ASA REST API v1.2.2 について

初版発行日：2014 年 12 月 16 日

改訂日：2015 年 2 月 5 日

目次

[非表示]

- 概要
  - サポートされるプラットフォーム
  - サポート対象モード
  - ハイレベルのアーキテクチャ
  - 一般的な要求フロー
- リソース ID
  - リソース URL：「selfLink」属性
  - リソース タイプ：「kind」属性
    - プリミティブ kind
  - リソース アソシエーション
  - オブジェクト「rangeInfo」
- REST API 認証
- REST API 規則
- REST API コード
  - JSON のエラーまたは警告の応答スキーマ
- ASA の REST API エージェント
  - ASA REST API エージェントのインストールと有効化
    - 「rest-api image」コマンド
    - REST API エージェントに必要な追加のブートストラップ
    - 「rest-api agent」コマンド
    - 「show rest-api agent」コマンド
    - 「show version」コマンド
  - REST API エージェント デバッグ
    - サポート対象モード
    - show コマンドの出力
- Syslog
  - Syslog # 342001
  - Syslog # 342002
  - Syslog # 342003

Syslog # 342004
Syslog # 342005
Syslog # 342006
Syslog # 342007
Syslog # 342008
アウトオブバンド変更処理
サポートされる ASA 機能
AAA
認証
認証
コマンド特権
アクセス ルール
Back Up and Restore
DHCP
DNS
フェールオーバー
インターフェイス
IP 監査
ライセンス
永久ライセンスおよびアクティベーション キーのライセンス
共有ライセンス
スマート ライセンス
ロギング
syslog サーバ
syslog サーバの設定
syslog メッセージの構成
syslog メッセージの設定
Netflow の構成
管理アクセス
汎用管理アクセス
ホスト
モニタリング
マルチ コンテキスト モード
NTP
NAT
ObjectNAT（AutoNAT）
TwiceNAT（手動 NAT）
オブジェクト
プロトコルのタイムアウト
ルーティング
サービス ポリシー
VPN
特別 API
一括 API
汎用 CLI コマンド Executer API

制限事項
トークン認証 API
メモリ書き込み API
REST API のオンライン マニュアル
スクリプトの種類
生成されたスクリプトを使用するための前提条件
法的情報
シスコの商標

概要

この REST API は、9.3.2 リリース以降、標準 ASA 設定用にプログラミング モデル ベースのインターフェイスを提供します。この「標準 ASA」という用語は、CX または SourceFire センサーを含まないアプライアンスか、NGFW（次世代ファイアウォール）サービスの統合機能を示します。その他のセキュリティ モジュールに標準 ASA がある場合、これらのモジュール用の API は無いことにも注意してください。

REST API は、既存の管理インターフェイスおよびアプリケーション（コマンドライン インターフェイス（CLI）、Adaptive Security Device Manager（ASDM）、Cisco Security Manager（CSM））と共に ASA 設定に使用できます。

REST API 1.2.2 の新機能：

- スマート ライセンス
- IP 監査と追加アプリケーション インスペクション プロトコル（FTP、NetBIOS、RTSP、SIP、SQL*Net）のサポート。
- ASA のシリアル番号により照会できます。
- REST API 1.2.2.200 のリリースは CSCux92088 の変更（API 一括要求エントリの上限を 1000 個に増やす）を含みます。

REST API 1.2.1 の新機能：

- マルチ コンテキスト モードのモニタリング サポート。
- 次の ASA 機能のサポート：DHCP サーバおよび DHCP リレー、DNS クライアントおよびダイナミック DNS、プロトコル タイムアウト（PTO）、GTP インスペクション。

REST API 1.1.1 の新機能：

- トークン ベース認証のサポート。
- 次の ASA 機能のサポート：アプリケーション インスペクション プロトコル（DNS over UDP、HTTP、ICMP、ICMP エラー、RTSP、DCERPC、IP オプション）、バックアップと復元、接続制限、マルチ コンテキスト（限定的なサポート）、NTP およびメモリ書き込みコマンド API。

REST API 1.0.1 の機能：

- 次の ASA 機能のサポート：AAA、アクセス ルール、フェールオーバー、インターフェイス、ライセンス（永久ライセンスおよびアクティベーション キーのライセンス）、共有秘密ライセンス、ロギング、管理アクセス、モニタリング、NAT（Twice NAT およびオブジェクト NAT）、オブジェクト、スタティック ルーティング、サービス ポリシーおよびサイト間 VPN。

- 一括 API。
- 汎用 CLI コマンド Executor API。REST API を使用してすべての CLI コマンドを送信できます。

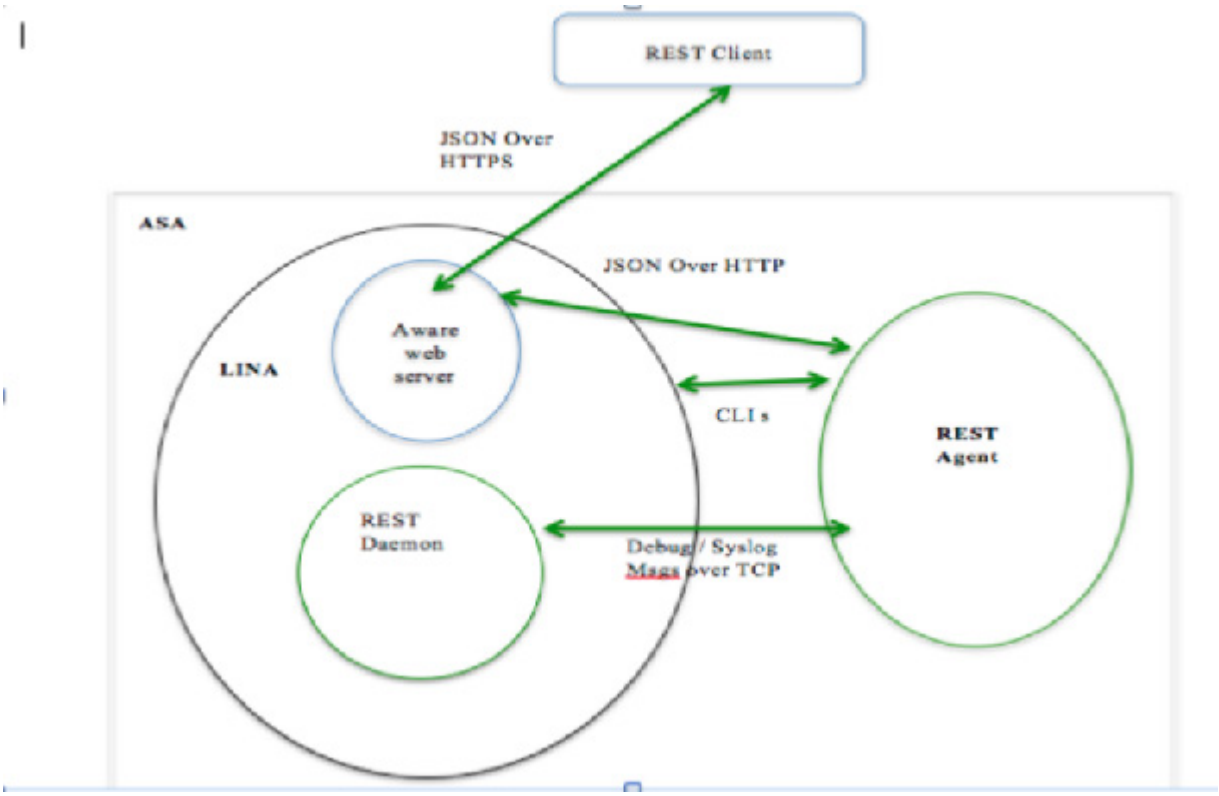
サポートされるプラットフォーム

REST API は 5500-X シリーズ（5585-X を含む）および ASAv プラットフォーム、および Firepower 9300 ASA セキュリティ モジュールでのみサポートされ、ASA サービス モジュール（ASA-SM）ではサポートされません。詳細については、「ASA REST API Compatibility」を参照してください。

サポート対象モード

REST API は現在マルチモードのオプションの直接設定をサポートしません。汎用 CLI コマンド Executer API（CLI パススルー）、トークン認証 API およびモニタリングのみが、マルチ コンテキスト モードでサポートされます。詳細については、「マルチ コンテキスト モード」を参照してください。

ハイレベルのアーキテクチャ



一般的な要求フロー

これは、REST PUT/POST/DELETE API 要求のフローです。

- REST クライアントは ASA への SSL 接続を確立します。
- REST クライアントは基本認証ヘッダーと共に API 要求を ASA に送信します。
- ASA HTTP サーバがクライアント要求を検証し、処理します。
- ASA HTTP サーバは TCP チャネルを使用する REST エージェントへの接続を開き、REST エージェントに HTTP 要求を書き込みます。
- ASA HTTP サーバが REST エージェント プロセスからの応答を待ちます。
- REST エージェントは API 要求を処理し、セッションおよびユーザ情報を選択し、ASA の「localhost」ポートでリッスンしている LINA に CLI コマンド要求を呼び出します。REST エージェントは要求にセッションおよびユーザ情報を含めます。
- LINA 管理ハンドラは CLI コマンドを処理し、結果の出力を収集します。
- LINA は CLI コマンド要求に対する応答を REST エージェントに送信します。
- REST エージェントは REST API 要求への応答を準備して ASA の HTTP サーバに送信します。
- ASA の HTTP サーバはクライアントに応答を転送します。サーバは、REST エージェント プロセスから受信した応答を処理しません。

リソース ID

すべてのリソースには固有識別子「オブジェクト ID」があり、これはユーザが割り当てた特定のタイプの一意の自然な名前または一意の複合属性から生成されたハッシュのいずれかになります。CLI には固有識別子（UID）の概念がなく、REST エージェントはステートレスであるため、REST エージェントは別の固有識別子を生成できないことに注意してください。

例：

```
{
  "kind": "object#AccessGroup",
  "selfLink": "https://<asa_ip>/api/access/in/inside",
  "ACLName": "inside_in_acl",
 " direction": "IN",
  "interface": {
    "kind": "objectRef#Interface",
    "refLink": "https://<asa_ip>/api/interfaces/physical/GigabitEthernet0_API_SLASH_1",
    "objectId": "GigabitEthernet0_API_SLASH_1",
    "name": "inside"
  }
}
```

リソース URL：「selfLink」属性

「selfLink」属性は、そのオブジェクトの JSON 定義内で指定されたリソースの完全 URL です。つまり、オブジェクト要素の集合に対しオブジェクト ID の URL を構築することなくダイレクト アクセスが可能です。この属性は、すべてのリソース オブジェクトの JSON 定義で指定されます。

selfLink のオブジェクト ID 部分は、selfLink が JSON 応答またはロケーション ヘッダーのどちらの一部であるかに関係なく URL エンコードされます。

API 要求を受け取ると、二重または混合エンコーディングに対する標準化チェックが、要求 URL で実行されます。URL が二重エンコードの場合は、「400 Bad Request」が返されます。URL が標準化チェックにパスした場合、要求 URL はデコードされ、その後の処理のために送信されます。

注： JSON 応答内のオブジェクト ID は URL エンコードされません。URL が（selfLink を使用するのではなく）JSON 応答からのオブジェクト ID を使用して明示的に構築される場合、適切にオブジェクト ID を URL エンコードした後にこの URL を作成する必要があります。

リソース タイプ：「kind」属性

すべての JSON オブジェクトには、オブジェクトのコンテンツのタイプを示す次の「kind」属性があります。オブジェクトがリストを表す場合、そのオブジェクトは「collection#{type}」という kind 属性になります。それ以外では、次の項で説明するように「object#{type}」、またはプリミティブ kind などのいくつかの形式があります。

例：

kind: collection#accessPolicySet => ACL エントリのリスト
kind: object#networkobject => 「networkobject」タイプのオブジェクト
kind: objectref#networkobject => 「networkobject」タイプのオブジェクトへの参照
kind: IPAddress => 「ipAddress」タイプのプリミティブ リソース

プリミティブ kind

IP アドレス、ネットワーク、FQDN、サービス タイプなどのようないくつかのプリミティブが他のリソース タイプと混合するとき、「kind」を使用して表すことができます。この場合、「kind」は「#」を使用せず直接指定できます。このようなリソースは、非常にシンプルであり、「kind」以外には、値を指定する「value」属性だけが含まれます。

例：

```
{
"kind: "IPv4Address "
"value": "1.1.1.1"
}
```

リソース アソシエーション

その他のリソースは、特定のリソースから参照できます。次の 2 種類の参照があります。

1. 完全な参照オブジェクトがそっくりそのまま存在するインライン オブジェクトを使用する。このアプローチは、ほとんど使用されず、特定の API でのみサポートされます。
2. 別のリソースを参照する最も一般的な方法は、リソース識別子（objectId または refLink）を使用する方法です。

例：

```
{
"kind": "objectref#networkObjectGroup" ,
"refLink": "http://host/api/object/networkObjectGroups/548292" ,
"objectId":"548292"
}
または
{
"kind": "objectref#networkObjectGroup" ,
"refLink": "http://host/api/object/networkObjectGroup/Lab%20Printers" ,
"objectId": "Lab Printers"
}
```

オブジェクト「rangeInfo」

ほとんどのコレクション リソースは「rangeInfo」オブジェクトを含みます。このオブジェクトは、コレクションに含まれている項目の範囲についての詳細を提供します。GET 要求とクエリ API は改ページをサポートし、また定義された最大数よりも多く項目を返すことはありません。したがって、ネットワーク オブジェクトが 20,000 ある場合は、単一のコールですべてを取得することはできません。また、API 要求では、オフセットおよび結果で返す必要のある、そのオフセットの制限を指定できます。この結果には、オフセットの内容、返される制限、項目の総数を指定する「rangeInfo」エントリが常に含まれます。

```
"rangeInfo": {
"offset": "integer",
"limit": "integer",
"total": "integer",
},
```

制限の最大許容値は 100 です。100 以上の項目の REST クライアント クエリがあり、100 以上の項目が使用可能な場合、100 項目だけが返され、「合計」は使用可能な項目数を示します。

REST API 認証

セキュアな HTTPS トランスポートを使用する HTTP 基本認証またはトークン ベース認証がサポートされ、認証は、要求ごとに実行されます。

注：ASA では、認証局（CA）発行済み証明書の使用が推奨されているため、SSL 接続を確立するとき、REST API クライアントが ASA サーバ証明書を検証できます。

モニタリング API を呼び出すには、権限 3 以上が必要です。GET API を呼び出すには、権限 5 以上が必要です。PUT/POST/DELETE 操作を呼び出すには、権限 15 以上が必要です。

REST API 規則

- HTTP PUT 要求は、既存リソースの置換、更新、または変更に使用します。一方、HTTP POST は新しいリソース（または PUT の対象外のアクション）の作成に使用します。リソースを作成するために HTTP PUT を使用しないでください。
- HTTP PUT 要求の要求本文は、リソースの必須属性の完全な表現が含まれている必要があります。
- HTTP PUT は、完全なリソースを受け入れます。これは、応答の更新バージョンを返しません。変更されたリソースが応答で送信されない場合、HTTP ステータス コードは、HTTP ヘッダー応答で 204 になります（OK の 200 ではない）。
- HTTP PATCH は、部分的にリソースを更新する場合にサポートされます。指定されていない属性は、サーバ値の値を取得します。
- HTTP POST 要求には、JSON 形式で作成する新しいリソースの詳細が含まれます。
- Create 要求に対する HTTP POST 応答の HTTP ヘッダーには、戻りコード 201 と新しく作成されたリソースの URI を含む Location ヘッダーがあります。
- 自動作成された設定（リソース）は、作成および削除 REST 操作をサポートしません。つまり、HTTP POST および DELETE 要求はありません。たとえば、ロギング関連の設定を作成または削除はできませんが、変更（PUT）または取得（GET）はできます。
- HTTP GET も HTTP DELETE も要求本文はありません。
- リソースのコレクションの HTTP DELETE は、URL で指定されたリソースを削除してしまうため、サポートされません。そのリソースが削除された場合、サブリソース（コレクションの項目）を作成できません。
- HTTP GET 応答には、オブジェクト名またはオブジェクトのコレクション名を示す「kind」属性があります。
- すべての REST API 要求および応答は JSON 形式である必要があります。
- すべての JSON 属性は、「CamelCase」の命名規則を使用する必要があります。たとえば、「policyType」などです。
- JSON 値の文字列タイプは二重引用符で囲む必要があり、ブール値または数値タイプは二重引用符で囲む必要はありません。ブール値は小文字で true または false です。
- すべての受信 HTTP 要求は、REST 応答が JSON 形式であることを REST クライアントが想定していることを示す、この「Accept: application/json」ステートメントが HTTP ヘッダーにあることを想定しています。

- すべての HTTP POST 要求に JSON の本文（属性）を含める必要があります。
- HTTP 応答の Location ヘッダーはすべての POST（作成）シナリオの完全な URL を含んでいます。
- スキーマの JSON 表現にある [<items>] のようなブラケットは、アイテムのリストを示します。
- 特に指定がない場合、HTTP GET は現在設定されている状態を返します。
- 値がない場合に属性が表示されるかどうかは、それがオプション属性かどうかによって異なります。オプションの場合、HTTP GET 応答で省略されることがあります。オプションでない場合、値は属性が文字列タイプの場合は空白で表示され、数値の場合は 0（ゼロ）で表示されます。
- 改ページはサポートされており、GET またはクエリ API コールによって取得できる項目の最大数を制限します。

REST API コード

HTTP エラー コードは、次の標準に基づいて報告されます。

HTTP ヘッダー内の HTTP エラー コード	説明
400 Bad Request	無効なクエリ パラメータ：認識されないパラメータ、欠落しているパラメータ、または無効な値。
404 Not Found	URL が既存のリソースと一致しない。たとえば、リソースの HTTP DELETE は、リソースが利用できないために失敗する場合があります。
405 Method not Allowed	読み取り専用リソースで、POST などの、許可されない HTTP 動詞。
500 Internal Server Error	サーバ エラー。その他のすべてのエラー：その他の応答コードが使用できない場合の最後の候補です。

エラー コードに加えて、返された応答には、エラーの詳細を提供するエラー オブジェクトを持つ本文が含まれます。

HTTP の成功コードは次の標準に基づいて報告されます。

HTTP ヘッダー内の HTTP の成功コード	説明
200 Success OK	リソースは、GET メソッドを使用して正常に取得されました。
201 Created	リソースは、POST メソッドを使用して正常に作成されました。
204 No Content	リソースは、PUT または PATCH メソッドを使用して正常に更新されました。または正常に削除されました（DELETE）。

JSON のエラーまたは警告の応答スキーマ

```
{
  "level"  : "string",
  "code"   : "string",
  "context": "string",
  "details": "string"
}
```

プロパティ	タイプ	説明
level	文字列	「Error」、「Warning」または「Info」。
code	文字列	「READ-ONLY-FIELD」のような応答コード、または特定の機能へのコード指定。
context	文字列	このエラー、警告、情報の応答が適用される属性の名前。
details	文字列	このエラー、警告、情報の応答の詳細なメッセージ。

ASA の REST API エージェント

ASA REST API エージェントのインストールと有効化

REST API エージェントは、Cisco.com の ASA イメージとは別にパブリッシュされます。つまり、出荷済みの ASA イメージには REST API プラグイン パッケージは含まれません。REST API パッケージはフラッシュにダウンロードされ、「rest-api image」コマンドを使用してインストールされます。REST API エージェントは「rest-api agent」コマンドを使用して有効にします。

マルチ コンテキスト モードでは、REST API エージェント コマンドはシステム コンテキストでのみ使用できます。

注：REST API エージェントは、Java ベースのアプリケーションです。Java Runtime Environment（JRE）は REST API のパッケージに含まれています。

「rest-api image」コマンド

このコマンドは、互換性や検証チェックを実行して、問題があれば通知します。すべてのチェックにパスすると、REST API イメージをインストールします。アンインストールするには、コマンドの「no」形式を使用します。

```
[no] rest-api image disk0:/<package>
```

image：このキーワードを使用して、ASA の REST API イメージをインストールまたはアンインストールします。接続先（この場合は、ASA のフラッシュ メモリの「disk0:」）および REST API イメージ パッケージ名を入力します。

rest-api パッケージをインストールまたは更新した後、ASA はリブートされません。

この構成は、スタートアップ コンフィギュレーション ファイルに保存されます。

例

次に、REST API パッケージを TFTP サーバからダウンロードしてインストールする例を示します。

```
copy tftp://<tftpserver>/asa-restapi-121-lfbff-k8.SPA disk0:
rest-api image disk0:/asa-restapi-121-lfbff-k8.SPA
```

サポート対象モード

シングル コンテキストまたはマルチ コンテキスト、ルーテッドまたはトランスペアレント

REST API エージェントに必要な追加のブートストラップ

- HTTP サーバを有効にして、管理インターフェイスを介してクライアントを接続します。
  http server enable
  http 0.0.0.0 0.0.0.0 <mgmt interface nameif>
- （スタティック）ルートを設定します。
- コマンド許可が有効の場合、特権レベル 15 でローカル ユーザ「enable_1」が使用可能であることを確認します（REST API エージェントはこのアカウントを使用して ASA と通信します）。
  username enable_1 password <pass> encrypted privilege 15

「rest-api agent」コマンド

REST API イメージをインストールした後、REST API エージェントを有効にするには、「rest-api agent」コマンドを使用します。REST API エージェントを無効にするには、このコマンドの「no」形式を使用します。

```
[no] rest-api agent
```

agent：ASA の REST API エージェント プロセスを開始します。

前提条件：HTTP サーバは、REST API エージェントが機能するように有効にする必要があります。

REST API エージェントが有効の場合、「/api」URL 要求が ASA HTTP サーバから REST API エージェントにリダイレクトされます。

サポート対象モード

シングル コンテキストまたはマルチ コンテキスト、ルーテッドまたはトランスペアレント

「show rest-api agent」コマンド

「show rest-api agent」コマンドは、REST API エージェントの現在のステータスを示します。

```
ciscoasa(config)# show rest-api agent
The REST API Agent is currently enabled
```

または

```
ciscoasa(config)# show rest-api agent
The REST API Agent is currently disabled
```

REST API エージェントが無効の場合、「/api」URL 要求は、404 ステータス コードの応答で ASA HTTP サーバによって拒否されます。

サポート対象モード

シングル コンテキストまたはマルチ コンテキスト、ルーテッドまたはトランスペアレント

「show version」コマンド

REST API のバージョンは、「show version」コマンドの出力に表示されます。

```
ciscoasa(config)# show version
Cisco Adaptive Security Appliance Software Version 9.4(1)
REST API Agent Version <version number>
```

REST API エージェント デバッグ

「debug rest-api」コマンドは、CLI 端末の REST API エージェント デバッグ トレースを有効します。

このコマンドを呼び出すと、デバッグ ログを有効にし転送するために、REST API デーモンから REST API エージェントにメッセージをトリガーします。続いて、REST API エージェントは、REST API デーモンに TCP 上でデバッグ ログを転送します。これらのログは CLI セッション中に表示されます。CLI セッションを閉じるとき、または「no debug rest-api」コマンドが発行されるとき、REST API デーモンは、セッションのロギングを無効にするよう REST API エージェントに通知します。

```
debug rest-api [agent | cli | client | daemon | process | token-auth] {event, error}
```

agent：REST API エージェント操作のデバッグ情報。

cli：REST API エージェントとの通信の REST API CLI デーモンのデバッグ情報。

client：REST API クライアントと REST API エージェント間のメッセージのルーティングのデバッグ情報。

daemon：REST API エージェントとの通信の REST API デーモンのデバッグ情報。

process：REST API エージェントの開始または停止処理のデバッグ情報。

token-auth：REST API トークン認証処理のデバッグ情報。

サポート対象モード

シングル コンテキストまたはマルチ コンテキスト、ルーテッドまたはトランスペアレント

show コマンドの出力

「debug rest-api agent is enabled」または「debug rest-api agent is disabled」

「debug rest-api cli is enabled」または「debug rest-api cli is disabled」

「debug rest-api daemon is enabled」または「debug rest-api daemon is disabled」

「debug rest-api http is enabled」または「debug rest-api http is disabled」

「debug rest-api process is enabled」または「debug rest-api process is disabled」

「debug rest-api token-auth is enabled」または「debug rest-api token-auth is disabled」

Syslog

Syslog # 342001

説明/根拠/概要：

REST API エージェントが正常に起動しました。

デフォルト レベル：

7

Syslog 番号と形式：

%ASA-7-342001: REST API Agent started successfully.

説明：

REST API クライアントで ASA を設定するには、その前に REST API エージェントを正常に起動する必要があります。

推奨事項/アクション：

なし

Syslog # 342002

説明/根拠/概要：

REST API エージェントが失敗しました。

デフォルト レベル：

3

Syslog 番号と形式：

%ASA-3-342002: REST API Agent failed, reason: *<reason>*

<reason> The reason why the REST API Agent failed.

説明：

REST API エージェントが、さまざまな理由で起動に失敗したかクラッシュした可能性があります。理由の 1 つとして、REST API エージェントでメモリが不足していることが考えられます。または、REST API エージェントを有効または無効にするために実行されるメッセージングが失敗している可能性もあります。

推奨事項/アクション：

管理者は無効化を試行し（「no rest-api agent」）、続いて「rest-api agent」を使用して REST API エージェントを再度有効にする必要があります。

Syslog # 342003

説明/根拠/概要：

REST API エージェントに障害が発生し、再起動していることを通知します。

デフォルト レベル：

3

Syslog 番号と形式：

%ASA-3-342003: REST API Agent failure notification received. Agent will be restarted automatically.

説明：

REST API エージェントに障害が発生し、エージェントの再起動が試みられています。

推奨事項/アクション：

なし

Syslog # 342004

説明/根拠/概要：

REST API エージェントは複数の試行後、正常に起動できませんでした。

デフォルト レベル：

3

Syslog 番号と形式：

%ASA-3-342004: Failed to automatically restart the REST API Agent after five unsuccessful attempts. Use the 'no rest-api agent' and 'rest-api agent' commands to manually restart the Agent.

説明：

REST API エージェントは連続の試行後、起動に失敗しました。

推奨事項/アクション：

失敗の理由を特定するには、syslog %ASA-3-342002（記録されている場合）を参照してください。REST API エージェントを無効（「no rest-api agent」）にして、再度有効（「rest-api agent」）にしてください。

Syslog # 342005

説明/根拠/概要：

REST API イメージが正常にインストールされました。

デフォルト レベル：

7

Syslog 番号と形式：

%ASA-7-342005: REST API image has been installed successfully.

説明：

REST API イメージは、REST API エージェントを起動する前に正常にインストールする必要があります。

推奨事項/アクション：

なし

Syslog # 342006

説明/根拠/概要：

REST API イメージのインストールに失敗しました。

デフォルト レベル：

3

Syslog 番号と形式：

%ASA-3-342006: Failed to install REST API image, reason: <reason>

<reason> The reason why the REST API Agent installation failed

説明：

REST API イメージが、次の理由でインストールに失敗した可能性があります。

バージョン チェックの失敗|イメージ検証の失敗|イメージ ファイルが見つからない|フラッシュの容量不足|マウント失敗

推奨事項/アクション：

管理者は、障害を修復し、「rest-api image <image>」を使用してイメージを再インストールする必要があります。

Syslog # 342007

説明/根拠/概要：

REST API イメージが正常にアンインストールされました。

デフォルト レベル：

7

Syslog 番号と形式：

%ASA-7-342007: REST API image has been uninstalled successfully.

説明：

新しいイメージをインストールする前に、古い REST API イメージを正常にアンインストールする必要があります。

推奨事項/アクション：

なし

Syslog # 342008

説明/根拠/概要：

REST API イメージのアンインストールに失敗しました。

デフォルト レベル：

3

Syslog 番号と形式：

%ASA-3-342008: Failed to uninstall REST API image, reason: <reason>.

説明：

REST API イメージのアンインストールが、次の理由で失敗した可能性があります。

マウント解除の失敗|rest エージェントが有効になっている

推奨事項/アクション：

管理者は REST API イメージをアンインストールする前に REST エージェントを無効にする必要があります。

アウトオブバンド変更処理

REST API 要求を処理しているときにアウトオブバンド変更が発生した場合は、要求を処理する前に REST API エージェントに設定がリロードされます。

サポートされる ASA 機能

AAA

AAA API は、認証、許可、およびコマンド特権の AAA 関連機能の構成をサポートします。

AAA サーバ グループおよびアカウンティングはまだサポートされていません。

認証

api/aaa/authentication

ネットワーク認証の設定

制限事項：

現在、ローカル サーバ グループのみサポートされます。

認証

api/aaa/authorization

ネットワーク認証の設定

制限事項：

現在、ローカル サーバ グループのみサポートされます。

コマンド特権

api/aaa/commandprivileges

ローカル コマンド特権レベルを設定します。

制限事項：

該当なし

アクセス ルール

/api/access

ルーテッド ファイアウォール モードの場合もトランスペアレント ファイアウォール モードの場合も、ネットワーク アクセスを設定するには、Access API を使用します。

REST API を使用してアクセス グループのアクセス ルールを GET できます。最初のアクセス ルールが特定のインターフェイスおよび方向に対して作成されるとアクセス グループが自動的に作成されます。同様に、最後のアクセス ルールが削除されるとアクセス グループは削除されます。グローバル アクセス ルールもサポートされます。

REST API を使用するとアクセス ルールを GET、POST、PUT、PATCH、DELETE できます。アクセス URI は、インターフェイスと方向ごとにグループ化され、/access の共通 URI ルートがあります。

制限事項：

制限はありません。ASDM アプリケーションと同じ機能をサポートします。

Back Up and Restore

ASA 設定のバックアップにはこの API を使用します。**/api/backup**

ASA 設定の復元にはこの API を使用します。**/api/restore**

制限事項：

該当なし

DHCP

DHCP クライアントと DHCP リレーを設定するにはこれらの API を使用します。**/api/dhcp**

制限事項：

トランスペアレント モードでは DHCP リレーはサポートされていません。

DNS

DNS を設定するにはこれらの API を使用します。**/api/dns**

制限事項：

該当なし

フェールオーバー

/api/failover

制限事項：

該当なし

インターフェイス

インターフェイス関連の設定を入力するために使用できる 6 種類の API があります。物理インターフェイス用の **/api/interfaces/physical**、VLAN インターフェイス用の **/api/interfaces/vlan**、ポート チャネル インターフェイス用の **/api/interfaces/portchannel**、冗長インターフェイス用の **/api/interfaces/redundant**、ブリッジ グループ インターフェイス（BVI）用の **/api/interfaces/bvi** はトランスペアレント モードで使用可能です。また、グローバル インターフェイス設定（**/api/interfaces/setup**）もあります。

制限事項：

該当なし

IP 監査

/api/firewall/ipaudit

制限事項：

該当なし

ライセンス

永久ライセンスおよびアクティベーション キーのライセンス

api/licensing/activation

キー ベースのライセンスを表示、設定するための API。永久ライセンスは、アクティベーション ライセンスと同様に GET を介して取得されます。

制限事項：

ASA は、アクティベーション ライセンスの設定を変更（新しいライセンスの追加およびライセンスの有効化や無効化など）した後、手動でリロードする必要があります。

共有ライセンス

api/licensing/shared

アクティブなライセンスで定義された共有ライセンス（クライアントまたはサーバ共有ライセンス）設定の構成をサポートする API。

制限事項：

該当なし

スマート ライセンス

api/licensing/smart

スマート ライセンスを設定し、サポートされるプラットフォームのエンタイトルメントをモニタする API。

api/licensing/smart/asav/register への POST 要求は、無効なトークン ID でもコード 201（成功）を返すことに注意してください。ASAv 自体は、トークン ID を確認できません。これは検証用のライセンス サーバに依存します。しかし、ライセンス サーバへの呼び出しは、トークン ID が ASAv により受け取られると、非同期的に発行され処理されます。

制限事項：

該当なし

ロギング

syslog サーバ

api/logging/syslogserver

syslog サーバの CRUD 操作をサポートする API。

制限事項：

該当なし

syslog サーバの設定

/api/logging/syslogserversettings

syslog サーバがダウンしたときのロギング キューの設定と TCP ロギングの許可を含む、syslog サーバの詳細設定をサポートする API。

制限事項：

該当なし

syslog メッセージの構成

/api/logging/syslogconfig

レベルおよびメッセージの有効化または無効化を含む、syslog メッセージの詳細の構成をサポートする API。

制限事項：

該当なし

syslog メッセージの設定

/api/logging/syslogconfigsettings

非 EMBLEM 形式、タイムスタンプ、またはクラスタ IP（該当する場合）のデバイス ID を含めるなどの syslog メッセージ設定の構成をサポートする API。

制限事項：

該当なし

Netflow の構成

/api/logging/netflow

Netflow 構成の CRUD 操作をサポートする API。

制限事項：

該当なし

Netflow コレクタの設定

Netflow コレクタ設定の CRUD 操作をサポートする API。

制限事項：

Netflow のサービス ポリシー ルールはサポートされていません

管理アクセス

汎用管理アクセス

api/mgmtaccess

Telnet、SSH、および HTTPS（ASDM）に関連する ASA アクセス設定を構成するためにこの API を使用します。

制限事項：

該当なし

ホスト

/api/mgmtaccess/hosts

Telnet、SSH、および HTTPS（ASDM）接続の管理アクセス ホストの CRUD 操作を可能にします。

制限事項：

該当なし

モニタリング

/api/monitoring/

これらの API は、稼働状態、パフォーマンス、および REST API のモニタリング統計情報を取得するのに使用できます。

マルチ コンテキスト モードでは、システム コンテキストを含む特定のコンテキストのモニタリング統計情報を取得するには https: //<asa_admin_context_ip>/api/cli?context=<context_name> という「context」パラメータを使用してクエリを追加します。「context」クエリ パラメータがモニタリング要求にない場合、REST API エージェントは、独自のターゲット コンテキストを決定します。CPU プロセス使用率などのシステム コンテキストでのみ使用できるリソースでは、要求はシステム コンテキストに転送されます。他のコマンドは、管理コンテキストに転送されます。

制限事項：

該当なし

マルチ コンテキスト モード

マルチ コンテキスト モードのサポートは、汎用 CLI コマンド Executer API、トークン認証 API とモニタリングに制限されます。現在、REST API は、CLI コマンド Executer API 経由を除き、マルチ コンテキスト モードの ASA の構成をサポートしていません。

注：

- REST API エージェントは、マルチ コンテキスト モードで有効にすることができます。REST API エージェント CLI はシステム コンテキストのみに提供されています。
- トークン認証を使用する場合、REST API コマンドを実行する前に https://<asa_admin_context_ip>/api/tokenservices 経由で認証トークンを取得する必要があります。

  管理コンテキストで受信したトークンは、その他のコンテキストの設定やモニタにも使用できることに注意してください。
- 汎用 CLI コマンド Executer API は、https://<asa_admin_context_ip>/api/cli?context=<context_name> としてコンテキストを設定するのに使用できます。「context」クエリ パラメータがない場合、要求は管理コンテキストに転送されます。
- 「context」クエリ パラメータがモニタリング要求にない場合、REST API エージェントは、独自のターゲット コンテキストを決定します。CPU プロセス使用率などのシステム コンテキストでのみ使用できるリソースでは、要求はシステム コンテキストに転送されます。他のコマンドは、管理コンテキストに転送されます。

制限事項：

REST API コマンドはシステム コンテキストでのみ使用できます。REST API エージェントは、ASA がシングル コンテキスト モードからマルチ コンテキスト モードに切り替わる、またはその逆に切り替わるときに、再起動する必要があります。

NTP

/api/devicesetup/ntp/

制限事項：

該当なし

NAT

/api/nat

NAT API は、TwiceNAT（別名手動 NAT）および ObjectNAT（別名 AutoNAT）をサポートします。各 NAT タイプに一意の URI があります。Before AutoNAT と After AutoNAT は、完全にサポートされます（ルーテッドおよびトランスペアレント モード）。

InterfacePAT、DynamicPAT（非表示）、PAT プールを設定する属性も、API に含まれます。

同じリスト内のすべての NAT タイプ（Twice および Auto）を示す単一のリストはサポートされていません。

ObjectNAT（AutoNAT）

制限事項：

NAT ルールでのインライン ネットワーク オブジェクトの作成はサポートされません。既存のネットワーク オブジェクトのオブジェクト NAT を作成するには、変換されるネットワーク オブジェクトを送信元アドレスに指定する必要があります。

TwiceNAT（手動 NAT）

Before NAT と After NAT は、2 つのリストに分割され、独自の URI が指定されます。Before NAT ルールの After NAT ルールへの移動、またはその逆の移動はサポートされません。

制限事項：

該当なし

オブジェクト

/api/objects/

オブジェクトは再利用可能な設定コンポーネントです。オブジェクトは、ASA コンフィギュレーションの中で定義して、インライン IP アドレス、サービス、名前などの代わりに使用できます。REST API は次のオブジェクト タイプをサポートしています。

- 拡張 ACL。アクセス ルールと同様、拡張 ACL は、最初の ACE が作成されたときに作成され、最後の ACE が削除されたときに削除されます。
- ローカル ユーザとユーザ グループ。
- ネットワーク オブジェクトとオブジェクト グループ。

- （事前定義されたネットワーク サービスを含む）ネットワーク サービスとサーバ グループ。事前定義されたサービス オブジェクトは、変更または削除することはできません。事前定義されたサービス オブジェクトは、インライン サービスをカット アンド ペーストするため、またはサービス オブジェクトを作成するときに使用できます。
- 正規表現。
- セキュリティ オブジェクト グループ。
- 時間範囲。
- ユーザ オブジェクト。

ASDM と同様に、REST API は、アクセス時のインライン オブジェクトおよびオブジェクト グループの使用、NAT およびサービス ポリシー ルールの使用をサポートします。

制限事項：

ローカル ユーザのみがサポートされます。

プロトコルのタイムアウト

/api/firewall/timeouts

グローバル プロトコルおよびセッション タイムアウトを設定する API。

制限事項：

該当なし

ルーティング

/api/routing/static

スタティック ルートのみ、現在サポートされています。

制限事項：

該当なし

サービス ポリシー

/api/servicepolicy/

REST API は次のプロトコル インスペクションをサポートします。

DCERPC
DNS over UDP
FTP
HTTP
ICMP
ICMP ERROR
IP Options

NetBIOS
RTSP
SIP
SQL*Net

正規表現と接続制限は別のリソース URI としてサポートされます。

制限事項：

該当なし

VPN

/api/vpn/

サイト間 VPN 設定のみ REST API でサポートされます。IPv4 および IPv6 の両方がサポートされます。サイト間 VPN のモニタリングはサポートされません。

制限事項：

サイト間の設定のみサポートされます。ASDM に表示される証明書管理はサポートされていません。

特別 API

一括 API

便宜上、この API は、さまざまなリソースに対する複数の POST、PUT、PATCH および DELETE 要求を単一の HTTP POST コールにグループ化します。つまり、1 つの要求で複数のリソースを変更できます。含まれている各要求は、ペイロード内の出現順に処理されます。ただし、一括要求の内容は、アトミックな設定変更として処理されることに注意してください。そのいずれかの要求が失敗した場合、全ペイロードが拒否され、ASA 設定の変更は行なわれません。

要求のペイロードと応答構造の詳細は次のとおりです。

POST URL: /api

Request payload format: [{}, {}, {}, ...] where each JSON object is an operation wrapper:

```
{
 method:<HTTP_REQUEST_METHOD_FOR_RESOURCE >,
 resourceUri:<RESOURCE_URI>,
 data:<POST_CONTENT_FOR_THIS_URI_IF_APPLICABLE>
}
```

プロパティ	タイプ	説明
method	string	「GET」、「POST」、「DELETE」、「PATCH」コールがサポートされています。
resourceUri	string	要求が単独で行われている場合のリソース URI。
data	string	要求が個別に行われている場合、raw 本文として送信される JSON データ。「DELETE」メソッドでは必要ではありません。

一括要求応答形式は次のとおりです。

```
{
entryMessages:[{}, {}, ...],
commonMessages: []
}
```

entryMessages はオブジェクト アレイで、各オブジェクトは一括要求エントリに対応します。

汎用 CLI コマンド Executer API

この特別な API は、シングルラインまたはマルチラインの CLI コマンドを使用して、API 応答として CLI 出力を表示できます。

POST URL: **/api/cli**

要求ペイロードの形式：

```
{
"commands": ["command-1", "command-2",…, "command-n"]
}
```

応答の形式：

```
{
"response": ["command-1 response", "command-2 response",…, "command-n response"]
}
```

制限事項

デバッグ コマンドは、CLI パススルーではサポートされません。デバッグ コマンドは、すべてターミナル セッションごとであり、グローバル コンフィギュレーションではありません。デバッグ コマンドは、CLI パススルーを介して送信され、エラー応答か成功応答かのどちらかを返しますが、デバイスには影響しません。

トークン認証 API

REST API クライアントは、基本認証ヘッダーのユーザ情報と共に「/api/tokenservices」に POST 要求を送信し、そのユーザのトークンを取得する必要があります。続いて、REST API クライアントは、後続の REST AP コールの「X-Auth-Token」要求ヘッダーでこのトークンを使用できます。「トークン」は、基本認証ヘッダーのユーザー情報を使用して、「DELETE /api/tokenservices/<token>」要求で明示的に無効化されるまで、またはセッションがタイムアウトになるまで有効です。

POST URL: /api/tokenservices

要求ペイロードは空です。ユーザ情報は、基本認証ヘッダー内にあります。

応答は次のとおりです。

理由	HTTP ステータス コード
AAA 検証エラーまたは許可ヘッダーが存在しません。	401 Unauthorized
認証成功。	204 No Content + X-Auth-Token <token id> (header)
ヘッダーからユーザ名またはパスワードを取得できません。またはその他の健全性チェックが失敗しました。	400 Bad Request
最大セッション数に到達しました。 **注：**コンテキストごとの最大セッション数は 25 です。	503 Service unavailable

トークンを削除するには、DELETE を /api/tokenservices/<token> という URL に発行します。

要求ペイロードは空です。ユーザ情報は、基本認証ヘッダー内にあります。

応答は次のとおりです。

理由	HTTP ステータス コード
AAA 検証エラーまたは無効なトークン。	401 Unauthorized
成功。	204 No Content
ヘッダーからユーザ名またはパスワードを取得できません。またはその他の健全性チェックが失敗しました。	400 Bad Request.

注：

- 既存の syslog 605004 および 605005 がトークンの作成または削除に使用されます。
- 既存の syslog 109033 は、認証サーバが「Challenge」を要求した場合にそれがサポートされていないことをユーザに通知するために使用されます。

- REST API 要求を受信すると、「X-Auth-Token」ヘッダーが最初に確認されます。存在しない場合は、サーバが基本認証にフォール バックします。
- トークン認証は Oauth 2.0 の RFC 6749 仕様に準拠していません。
- 生成されたトークン データベースは、ASA のメモリ上にあり、フェールオーバー ペアまたはクラスタ間で複製されません。つまり、フェールオーバーがフェールオーバー ペア内で発生した場合、またはクラスタ マスター デバイスが変更された場合、認証を再実行する必要があります。
- マルチ コンテキスト デバイスでは、トークンは管理コンテキスト用に受信されると同時に、その他のコンテキストを設定するためにも使用できます。

メモリ書き込み API

REST API コールによって ASA の設定に加えられた変更は、起動設定には反映されません。つまり、変更は実行時設定だけに割り当てられます。この「メモリ書き込み API」は、起動設定に現在の実行時設定を保存するために使用することができます。

POST URL: /api/commands/writemem

要求ペイロードは空です。

REST API のオンライン マニュアル

オンライン マニュアルのインターフェイス（「ドキュメント UI」）では、ユーザ インターフェイス機能と、内蔵 API マニュアルに含まれているすべての情報が統合されています。ドキュメント UI は、次のブラウザのいずれかで実行できます。Chrome（最新版）、Firefox（最新版）、Internet Explorer 9+、Safari 5.1+、Opera（最新版）。旧バージョンでも動作しますが、Internet Explorer 8 以下では動作しません。

REST API エージェントは、ドキュメント UI にアクセスできるように設定する必要があります。ドキュメント UI は、`https://<asa management interface ip>/doc/` からアクセスできます（最後の「/」はドキュメント UI にアクセスするために必要です）。

注：ローカルの REST API のマニュアル ページにアクセスすると、ブラウザはそのページの ASA に要求を送信し、さまざまな Web サイトの特定の jQuery および JSON ファイルも要求します。これらの場所の 1 つが、`https://cdnjs.cloudflare.com` です。

ただし、FirePOWER サービスを有効にして ASA を通過したとき、「Categories: ad portal」ブロッキング フィルタが設定されていると、その要求が FirePOWER モジュールによりブロックされる可能性があります。cloudflare サイトをブロック解除するには、明示的にこのサイトを許可するアクセス コントロール ルールを作成し、「ad portals」をブロックするアプリケーション条件を含むルールの上位にそのルールを配置します。

URL フィルタリングやアプリケーション制御により Web サイトおよび Web アプリケーションがブロックされないようにする方法については、http://www.cisco.com/c/en/us/support/docs/security/firesight-management-center/117956-technote-sourcefire-00.html#anc9 を参照してください。

スクリプトの種類

JavaScript、Python、Perl の 3 種類のスクリプトは、ドキュメント UI から生成できるため、REST API オペレーションを自動化できます。

生成されたスクリプトを使用するための前提条件

JavaScript のスクリプトには、http://nodejs.org/ にある Node.js のインストールが必要です。Node.js により、（Python または Perl のような）コマンドライン スクリプトなどのブラウザ向けに通常記述された JavaScript アプリケーションを使用できます。インストール手順に従った後、次のように目的のスクリプトを実行します。

```
node script.js
```

Python スクリプトでは、https://www.python.org/ にある Python をインストールする必要があります。Python をインストールすると、次のスクリプトが実行できます。

```
python script.py <username> <password>
```

Perl スクリプトは、いくつかの追加設定が必要です。Perl 自体、4 つの Perl ライブラリの 5 つのコンポーネントが必要です。

Perl パッケージは、http://www.perl.org/ にあります。

Bundle::CPAN は、http://search.cpan.org/~andk/Bundle-CPAN-1.861/CPAN.pm にあります。

REST::Client は、http://search.cpan.org/~mcrawfor/REST-Client-88/lib/REST/Client.pm にあります。

MIME::Base64 は、http://perldoc.perl.org/MIME/Base64.html にあります。

JSON は、http://search.cpan.org/~makamaka/JSON-2.90/lib/JSON.pm にあります。

次に示すのは、Macintosh に Perl をインストールする例です。

```
Boot strapping for MAC:
$ sudo perl -MCPAN e shell
cpan> install Bundle::CPAN
cpan> install REST:: Client
cpan> install MIME::Base64
cpan> install JSON
```

依存関係をインストール後に、次のスクリプトを実行できます。

```
perl script.pl <username> <password>
```

法的情報

このマニュアルに記載されている仕様および製品に関する情報は、予告なしに変更されることがあります。このマニュアルに記載されている表現、情報、および推奨事項は、すべて正確であると考えていますが、明示的であれ黙示的であれ、一切の保証の責任を負わないものとします。このマニュアルに記載されている製品の使用は、すべてユーザ側の責任になります。

対象製品のソフトウェア ライセンスおよび限定保証は、製品に添付された『Information Packet』に記載されています。添付されていない場合には、代理店にご連絡ください。

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright 1981, Regents of the University of California.

ここに記載されている他のいかなる保証にもよらず、各社のすべてのマニュアルおよびソフトウェアは、障害も含めて「現状のまま」として提供されます。シスコおよびこれら各社は、商品性の保証、特定目的への準拠の保証、および権利を侵害しないことに関する保証、あるいは取引過程、使用、取引慣行によって発生する保証をはじめとする、明示されたまたは黙示された一切の保証の責任を負わないものとします。

いかなる場合においても、シスコおよびその供給者は、このマニュアルの使用または使用できないことによって発生する利益の損失やデータの損傷をはじめとする、間接的、派生的、偶発的、あるいは特殊な損害について、あらゆる可能性がシスコまたはその供給者に知らされていても、それらに対する責任を一切負わないものとします。

このマニュアルで使用している IP アドレスおよび電話番号は、実際のアドレスおよび電話番号を示すものではありません。マニュアル内の例、コマンド出力、ネットワーク トポロジ図、およびその他の図は、説明のみを目的として使用されています。説明の中に実際のアドレスおよび電話番号が使用されていたとしても、それは意図的なものではなく、偶然の一致によるものです。

ハード コピーおよびソフト コピーの複製は公式版とみなされません。最新版はオンライン版を参照してください。

シスコは世界各国 200 箇所にオフィスを開設しています。各オフィスの住所、電話番号、FAX 番号は当社の Web サイト（www.cisco.com/go/offices）をご覧ください。

シスコの商標

Cisco および Cisco ロゴは、シスコや米国および他の国の関連会社の商標です。シスコの商標の一覧は、www.cisco.com/go/trademarks に掲載されています。掲載されている第三者の商標はそれぞれの権利者の財産です。「パートナー」または「partner」という用語は、シスコと他社との間のパートナーシップ関係を意味するものではありません。(1110R)