56fc6">
<link type="application/xml" href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/444"
rel="self"/>
<guestAccessInfo>
<fromDate>14/08/06 23:26</fromDate>
<toDate>14/08/07 23:26</toDate>
<validDays>1</validDays>
</guestAccessInfo>
<guestInfo>
<company>New Company</company>
<emailAddress>john@example.com</emailAddress>
<firstName>John</firstName>
<lastName>Doe</lastName>
<notificationLanguage>English</notificationLanguage>
<phoneNumber>9999998877</phoneNumber>
<smsServiceProvider>Global Default</smsServiceProvider>
<userName>user1</userName>
</guestInfo>
<guestType>Daily (default)</guestType>
<personBeingVisited>sponsor@example.com</personBeingVisited>
<portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
<reasonForVisit>Interview</reasonForVisit>
</ns3:guestuser>
```

関連項目

API でのパスワード可視性の詳細については、「ゲスト パスワード」(P.6-6) を参照してください。

すべてのゲスト ユーザの取得

この GET 操作を使用して、ISE データベースのすべてのゲスト ユーザを取得し、その結果を名前、ユーザ名、電子メール アドレスなどの条件に基づいてフィルタリングすることができます。応答には、ゲストのユーザ名、ID、および完全な表現へのリンクが含まれています。

表 7-28　Get All Guest Users の主な特性

説明	ゲスト ユーザのコレクションを取得します
概要	GET /ers/config/guestuser/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortasc、sortdsc、filter
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

関連項目

「フィルタリング パラメータ」（P.2-9）

すべて取得の例

次の例では、GET 操作で、ユーザ名が `ilu` から始まり、姓が `b` から始まるすべてのゲスト ユーザを取得します。

要求

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/?page=0&size=10&sortasc=name&filter=nam
e.STARTSW.ilu&filter=firstName.STARTSW.b

Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxx
```

応答

```
HTTP/1.1 200 OK;
Date:Sat, 15 Dec 2012 21:55:05 GMT;
Content-Length:1439;
Content-Type:application/vnd.com.cisco.ise.ers.searchresult.1.0+xml;

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult xmlns:ns2="ers.ise.cisco.com" total="6">
  <resources>
    <ns2:resource name="ilucky01" id="61dc9060-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/61dc9060-46a1-11e2-b141-000c290fc
f9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky02" id="3f43bb40-468e-11e2-8f92-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3f43bb40-468e-11e2-8f92-000c290fc
f9a" rel="self"/>
```

```
    </ns2:resource>
    <ns2:resource name="ilucky03" id="6c65d6d0-468e-11e2-8f92-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/6c65d6d0-468e-11e2-8f92-000c290fc
f9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky04" id="6948bdb0-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/6948bdb0-46a1-11e2-b141-000c290fc
f9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky05" id="abbb6440-46a1-11e2-b141-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/abbb6440-46a1-11e2-b141-000c290fc
f9a" rel="self"/>
    </ns2:resource>
    <ns2:resource name="ilucky06" id="4d9a1530-46fd-11e2-b70b-000c290fcf9a">
      <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/4d9a1530-46fd-11e2-b70b-000c290fc
f9a" rel="self"/>
    </ns2:resource>
  </resources>
</ns2:searchResult>
```

ゲスト ユーザの作成

POST 操作を使用して、ゲストにゲスト フローからのログインを許可する新しいゲスト ユーザ アカウントを作成できます。

guestType は、ゲスト ユーザ アカウントの作成に必要です。

表 7-29　　Create a Guest User の主な特性

説明	指定された内部ユーザを作成します
概要	POST /ers/config/guestuser/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	GuestUser
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	GuestUser タイプのリソース
応答ステータス	201、400、401、403、415、500

ゲスト ユーザの作成の例

要求

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/

Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxx

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
    <guestAccessInfo>
```

```
        <fromDate>14/08/08 08:15</fromDate>
        <toDate>14/08/09 08:15</toDate>
        <validDays>1</validDays>
    </guestAccessInfo>
    <guestInfo>
        <company>New Company</company>
        <emailAddress>doe@example.com</emailAddress>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <notificationLanguage>English</notificationLanguage>
        <phoneNumber>9999998877</phoneNumber>
        <smsServiceProvider>Global Default</smsServiceProvider>
        <userName>guestuser1</userName>
    </guestInfo>
    <guestType>Daily (default)</guestType>
    <personBeingVisited>sponsor@example.com</personBeingVisited>
    <portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
    <reasonForVisit>Interview</reasonForVisit>
</ns2:guestuser>
```

応答

```
HTTP/1.1 201 Created;
Date:Sat, 15 Dec 2012 21:20:51 GMT;
Content-Length:0;
Location:https://<ISE-ADMIN-NODE>/ers/config/guestuser/e1bb8290-6ccb-11e3-8cdf-000c29c56fc
6;
Set-Cookie:JSESSIONID=28CF43F1ACCC7448BED7255DC7B787EE; Path=/ers;
Secure;JSESSIONIDSSO=DB6D6900088D1863CA84863570392E4C; Path=/; Secure;
Content-Type:application/xml;
```

関連項目

API でのパスワード可視性の詳細については、「ゲスト パスワード」(P.2-8)を参照してください。

ゲスト ユーザの更新

PUT 操作を使用してリソースを更新すると、既存のゲスト ユーザの属性を変更できます。ゲスト ユーザ属性について、フル更新または部分更新を実行できます。

表 7-30　Update a Guest User の主な特性

説明	指定されたゲスト ユーザを更新します
概要	PUT /ers/config/guestuser/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	GuestUser
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、404、415、500

ゲスト ユーザの更新の例

ユーザ更新の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/name/ilucky101

Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxx

<?xml version="1.0" encoding="UTF-8"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
    <portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
    <reasonForVisit>Interview</reasonForVisit>
</ns2:guestuser>
```

応答

```
Status:200 OK

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
 <updatedField field="ReasonForVisit">
  <newValue>Interview</newValue>
  <oldValue>no reason</oldValue>
 </updatedField>
 <updatedField field="validDays">
  <newValue>0</newValue>
  <oldValue>1</oldValue>
 </updatedField>
</ns2:updatedFields>
```

ユーザ パスワードのリセットの例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/e1bb8290-6ccb-11e3-8cdf-000c29c56f
c6
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxx

<?xml version="1.0" encoding="UTF-8"?>
<ns2:guestuser xmlns:ns2="identity.ers.ise.cisco.com">
    <guestInfo>
        <password/>
    </guestInfo>
    <portalId>ff2d99e0-2101-11e4-b5cf-005056bf2f0a</portalId>
</ns2:guestuser>
```

応答

```
Status:200 OK
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:updatedFields xmlns:ns2="ers.ise.cisco.com">
 <updatedField field="validDays">
```

```
  <newValue>0</newValue>
  <oldValue>1</oldValue>
 </updatedField>
 <updatedField field="GuestInfo.Password">
  <newValue>;AuJ5h:0</newValue>
 </updatedField>
</ns2:updatedFields>
```

関連項目

API でのパスワード可視性の詳細については、「ゲスト パスワード」(P.2-8) を参照してください。

ゲスト ユーザの削除

データベース レコード ID を使用して、ISE データベースからゲスト ユーザのレコードを削除できます。ユーザは、次回のログイン時にログインできなくなります。

表 7-31　Delete a Guest User の主な特性

説明	指定されたゲスト ユーザを削除します
概要	DELETE /ers/config/guestuser/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	適用対象外
応答ステータス	204、400、401、403、404、415、500

ゲスト ユーザの削除の例

要求

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/3333

Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Accept - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Authorization - Basic xxxxxxxxxxxxxxxxxxx
```

応答

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

ゲスト ユーザの一時停止

PUT 操作を使用して、特定のゲスト ユーザを一時停止します。ユーザは、次回のログイン時にログインできなくなります。一時停止の理由を含める必要があります。理由にはスペースを含めることができます。

表 7-32　Suspend a Guest User の主な特性

説明	指定されたゲスト ユーザを一時停止します
概要	PUT /ers/config/guestuser/suspend/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	理由
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	GuestUser タイプのリソース
応答ステータス	204、400、401、403、404、415、500

ID によるゲスト ユーザの一時停止の例

要求

```
PUT https:/<ISE-ADMIN-NODE>:9060/ers/config/guestuser/suspend/3333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
Content-Type - application/vnd.com.cisco.ise.identity.guestuser.2.0+xml

<?xml version="1.0" encoding="UTF-8"?>
<ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
    <requestAdditionalAttributes>
        <additionalAttribute name="reason" value="AUP not accepted"/>
    </requestAdditionalAttributes>
</ns3:operationAdditionalData>
```

応答

```
HTTP/1.1 204 No Content
Sat, 15 Dec 2012 10:14:38 GMT
```

ゲスト ユーザの復元

PUT 操作を使用して、一時停止したゲスト ユーザ アカウントを復元します。

表 7-33　Reinstate a Guest User の主な特性

説明	指定されたゲスト ユーザを復元します
概要	PUT /ers/config/guestuser/reinstate/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外

表 7-33　Reinstate a Guest User の主な特性

応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	GuestUser タイプのリソース
応答ステータス	204、400、401、403、404、415、500

ゲスト ユーザの復元の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/reinstate/33
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ゲスト ユーザへの電子メール送信

PUT 操作を使用して、ゲスト ユーザの電子メール アカウントに電子メールを送信します。Cisco ISE で SMTP サーバを設定する必要があります。

ポータル構成には電子メールの本文と件名に必要な情報が含まれるため、要求にはポータル ID が必要です。

表 7-34　Send an Email to a Guest User の主な特性

説明	指定されたゲスト ユーザに電子メールを送信します
概要	PUT /ers/config/guestuser/email/{id}/portalId/{portalID}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	senderEmail
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	適用対象外
応答ステータス	204、400、401、403、404、415、500

ゲスト ユーザへの電子メール送信の例

要求

```
PUT
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/email/4444/portalId/ff2d99e0-2101-11e4-b5cf-005056bf2f0a
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept:
    application/vnd.com.cisco.ise.identity.guestuser.2.0+xml

<?xml version="1.0" encoding="UTF-8"?>
```

```
 <ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
       <requestAdditionalAttributes>
           <additionalAttribute name="senderEmail" value="sender Email"/>
       </requestAdditionalAttributes>
</ns3:operationAdditionalData>
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ゲスト ユーザへの SMS テキスト送信

PUT 操作を使用して、ゲスト ユーザの携帯電話にテキスト メッセージを送信します。Cisco ISE で SMTP サーバを設定する必要があります。

ポータル構成には本文に必要な情報が含まれるため、要求にはポータル ID が必要です。

表 7-35　Send an Email to a Guest User の主な特性

説明	指定されたゲスト ユーザに SMS を送信します
概要	PUT /ers/config/guestuser/sms/{id}/portalId/{portalID}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	適用対象外
応答ステータス	204、400、401、403、404、415、500

SMS の送信の例

要求

```
PUT
https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/sms/444/portalId/ff2d99e0-2101-11e4-b5c
f-005056bf2f0a
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept:
    application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ゲスト ユーザの承認

この操作を使用して、ゲスト ユーザ アカウントを承認できます。ゲスト アカウント ID を使用する必要があります。

表 7-36　Get API Version の主な特性

説明	指定されたゲスト ユーザを承認します
概要	PUT /ers/config/guestuser/approve/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	API バージョン
応答ステータス	200、400、401、403、404、415、500

ゲスト ユーザ承認の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/approve/3333
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ゲスト ユーザ アカウントの承認拒否

この操作を使用して、ゲスト ユーザ アカウントの承認を拒否できます。ゲスト アカウント ID を使用する必要があります。

表 7-37　Get API Version の主な特性

説明	指定されたゲスト ユーザを拒否します
概要	PUT /ers/config/guestuser/deny/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	API バージョン
応答ステータス	200、400、401、403、404、415、500

ゲスト ユーザの承認拒否の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/deny/7777
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ゲスト ユーザのバルク実行の開始

バルク要求を使用すると、1 つの要求で最大 500 の操作、または ID に基づいて 5000 の操作を送信できます。

要求が有効な場合、サーバはステータス コード `202 (ACCEPTED)` と LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID によって、バルク ステータスの取得操作を使用する際にバルク ステータスを追跡できます。

一度に可能なバルクは 1 つのみです。別のバルク実行の実行中にバルク要求がポストされると、サーバは応答ステータス `503 (Service Unavailable)` とともに、クライアントに後で再試行するように促すメッセージを返します。

表 7-38　Start Bulk Execution の主な特性

説明	実行を開始します
概要	PUT /ers/config/guestuser/bulk
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	BulkRequest
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	適用対象外
応答ステータス	202、400、401、403、404、415、500

ゲストのバルク作成の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/bulk
Authorization: Basic
Content-Type: application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:bulkRequest xsi:type="ns2:guestUserBulkRequest"
                 resourceMediaType="vnd.com.cisco.ise.identity.guestuser.1.0+xml"
                 operationType="create"
                 xmlns:ns2="identity.ers.ise.cisco.com"
     xmlns:ns3="ers.ise.cisco.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <resourcesList>
  <resource xsi:type="ns2:GuestUser" description="created by bulk">
  <portalId>6ab68890-d0f1-11e3-a1d5-005056bf4687</portalId>
```

```
   <guestAccesstInfo>
        <groupTag>group</groupTag>
        <validDays>2</validDays>
        <location>London</location>
        <ssid>guest_ssid</ssid>
    </guestAccesstInfo>
<guestInfo>
        <company>new company</company>
        <emailAddress>joe@example.com</emailAddress>
        <enabled>true</enabled>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <phoneNumber>6033203311</phoneNumber>
        <userName>lucky7</userName>
        <password>1234</password>
        <notificationLanguage>English</notificationLanguage>
        <smsServiceProvider>ATT</smsServiceProvider>
    </guestInfo>
    <guestType>DAILY</guestType>
    <reasonForVisit>interview</reasonForVisit>
    <personBeingVisited>sponsor@cisco.com</personBeingVisited>
  </resource>

...
  <resource xsi:type="ns2:GuestUser" description="created by bulk">
  <portalId>6ab68890-d0f1-11e3-a1d5-005056bf4687</portalId>
   <guestAccesstInfo>
        <groupTag>group</groupTag>
        <validDays>3</validDays>
        <location>London</location>
        <ssid>guest_ssid</ssid>
    </guestAccesstInfo>
    <guestInfo>
        <company>new company</company>
        <emailAddress>mary@example.com</emailAddress>
        <enabled>true</enabled>
        <firstName>Mary</firstName>
        <lastName>Sue</lastName>
        <phoneNumber>6039990000</phoneNumber>
        <userName>lucky13</userName>
        <password>1234</password>
        <notificationLanguage>English</notificationLanguage>
        <smsServiceProvider>ATT</smsServiceProvider>
    </guestInfo>
    <guestType>DAILY</guestType>
    <reasonForVisit>interview</reasonForVisit>
    <personBeingVisited>sponsor@cisco.com</personBeingVisited>
  </resource>
 </resourcesList>
</ns3:bulkRequest>
```

応答

```
HTTP/1.1 202 ACCEPTED
        Date: Thu, 12 Jul 2012 23:59:59 GMT
        Location: https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/123443545334
```

関連項目

「エンドポイントのバルク ステータスの取得」（P.7-16）

ゲスト ユーザのバルク ステータスの取得

バルク実行要求が有効で、その他のバルクが進行中でない場合、サーバは LOCATION 応答ヘッダーの一意のバルク ID を返します。この ID を使用してバルク ステータスを追跡します。ステータス レポートは、操作の開始時刻から 2 時間以上経ってから使用できます。

表 7-39　Get Bulk Status の主な特性

説明	指定されたバルク実行の進捗を監視します
概要	GET /ers/config/guestuser/bulk/{bulkid}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	BulkStatus
応答ステータス	200、400、401、403、404、415、500

ゲスト ユーザのバルク ステータス取得の例

要求

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/bulk/53454354534 HTTP/1.1
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.guestuserbulkrequest.1.0+xml
```

応答

```
HTTP/1.1 200 OK
Date: Thu Mar 07 18:17:35 IST 2013 GMT
Content-Type: application/vnd.com.cisco.ise.ers.guestuserbulkrequest.1.0+xml
Content-Length: 16347
{
<ns2:bulkStatus
  xmlns:ns2 = "ers.ise.cisco.com"
  successCount = "50"
  startTime = "Thu Mar 07 17:17:35 IST 2013"
  resourcesCount = "50"
  operationType = "create"
  resourceMediaType = "vnd.com.cisco.ise.ers.identity.guestuser.1.0+xml"
  failCount = "0"
  executionStatus = "COMPLETED"
  bulkId = "53454354534">

  <resourcesStatus>
    <resourceStatus
     status = "SUCCUESS"
     description = "created by bulk request"
     id = "23d068d0-873a-11e2-bad4-00215edbb2a8"/>

...

    <resourceStatus
     status = "SUCCUESS"
     description = "created by bulk request"
     id = "23cfa580-873a-11e2-bad4-00215edbb2a8"/>
  </resourcesStatus>
</ns2:bulkStatus>
}
```

スポンサーのパスワードの変更

この操作で、現在ログインしているスポンサーのパスワードを変更できます。ポータル ID を使用する必要があります。

表 7-40　Get API Version の主な特性

説明	ログインしたスポンサーのパスワードを更新します
概要	PUT /ers/config/guestuser/changeSponsorPassword/{portalId}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	適用対象外
要求メッセージ本文	適用対象外
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	API バージョン
応答ステータス	200、400、401、403、404、415、500

スポンサーのパスワード変更の例

要求

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/guestuser/changeSponsorPassword/88888
    Host: cisco.com
    Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.identity.guestuser.2.0+xml

<?xml version="1.0" encoding="UTF-8"?>
<ns3:operationAdditionalData xmlns:ns2="identity.ers.ise.cisco.com"
xmlns:ns3="ers.ise.cisco.com">
    <requestAdditionalAttributes>
        <additionalAttribute name="newPassword" value="Cisco1234"/>
        <additionalAttribute name="currentPassword" value="Autom8me"/>
    </requestAdditionalAttributes>
</ns3:operationAdditionalData>
```

応答

```
HTTP/1.1 204 OK
Date: Sat, 15 Dec 2012 10:20:48 GMT
```

ポータルの外部 RESTful サービス API

次の表に、ポータルの外部 RESTful サービス API を示します。

表 7-41　　ポータルに使用できる API

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのポータルの取得	GET	/ers/config/portal	適用対象外	Page、Size、sortacs または sortdsn、Filter
ID によるポータルの取得	GET	/ers/config/portal/{id}	適用対象外	

すべてのポータルの取得

次の表に、Get All Portals API コールの主な特性を示します。

表 7-42　　Get All Portals API コールの主な特性

説明	ポータルのコレクションを取得します
概要	GET /ers/config/portal
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

Get All Portals コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/portal
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.portal.1.0+xml
```

Get All Portals コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="utf-8" standalone="yes"?> <ns2:searchResult total="2"
xmlns:ns2="ers.ise.cisco.com">
        <resources>
            <resource name="portal1" id="id1">
                <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/id1" rel="self"/>
            </resource>
            <resource name="portal2" id="id2">
                <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/id2" rel="self"/>
            </resource>
```

```
        </resources>
</ns2:searchResult>
```

ID によるポータルの取得

次の表に、Get Portal by ID API コールの主な特性を示します。

表 7-43　Get Portal by ID API コールの主な特性

説明	指定されたポータルを取得します
概要	GET /ers/config/portal/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	Portal タイプのリソース
応答ステータス	200、400、401、403、404、415、500

Get Portal by ID コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/portal/333
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.identity.portal.1.0+xml
```

Get Portal by ID コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.identity.portal.1.0+xml Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:portal name="sponsor" id="d7b703f0-b073-11e3-bd6c- 005056a15fa7"
        xmlns:ns2="ers.ise.cisco.com"
        xmlns:ns3="identity.ers.ise.cisco.com">
        <link type="application/xml"
href="https://<ISE-ADMIN-NODE>:9060/ers/config/portal/333" rel="self"/>
    <allowSponsorToChangeOwnPassword>false</allowSponsorToChangeOwnPassword>
        <GuestUserFieldList>
            <GuestUserField>
                <customType>false</customType>
                <dataType>DROPDOWN</dataType>
    <dictionaryLabelKey>ui_sms_provider_label</dictionaryLabelKey>
                <labelName>SMS Service Provider</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>TEXT</dataType>
    <dictionaryLabelKey>ui_company_label</dictionaryLabelKey>
                <labelName>Company</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
```

```
                <dataType>TEXT</dataType>
    <dictionaryLabelKey>ui_first_name_label</dictionaryLabelKey>
                <labelName>First name</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>TEXT</dataType>
    <dictionaryLabelKey>ui_reason_visit_label</dictionaryLabelKey>
                <labelName>Reason for visit</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>true</customType>
                <dataType>TEXT</dataType>
    <dictionaryLabelKey>ui_ssn-number_text_label</dictionaryLabelKey>
                <instructionText>social </instructionText>
                <labelName>ssn-number</labelName>
                <required>false</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>PHONE</dataType>
    <dictionaryLabelKey>ui_phone_number_label</dictionaryLabelKey>
                <labelName>Phone number</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>EMAIL</dataType>
    <dictionaryLabelKey>ui_person_visited_label</dictionaryLabelKey>
                <labelName>Person being visited</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>EMAIL</dataType>
    <dictionaryLabelKey>ui_email_address_label</dictionaryLabelKey>
                <labelName>Email address</labelName>
                <required>true</required>
            </GuestUserField>
            <GuestUserField>
                <customType>false</customType>
                <dataType>TEXT</dataType>
    <dictionaryLabelKey>ui_last_name_label</dictionaryLabelKey>
                <labelName>Last name</labelName>
                <required>true</required>
            </GuestUserField>
    </GuestUserFieldList>
</ns3:portal>
}
```

ネットワーク デバイスの外部 RESTful サービス API

次の表に、ネットワーク デバイスの外部 RESTful サービス API を示します。

表 7-44　　ポータルに使用できる API

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのネットワーク デバイスの取得	GET	/ers/config/networkdevice	適用対象外	Page、Size、sortacs または sortdsn、Filter
ネットワーク デバイスの取得	GET	/ers/config/networkdevice /{id}	適用対象外	
ネットワーク デバイスの作成	POST	/ers/config/networkdevice	networkdevice	
ネットワーク デバイスの更新	PUT	/ers/config/networkdevice /{id}	networkdevice	
ネットワーク デバイスの削除	DELETE	/ers/config/networkdevice /{id}	適用対象外	
ネットワーク デバイス リソースのバージョン情報の取得	GET	/ers/config/ networkdevice /versioninfo	適用対象外	

すべてのネットワーク デバイスの取得

次の表に、Get All Network Devices API コールの主な特性を示します。

表 7-45　　Get All Network Devices API コールの主な特性

説明	ネットワーク デバイス リソースの集合を取得します
概要	GET /ers/config/networkdevice
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

Get All Network Devices コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice?page=1&size=20&sortacs=name
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
```

Get All Network Devices コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:searchResult
        xmlns:ns2="ers.ise.cisco.com" total="1">
        <resources>
            <resource name="nd1" id="0d008bb0-2539-11e3-84ad-
    00215edbb2a8">
                <link type="application/xml"
    href="https://10.56.13.196:9060/ers/config/networkdevice/0d0
    08bb0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
            </resource>
        </resources>
    </ns2:searchResult>
}
```

ID によるネットワーク デバイスの取得

次の表に、Get Network Device by ID API コールの主な特性を示します。

表 7-46　Get Network Device by ID API コールの主な特性

説明	指定されたネットワーク デバイスを取得します
概要	GET /ers/config/networkdevice/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	Network Device タイプのリソース
応答ステータス	200、400、401、403、404、415、500

Get Network Device by ID コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/333
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
```

Get Network Device by ID コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml Content-Length:
16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns3:networkdevice
        xmlns:ns2="ers.ise.cisco.com"
        xmlns:ns3="network.ers.ise.cisco.com" name="nd1"
    id="0d008bb0-2539-11e3-84ad-00215edbb2a8">
        <link type="application/xml"
    href="https://10.56.13.196:9060/ers/config/networkdevice/0d0
```

```
    08bb0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
        <authenticationSettings>
            <enableKeyWrap>false</enableKeyWrap>
            <keyInputFormat>ASCII</keyInputFormat>
            <networkProtocol>RADIUS</networkProtocol>
            <radiusSharedSecret>*****</radiusSharedSecret>
        </authenticationSettings>
<NetworkDeviceIPList>
        <NetworkDeviceIP>
            <ipaddress>1.2.3.4</ipaddress>
            <mask>32</mask>
        </NetworkDeviceIP>
    </NetworkDeviceIPList>
    <modelName>Unknown</modelName>
    <NetworkDeviceGroupList>
        <NetworkDeviceGroup>1d8c62b0-2539-11e3-84ad-
00215edbb2a8</NetworkDeviceGroup>
        <NetworkDeviceGroup>37053aa0-2539-11e3-84ad-
00215edbb2a8</NetworkDeviceGroup>
    </NetworkDeviceGroupList>
    <softwareVersion>Unknown</softwareVersion>
</ns3:networkdevice>
}
```

ネットワーク デバイスの作成

次の表に、Create Network Device API コールの主な特性を示します。

表 7-47　　Create Network Device API コールの主な特性

説明	指定されたネットワーク デバイスを作成します
概要	POST /ers/config/networkdevice/
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	NetworkDevice
応答ヘッダー	Content-Length、Content-Type、Location
応答メッセージ本文	該当なし
応答ステータス	200、400、401、403、415、500

Create Network Device コールの要求例

```
POST https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml {
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <ns3:networkdevice
        xmlns:ns2="ers.ise.cisco.com"
        xmlns:ns3="network.ers.ise.cisco.com" name="nd2">
        <authenticationSettings>
            <enableKeyWrap>false</enableKeyWrap>
            <keyInputFormat>ASCII</keyInputFormat>
            <networkProtocol>RADIUS</networkProtocol>
            <radiusSharedSecret>acsi</radiusSharedSecret>
        </authenticationSettings>
        <NetworkDeviceIPList>
```

```
            <NetworkDeviceIP>
                <ipaddress>1.2.3.4</ipaddress>
                <mask>32</mask>
            </NetworkDeviceIP>
        </NetworkDeviceIPList>
        <modelName>Unknown</modelName>
        <NetworkDeviceGroupList>
            <NetworkDeviceGroup>1d8c62b0-2539-11e3-84ad-
    00215edbb2a8</NetworkDeviceGroup>
            <NetworkDeviceGroup>37053aa0-2539-11e3-84ad-
    00215edbb2a8</NetworkDeviceGroup>
        </NetworkDeviceGroupList>
        <softwareVersion>Unknown</softwareVersion>
    </ns3:networkdevice>
}
```

Create Network Device コールの応答例

```
HTTP/1.1 201 OK (see location header for the ID of the new device)
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
Location: https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/444
```

ネットワーク デバイスの更新

次の表に、Update Network Device API コールの主な特性を示します。

表 7-48　　Update Network Device API コールの主な特性

説明	指定されたネットワーク デバイスを更新します
概要	PUT /ers/config/networkdevice/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	NetworkDevice
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	更新されたフィールドのリスト
応答ステータス	200、400、401、403、415、500

Update Network Device コールの要求例

```
PUT https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevice/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Content-Type: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml {
    <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
    <ns3:networkdevice
        xmlns:ns2="ers.ise.cisco.com"
        xmlns:ns3="network.ers.ise.cisco.com"
    name="nd2_updated">
        <authenticationSettings>
            <enableKeyWrap>true</enableKeyWrap>
        </authenticationSettings>
    </ns3:networkdevice>
}
```

Update Network Device コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.updatedfields.1.0+xml Content-Length: 529
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns2:updatedFields
xmlns:ns2="ers.ise.cisco.com">
    <updatedField field="name">
        <newValue>nd2_updated</newValue>
        <oldValue>nd2</oldValue>
    </updatedField>
    <updatedField field="enableKeywrap">
        <newValue>true</newValue>
        <oldValue>false</oldValue>
    </updatedField>
</ns2:updatedFields>
}
```

ネットワーク デバイスの削除

次の表に、Delete Network Device API コールの主な特性を示します。

表 7-49　Delete Network Device API コールの主な特性

説明	指定されたネットワーク デバイスを削除します
概要	DELETE /ers/config/networkdevice/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	該当なし
応答ステータス	200、204、400、401、403、404、415、500

Delete Network Device コールの要求例

```
DELETE https://<ISE-ADMIN-NODE>:9060/ers/config/networjdevice/333
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.network.networkdevice.1.0+xml
```

Delete Network Device コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
```

ネットワーク デバイス グループの外部 RESTful サービス API

次の表に、ネットワーク デバイス グループの外部 RESTful サービス API を示します。

表 7-50　SGT に使用できる API

オペレーション	メソッド	URL	内容	クエリー文字列
すべてのネットワーク デバイス グループの取得	GET	/ers/config/networkdevicegroup	適用対象外	page、size、sortacs または sortdsn、filter
ネットワーク デバイス グループの取得	GET	/ers/config/networkdevicegroup/{id}	適用対象外	
ネットワーク デバイス グループ リソースのバージョン情報の取得	GET	/ers/config/networkdevicegroup/versioninfo	適用対象外	

すべてのネットワーク デバイス グループの取得

次の表に、Get All Network Device Groups API コールの主な特性を示します。

表 7-51　Get All Network Device Groups API コールの主な特性

説明	ネットワーク デバイス グループ リソースの集合を取得します
概要	GET /ers/config/networkdevicegroup
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、500

Get All Network Device Groups API コールの要求例

```
GET
https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevicegroup?page=1&size=20&sortacs=name
    Authorization: Basic xxxxxxxxxxxxxxxxxx
    Accept: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0+xml
```

Get All Network Device Groups API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml Content-Length: 16347
{
   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
                        <ns2:searchResult
             xmlns:ns2="ers.ise.cisco.com" total="0">
                           <resources>
  <resource name="Location#All Locations#loc1" id="1d8c62b0-2539-11e3-84ad-00215edbb2a8"
description="xxx">
                  <link type="application/xml"
href="https://10.56.13.196:9060/ers/config/networkdevicegroup/1d8c62b0-2539-11e3-84ad-0021
5edbb2a8" rel="self"/>
                           </resource>
   <resource name="Device Type#All Device Types#device type 555"
id="37053aa0-2539-11e3-84ad-00215edbb2a8" description="vvv">
                  <link type="application/xml"
 href="https://10.56.13.196:9060/ers/config/networkdevicegrou
     p/37053aa0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
                           </resource>
                           </resources>
                      </ns2:searchResult>
}
```

ネットワーク デバイス グループの取得

次の表に、Get Network Device Groups API コールの主な特性を示します。

表 7-52　　Get Network Device Group API コールの主な特性

説明	指定されたネットワーク デバイス グループを取得します
概要	GET /ers/config/networkdevicegroup/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	NetworkDeviceGroup タイプのリソース
応答ステータス	200、400、401、403、404、415、429、500

Get Network Device Group API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/networkdevicegroup/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0+xml
```

Get Network Device Group API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.network.networkdevicegroup.1.0 +xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?> <ns3:networkdevicegroup
        xmlns:ns2="ers.ise.cisco.com"
        xmlns:ns3="network.ers.ise.cisco.com" name="Location#All
    Locations#loc1" id="1d8c62b0-2539-11e3-84ad-00215edbb2a8"
    description="xxx">
        <link type="application/xml"
    href="https://10.56.13.196:9060/ers/config/networkdevicegrou
    p/1d8c62b0-2539-11e3-84ad-00215edbb2a8" rel="self"/>
        <type>Location</type>
</ns3:networkdevicegroup>
}
```

SGT の外部 RESTful サービス API

次の表に、SGT の外部 RESTful サービス API を示します。

表 7-53　SGT に使用できる API

オペレーション	メソッド	URL	内容	クエリー文字列
すべての SGT の取得	GET	/ers/config/sgt	適用対象外	page、size、sortacs または sortdsn、filter
SGT の取得	GET	/ers/config/sgt/{id[1] }	適用対象外	
GST リソースのバージョン情報の取得	GET	/ers/config/sgt/versioninfo	適用対象外	

1. SGT ID は、Cisco ISE データベースに保存された UUID タイプです。

すべての SGT の取得

次の表に、Get All SGTs API コールの主な特性を示します。

表 7-54　Get All SGTs API コールの主な特性

説明	SGT リソースの集合を取得します
概要	GET /ers/config/sgt
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	page、size、sortbyacn、sortbydcn、filter
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	SearchResult
応答ステータス	200、400、401、403、404、415、429、500

Get All SGTs API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/sgt?page=0&size=20&sortacs=name
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.sga.sgt.1.0+xml
```

Get All SGTs API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.ers.searchresult.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns2:searchResult total="2" xmlns:ns2="ers.ise.cisco.com">
<resources>
  <resource name="name1" id="id1" description="description1">
   <link type="application/xml" href="https://<ISE-ADMIN-NODE>:9060/ers/config/sgt/id1"
rel="self"/>
  </resource>
 </resources>
</ns2:searchResult>
}
```

ID による SGT の取得

次の表に、Get SGT by ID API コールの主な特性を示します。

表 7-55　Get SGTs API コールの主な特性

説明	指定された SGT を取得します
概要	GET /ers/config/sgt/{id}
要求ヘッダー	Accept、Authorization、Host
クエリー文字列	該当なし
要求メッセージ本文	該当なし
応答ヘッダー	Content-Length、Content-Type
応答メッセージ本文	InternalUser タイプのリソース
応答ステータス	200、400、401、403、404、415、429、500

Get SGT by ID API コールの要求例

```
GET https://<ISE-ADMIN-NODE>:9060/ers/config/sgt/333
Authorization: Basic xxxxxxxxxxxxxxxxxx
Accept: application/vnd.com.cisco.ise.sga.sgt.1.0+xml
```

Get SGT by ID API コールの応答例

```
HTTP/1.1 200 OK
Date: Thu, 12 Jul 2012 23:59:59 GMT
Content-Type: application/vnd.com.cisco.ise.sga.sgt.1.0+xml
Content-Length: 16347
{
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ns3:sgt description="description" name="name" id="id" xmlns:ns2="ers.ise.cisco.com"
xmlns:ns3="sga.ers.ise.cisco.com">
    <generationId>generationId</generationId>
    <isTagFromRange>isTagFromRange</isTagFromRange>
    <value>1</value>
</ns3:sgt>
}
```

REST API クライアント

外部 RESTful サービス API を使用すると、Cisco ISE リソースに対して CRUD（作成、読み取り、更新、削除）操作を実行できます。Cisco ISE サーバと通信し、Cisco ISE サーバで操作を実行する外部 RESTful サービス API を使用してアプリケーションを構築し、テストするには、業界標準の REST API クライアント（Google Chrome の POSTMAN プラグインなど）を使用します。

REST のアーキテクチャおよび原則に従って設計されている POSTMAN を使用して、Google Chrome Web ブラウザで、標準の HTTP/HTTPS 要求/応答を取得および送信することができます。次の標準 HTTP メソッドを使用して、Cisco ISE リソースに対して CRUD 操作を実行できます。

- GET
- POST
- PUT
- DELETE

ERS API により、さまざまな API コールでこれらの HTTP 要求が使用できるようになり、これらの HTTP 要求により、Cisco ISE サーバに対する操作を実行できるようになります。これらの HTTP 要求が使用される操作の包括的なリストについては、<ERS API Operations> を参照してください。

（注）　POSTMAN プラグインをダウンロードするには、https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojpjoooidkmcomcm?hl=en にアクセスしてください。POSTMAN プラグインの使用方法の詳細については、https://github.com/a85/POSTMan-Chrome-Extension/wiki を参照してください。

GET メソッド

指定されたリソースの表現を要求します。GET を使用した要求は、データを取得するだけで、他の効果はありません。

（注）　ここでは、POSTMAN を使用して ERS API コールを呼び出す方法を示します。この API コールは、GET HTTP メソッドを使用します。また、このセクションで説明されていない ERS API の他のコンポーネントも使用します。特性、要求、および応答など、さまざまな ERS API コンポーネントの詳細については、「外部 RESTful サービス API の操作」を参照してください。

GET HTTPS メソッドを使用する ERS API コールの要求本体には、次の 3 つの構成ブロックが含まれます。

- URI
- Accept ヘッダー
- Authorization ヘッダー

URI

GET メソッドは、URI を Cisco ISE サーバに送信し、HTTP 返信は、未処理の結果データになります。一般的な URI は、次の形式に従う必要があります。

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*

  ここで、*<Cisco ISE Server Address>* は Cisco ISE サーバのサーバ アドレスを示し、*<port>* はポート 9060 を示し、*<namespace>* は ISE リソースが属する名前空間を示し、*<Cisco ISE Resource Name>* は Cisco ISE リソースの名前を示します。

次に、*interaluser* ISE リソースのデータを要求する URI の例を示します。

- *https://10.56.13.196:9060/ers/config/internaluser*

（注）　URI は、要求本文ではなく、URL にすぎません。この URL は、GET メソッドを使用してサーバに送信されます。

Accept ヘッダー

Accept ヘッダーは次の形式に従う必要があります。

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

ここで、*<resource-namespace>* は Cisco ISE リソースが属する名前空間を示し、*<resource-type>* は ISE リソースのタイプを示し、*<major-version>* は ISE 導入のメジャー バージョン番号を示し、*<minor-version>* は ISE 導入のマイナー バージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization ヘッダー

Authorization ヘッダーには、GET 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。

（注）　暗号化キーの生成の詳細については、「POSTMAN を使用した GET 要求の実行」（P.7-57）を参照してください。

POSTMAN を使用した GET 要求の実行

手順

ステップ 1　Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2　左側のペインのオプションを使用して、新しい収集を作成します。

（注）　POSTMAN プラグインの使用方法の詳細については、https://github.com/a85/POSTMan-Chrome-Extension/wiki を参照してください。

ステップ 3　ドロップダウン メニューから、**[GET]** を選択します。

ステップ 4　URL バーに、URI を入力します。

URI で、通信する Cisco ISE サーバ、およびアクセスする ISE リソースを指定します。URI の形式の詳細については、「URI」（P.7-56）を参照してください。

ステップ 5　**[Basic Auth]** タブをクリックします。

ユーザ アクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6　アクセス クレデンシャルを [Username] および [Password] フィールドに指定し、**[Refresh Headers]** をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7　次の値を指定して、Accept ヘッダーを追加します。
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml

（注）　Accept ヘッダーの詳細については、「Accept ヘッダー」（P.7-56）を参照してください。

ステップ 8　**[Send]** をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 200 OK ステータスの応答を表示します。要求は、URL で指定したリソースの詳細も返します。

POST メソッド

要求に含まれているエンティティを、URI で識別される Web リソースの新しい下位として、サーバが承認することを要求します。

（注）　ここでは、POSTMAN を使用して ERS API コールを呼び出す方法を示します。この API コールは、POST HTTP メソッドを使用します。また、このセクションで説明されていない ERS API の他のコンポーネントも使用します。特性、要求、および応答など、さまざまな ERS API コンポーネントの詳細については、「外部 RESTful サービス API の操作」を参照してください。

POST HTTP メソッドを使用する ERS API コールの要求本体には、次の 3 つの構成ブロックが含まれます。

- URI
- Content-Type ヘッダー
- Authorization ヘッダー

URI

POST メソッドは、URI を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*

  ここで、*<Cisco ISE Server Address>* は Cisco ISE サーバのサーバ アドレスを示し、*<port>* はポート 9060 を示し、*<namespace>* は ISE リソースが属する名前空間を示し、*<Cisco ISE Resource Name>* は Cisco ISE リソースの名前を示します。

次に、*interaluser* ISE リソースのデータを要求する URI の例を示します。

- *https://10.56.13.196:9060/ers/config/internaluser*

（注）　URI は、要求本文ではなく、URL にすぎません。この URL は、POST メソッドを使用してサーバに送信されます。

Content-Type ヘッダー

Content-Type ヘッダーは次の形式に従う必要があります。

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

  ここで、*<resource-namespace>* は Cisco ISE リソースが属する名前空間を示し、*<resource-type>* は ISE リソースのタイプを示し、*<major-version>* は ISE 導入のメジャー バージョン番号を示し、*<minor-version>* は ISE 導入のマイナー バージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization ヘッダー

Authorization ヘッダーには、POST 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。

（注）　暗号化キーの生成の詳細については、「POSTMAN を使用した POST 要求の実行」（P.7-59）を参照してください。

POSTMAN を使用した POST 要求の実行

手順

ステップ 1　Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2　左側のペインのオプションを使用して、新しい収集を作成します。

（注）　POSTMAN プラグインの使用方法の詳細については、https://github.com/a85/POSTMan-Chrome-Extension/wiki を参照してください。

ステップ 3　ドロップダウン メニューから、**[POST]** を選択します。

ステップ 4　URI バーに、URI を入力します。

URI で、通信する Cisco ISE サーバ、およびアクセスする ISE リソースを指定します。URI の形式の詳細については、「URI」（P.7-58）を参照してください。

ステップ 5　**[Basic Auth]** タブをクリックします。

ユーザ アクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6　アクセス クレデンシャルを [Username] および [Password] フィールドに指定し、[Refresh Headers] をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7　次の値を指定して、Content-Type ヘッダーを追加します。
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml

（注）　Accept ヘッダーの詳細については、「Content-Type ヘッダー」（P.7-58）を参照してください。

ステップ 8　[raw] の横に表示されるドロップダウン メニューから、**[XML]** を選択します。

ステップ 9　**[raw]** をクリックします。

ステップ 10　POSTMAN プラグインは、POST 要求の本文を指定できる編集ペインを開きます。

ステップ 11　編集ペインで POST 要求のメッセージ本文を入力します。

（注）　このメッセージ本文には、ISE サーバに作成しようとする ISE リソースに対応する詳細を含める必要があります。たとえば、interaluser の作成時に、internaluser 名前、internaluser の説明、パスワードなどの詳細を指定する必要があります。POST 要求を使用する ERS API コールのメッセージ本文、および指定する必要がある ISE リソースの詳細の詳細については、「外部 RESTful サービス API の操作」を参照してください。

ステップ 12　**[Send]** をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 201 CREATED ステータスの応答を表示します。ISE GUI に移動して、追加した ISE リソースが ISE GUI に表示されるかどうかを確認できます。

PUT メソッド

含まれているエンティティを指定された URI 下に格納することを要求します。URI がすでに存在しているリソースを指す場合は、変更されます。URI が既存のリソースを指さない場合は、サーバはその URI でリソースを作成できます。

（注）　ここでは、POSTMAN を使用して ERS API コールを呼び出す方法を示します。この API コールは、PUT HTTP メソッドを使用します。また、ここで説明されていない ERS API の他のコンポーネントも使用します。特性、要求、および応答など、さまざまな ERS API コンポーネントの詳細については、「外部 RESTful サービス API の操作」を参照してください。

POST HTTP メソッドを使用する ERS API コールの要求本体には、次の 3 つの構成ブロックが含まれます。

- URI
- Content-Type ヘッダー
- Authorization ヘッダー

URI

PUT メソッドは、URI を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*

  ここで、*<Cisco ISE Server Address>* は Cisco ISE サーバのサーバ アドレスを示し、*<port>* はポート 9060 を示し、*<namespace>* は ISE リソースが属する名前空間を示し、*<Cisco ISE Resource Name>* は Cisco ISE リソースの名前を示します。

次に、*interaluser* ISE リソースのデータを要求する URI の例を示します。

- *https://10.56.13.196:9060/ers/config/internaluser*

（注）　URI は、要求本文ではなく、URL にすぎません。この URL は、PUT メソッドを使用してサーバに送信されます。

Content-Type ヘッダー

Content-Type ヘッダーは次の形式に従う必要があります。

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

ここで、<*resource-namespace*> は Cisco ISE リソースが属する名前空間を示し、<*resource-type*> は ISE リソースのタイプを示し、<*major-version*> は ISE 導入のメジャー バージョン番号を示し、<*minor-version*> は ISE 導入のマイナー バージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization ヘッダー

Authorization ヘッダーには、PUT 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。

（注） 暗号化キーの生成の詳細については、「POSTMAN を使用した PUT 要求の実行」（P.7-61）を参照してください。

POSTMAN を使用した PUT 要求の実行

手順

ステップ 1 Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2 左側のペインのオプションを使用して、新しい収集を作成します。

（注） POSTMAN プラグインの使用方法の詳細については、https://github.com/a85/POSTMan-Chrome-Extension/wiki を参照してください。

ステップ 3 ドロップダウン メニューから、**[PUT]** を選択します。

ステップ 4 URI バーに、URI を入力します。

URI で、通信する Cisco ISE サーバ、およびアクセスする ISE リソースを指定します。URI の形式の詳細については、「URI」（P.7-60）を参照してください。

ステップ 5 **[Basic Auth]** タブをクリックします。

ユーザ アクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6 アクセス クレデンシャルを [Username] および [Password] フィールドに指定し、**[Refresh Headers]** をクリックします。

POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7 次の値を指定して、Content-Type ヘッダーを追加します。
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml

（注） Accept ヘッダーの詳細については、「Content-Type ヘッダー」（P.7-60）を参照してください。

ステップ 8 [raw] の横に表示されるドロップダウン メニューから、**[XML]** を選択します。

ステップ 9 **[raw]** をクリックします。

ステップ 10 POSTMAN プラグインは、POST 要求の本文を指定できる編集ペインを開きます。

ステップ 11　編集ペインで POST 要求のメッセージ本文を入力します。

（注）　このメッセージ本文には、ISE サーバで更新しようとする ISE リソースに対応する詳細を含める必要があります。たとえば、interaluser の更新時に、internaluser 名前、internaluser の説明、パスワードなどの詳細を指定する必要があります。POST 要求を使用する ERS API コールのメッセージ本文、および指定する必要がある ISE リソースの詳細の詳細については、「外部 RESTful サービス API の操作」を参照してください。

ステップ 12　**[Send]** をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 201 CREATED ステータスの応答を表示します。ISE GUI に移動して、追加した ISE リソースが ISE GUI に表示されるかどうかを確認できます。

Delete メソッド

指定されたリソースを削除します。

（注）　ここでは、POSTMAN を使用して ERS API コールを呼び出す方法を示します。この API コールは、DELETE HTTP メソッドを使用します。また、ここで説明されていない ERS API の他のコンポーネントも使用します。特性、要求、および応答など、さまざまな ERS API コンポーネントの詳細については、「外部 RESTful サービス API の操作」を参照してください。

DELETE HTTP メソッドを使用する ERS API コールの要求本体には、次の 3 つの構成ブロックが含まれます。

- URI
- Accept ヘッダー
- Authorization ヘッダー

URI

DELETE メソッドは、URI を Cisco ISE サーバに送信します。一般的な URI は、次の形式に従う必要があります。

- *https://<Cisco ISE Server address:<port>/<namespace>/config/<Cisco ISE Resouce Name>*

  ここで、*<Cisco ISE Server Address>* は Cisco ISE サーバのサーバ アドレスを示し、*<port>* はポート 9060 を示し、*<namespace>* は ISE リソースが属する名前空間を示し、*<Cisco ISE Resource Name>* は Cisco ISE リソースの名前を示します。

次に、*interaluser* ISE リソースのデータを要求する URI の例を示します。

- *https://10.56.13.196:9060/ers/config/internaluser*

（注）　URI は、要求本文ではなく、URL にすぎません。この URL は、GET メソッドを使用してサーバに送信されます。

Accept ヘッダー

Accept ヘッダーは次の形式に従う必要があります。

- *application/vnd.com.cisco.ise.<resource-namespace>.<resource-type>.<major version>.<minor version>+xml*

ここで、*<resource-namespace>* は Cisco ISE リソースが属する名前空間を示し、*<resource-type>* は ISE リソースのタイプを示し、*<major-version>* は ISE 導入のメジャー バージョン番号を示し、*<minor-version>* は ISE 導入のマイナー バージョン番号を示します。

次に、一般的な Accept ヘッダーの例を示します。

- *application/vnd.com.cisco.ise.identity.internaluser.1.0+xml*

Authorization ヘッダー

Authorization ヘッダーには、DELETE 要求に組み込まれている暗号化認証キーが含まれます。認可のクレデンシャルを指定した後、暗号化キーを生成する必要があります。その後、暗号化キーは要求の本体に埋め込まれます。

（注） 暗号化キーの生成の詳細については、「POSTMAN を使用した DELETE 要求の実行」（P.7-63）を参照してください。

POSTMAN を使用した DELETE 要求の実行

手順

ステップ 1　Google Chrome ブラウザで POSTMAN プラグインを開きます。

ステップ 2　左側のペインのオプションを使用して、新しい収集を作成します。

（注） POSTMAN プラグインの使用方法の詳細については、https://github.com/a85/POSTMan-Chrome-Extension/wiki を参照してください。

ステップ 3　ドロップダウン メニューから、**[DELETE]** を選択します。

ステップ 4　URL バーに、URI を入力します。

URI で、通信する Cisco ISE サーバ、およびアクセスする ISE リソースを指定します。URI の形式の詳細については、「URI」（P.7-62）を参照してください。

ステップ 5　**[Basic Auth]** タブをクリックします。

ユーザ アクセス クレデンシャルを指定できるオプションが表示されます。

ステップ 6　アクセス クレデンシャルを [Username] および [Password] フィールドに指定し、**[Refresh Headers]** をクリックします。

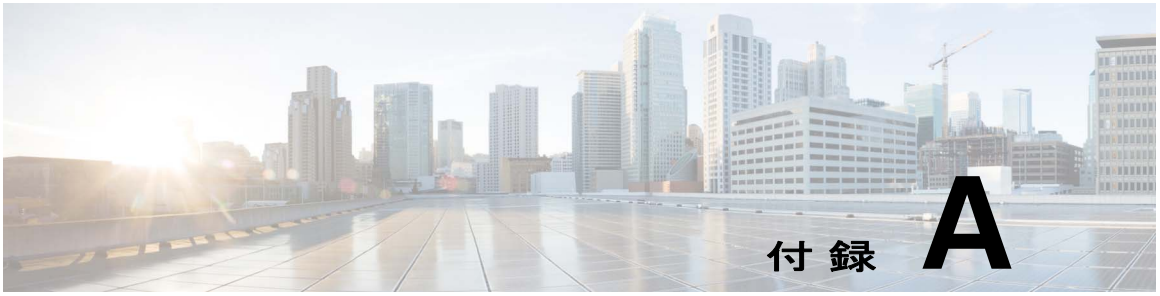
POSTMAN は暗号化キーを含む Authorization ヘッダーを表示します。

ステップ 7　次の値を指定して、Accept ヘッダーを追加します。
application/vnd.com.cisco.ise.ers.<namespace>.<ise resource>.1.0+xml

（注） Accept ヘッダーの詳細については、「Accept ヘッダー」（P.7-63）を参照してください。

ステップ 8　**[Send]** をクリックします。

POSTMAN プラグインは、要求が成功していることを示す 200 OK ステータスの応答を表示します。指定された ISE リソースが ISE サーバから削除されます。



Cisco ISE 障害理由レポート

この付録では、Cisco ISE 障害理由レポートにアクセスするための手順を提供します。Cisco ISE 障害理由レポートには、障害理由のリストが示されます。

はじめに

Cisco ISE 障害理由レポートは、検出できる障害理由すべてに関する情報を提供する Cisco ISE ユーザ インターフェイスのオプションです。API を解決する Cisco ISE クエリーを使用すると Get Failure Reason Mapping コールから出力として返されるオプションをチェックする場合に使用できます。

Cisco ISE 障害理由レポートを使用すると、Cisco ISE ソフトウェアによって定義された Cisco Monitoring ISE ノード動作に適用する障害理由の全リストにアクセスできるようになります。次の手順により、定義された障害理由のリストを表示または編集することができます。障害理由を表示し、ここにアクセスするには、宛先 Cisco Monitoring ISE ノードの Cisco ISE ユーザ インターフェイスにログインする必要があります。ロギングに関する詳細については、「モニタリング ノードの確認」（P.1-2）を参照してください。

障害理由の表示

ステップ 1 **[操作（Operations）] > [レポート（Reports）] > [認証の要約（Authentication Summary）]** レポートを選択します。

ステップ 2 ナビゲーション パネルの **[モニタリング（Monitoring）]** を展開し、**[障害理由エディタ（Failure Reason Editor）]** を選択します。

ステップ 3 提供されたフィルタのリストから [障害理由（Failure Reasons）] を選択します。

ステップ 4 探している障害理由を指定します。

ステップ 5 [実行（Run）] をクリックします。

障害理由のリストが右側のパネルに表示されます。

ステップ 6 任意の障害理由をクリックして、新しいウィンドウで詳細レポートを取得します。