自分のコンピュータを設定する方法

コーディング 101(初級 1):
REST API の基本

このラボでは、REST API の利用方法の基本と、Postman を使用して REST API のテストを行う方法について学びます。

目的

所用時間:20 分

- REST API の使用方法の基本を理解する
- Postman REST クライアントを使って API コールする方法について学ぶ

前提条件

このラボでは、REST API コールを発信する Google Chrome ブラウザのアプリケーションである Postman を使用します。

DevNet Zone のステーションにいる場合は、すでに Postman がインストールされています。Google App Launcher から起動するか、Chrome からここをクリックして Launch App を実行し、起動させることができます。

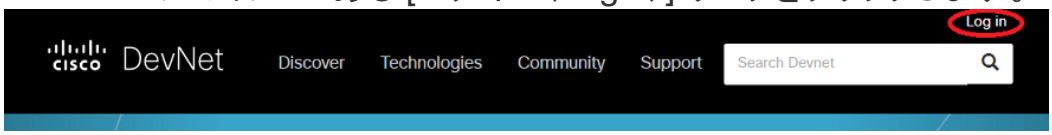
APIC-EM コントローラへのアクセス

- これらのサンプル コードを実行するためには、APIC-EM コントローラにアクセスできる必要があります。
- **自分の APIC-EM コントローラを使用しない場合は、**Always-On APIC-EM ラボ(https://sandboxapic.cisco.com/)を利用できます。Always-On のサンドボックスのログイン クレデンシャルは、**ユーザ名**:devnetuser、**パスワード**:Cisco123! です。詳細は別途示します。

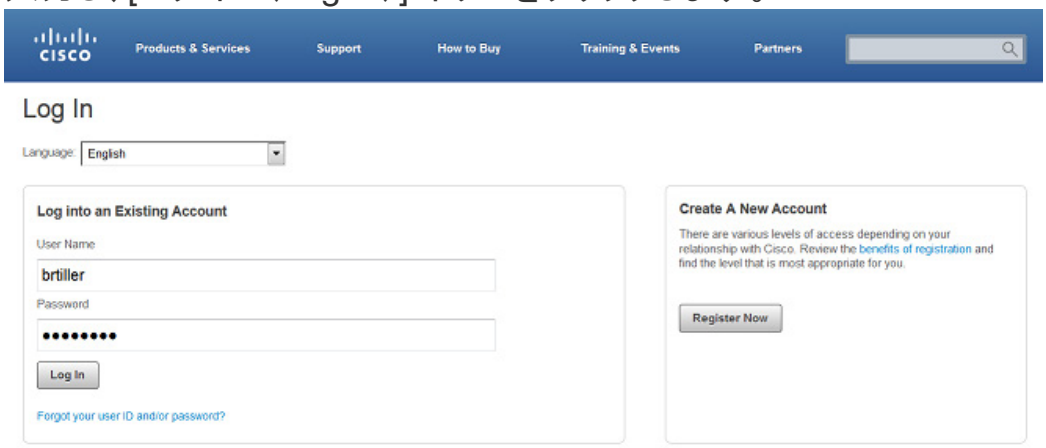
ステップ 1：APIC-EM API リソースの確認

このラボでは、REST API の例として APIC-EM API を使用します。始める前に、APIC-EM リソースを DevNet で確認しておく必要があります。

1. ブラウザで DevNet にアクセスします
   - Web ページの右上にある [ログイン（Login）] リンクをクリックします。

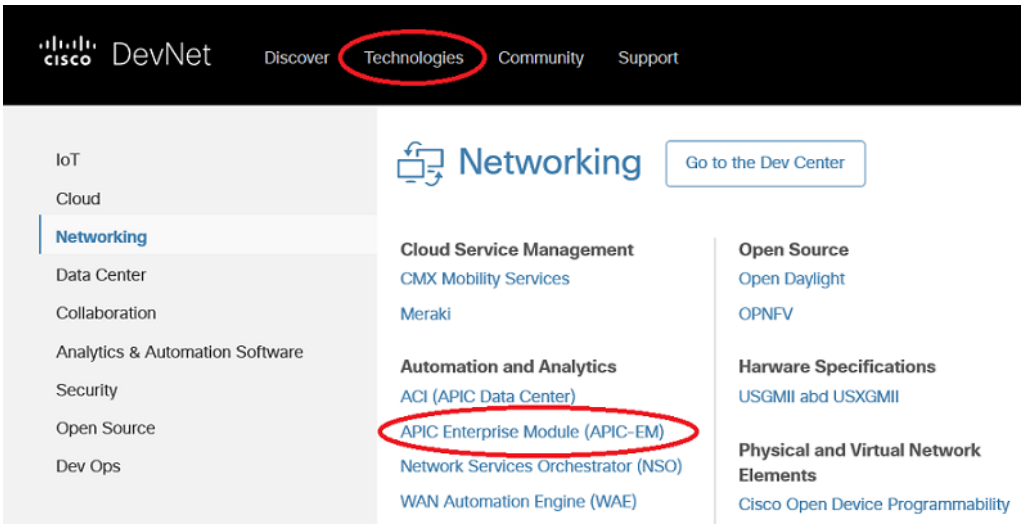


   - ログイン Web ページで、自分のログイン クレデンシャル（CCO ID）を入力し、[ログイン（Log In）] ボタンをクリックします。

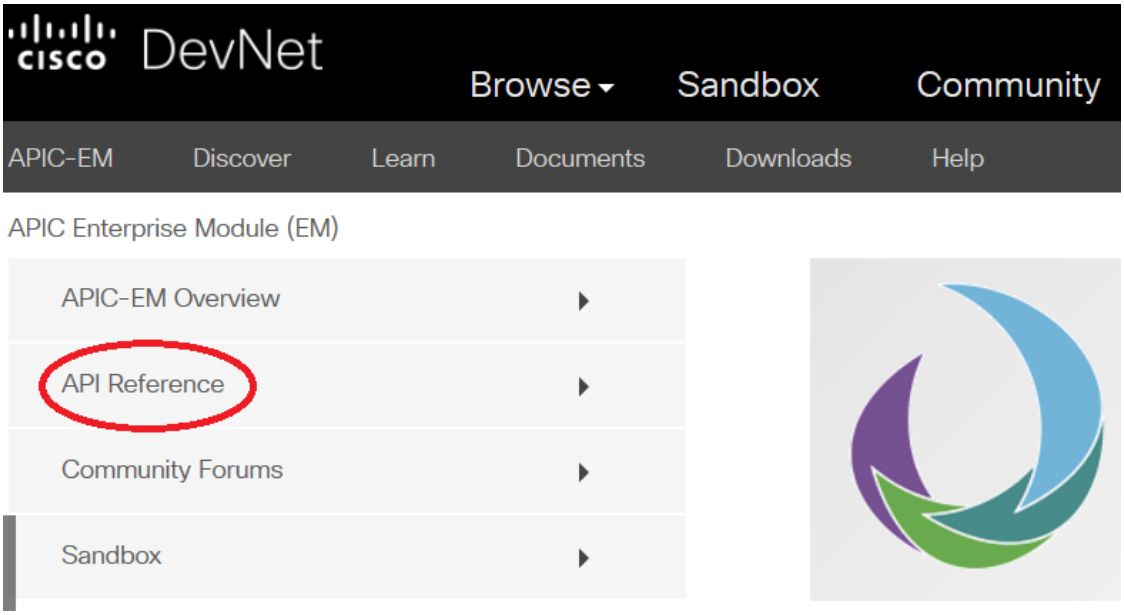


     - Cisco Connection Online ID（CCO ID）を持っていない場合は、[今すぐ登録（Register Now）] ボタンをクリックし、指示に従います。ログイン Web ページへ戻ってきたら、クレデンシャルを入力します。

2. APIC-EM の開発者用リソースに直接移動するには、上部のメニューを使用します。
   - [テクノロジー（Technologies）] メニューにアクセスするには、[テクノロジー（Technologies）] リンクをクリックします。メニューで、[ネットワーキング（Networking）] をクリックした後、[APIC-EM] をクリックします。



3. これで、APIC-EM の Web ポータルにログインできました。ラボでの実習中には、API リファレンス ドキュメントは、別のタブで開いてください。



ステップ 2：REST Web サービスとは

Web サービスとは

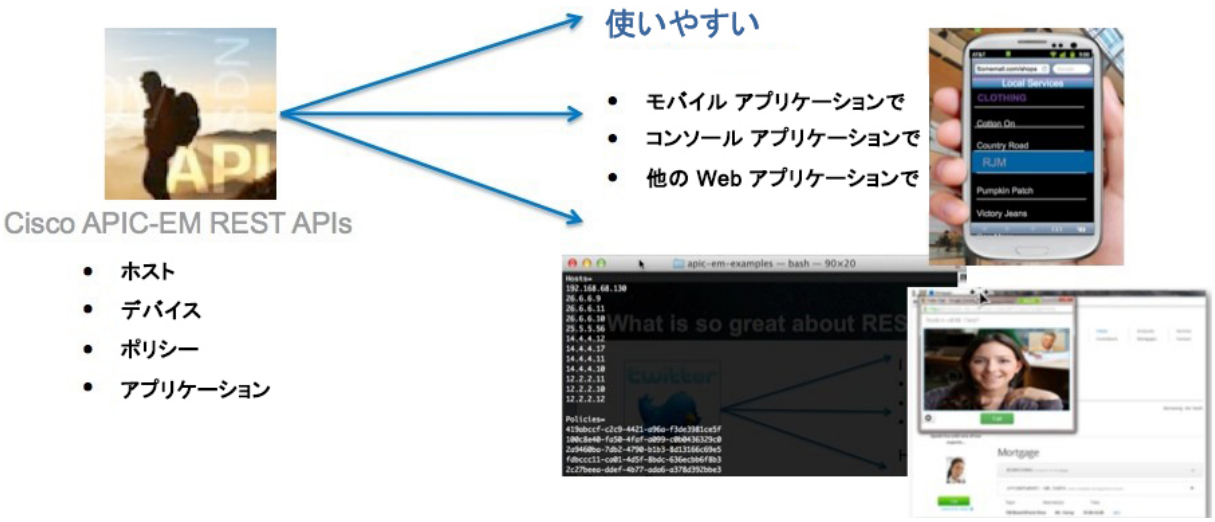
Web サービスとは、定義されたインターフェイスを介して 2 つのシステム間で情報をやり取りする際の方式です。Web サービスの主な 2 つの種類は、**REST** と **SOAP** です。

REST Web サービスとは

REST は、ネットワーク アプリケーションを設計する際のアーキテクチャ スタイルです。REST Web サービスは、HTTP リクエストのように簡単にコールできる Web サービスです。RESTful インターフェイスでは、通常、CRUD（作成、更新、削除）の処理が提供されます。さらに詳細を知りたい場合は、REST チュートリアル [英語] を参照してください。

REST のメリット

REST はプラットフォームを問わず簡単に使用できます。



APIC-EM API は REST API です。

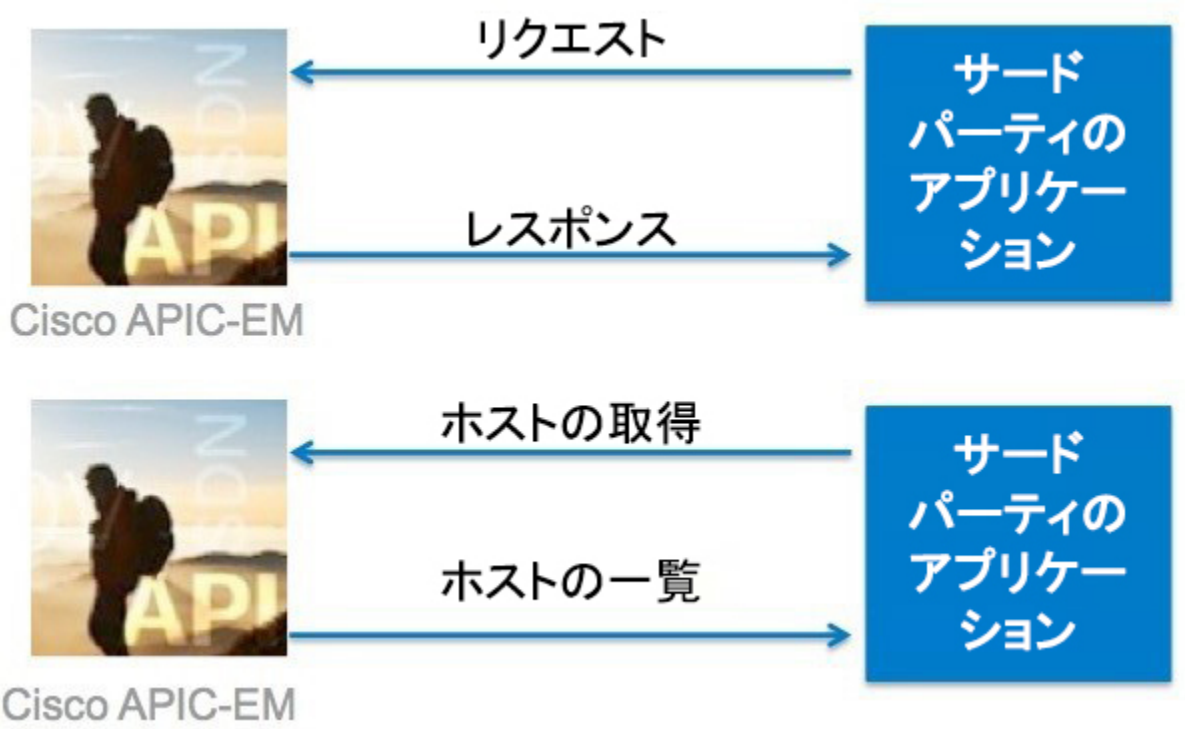
このラボでは、APIC-EM API を使用します。

Application Policy Infrastructure Control（APIC）エンタープライズ モジュール（EM）アプリケーション プログラミング インターフェイス（API）（APIC-EM API）により、ネットワーク インフラストラクチャ全体にアプリケーション ポリシーを導入し、実行することができます。

APIC-EM API を使用することで、ホストの一覧、ネットワーク デバイスの一覧、ユーザの一覧など、ネットワーク上のデバイスに関する情報を取得できます。Python から REST をコールする方法を学ぶための例としてこれらの機能を使用します。

APIC-EM のすべての機能の詳細については、API リファレンス ドキュメント [英語] を参照してください。

REST の仕組み



REST は、HTTP のリクエスト/レスポンス モデルの中核となるものです。API の実行は、HTTP リクエストのようにシンプルです。

この例では、ホストの一覧をリクエストし、その情報がレスポンスで返ります。レスポンスで返されるデータは、通常は JSON か XML です。

（JSON：JavaScript Object Notation は、人が判読可能な形式でデータ交換できるように設計された、テキスト ベースの軽量なオープン スタンダードです。）

ステップ 3：リクエスト作成のために必要なこと

リクエストを作成するには、コールする API に関して、次の情報が必要です。この情報は、API リファレンス ドキュメントで確認できます。

- **メソッド**
    - GET：データを取得します。
    - POST：新しいデータを作成します。
    - PUT：データを更新します。
    - DELETE：データを削除します。
- **URL**
    - コールするエンドポイントの URL
    - 例：
        - {APIC-EMController} の部分は、使用しているコントローラの IP もしくは DNS で引ける名前に置き換えられます。
    - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com/）を使用します。
- **URL パラメータ**
    - URL の一部として渡すパラメータのことです。
- **認証**
    - 使用する認証の種類を知る必要があります。Basic HTTP、トークンベース、OAuth が一般的な種類です。
    - 認証のクレデンシャル
- **カスタム ヘッダー**
    - API が HTTP ヘッダーを送信する必要があるかを確認します。
    - 例：Content-Type：application/json
- **リクエスト ボディ**
    - リクエストの処理に必要なデータを含んだ JSON や XML は、リクエストのボディで送信することができます。

認証について

REST API の認証には、一般的に 3 つの種類があります。認証は、REST API へのアクセス制御とアクセス権限に使用されます。たとえば、読み取り専用のアクセス権限を割り当てられているユーザの場合、データを読み取る API だけをコールできることになります。データの読み取りと変更（追加、編集、削除）の両方が許可されているユーザの場合は、すべての API をコールできます。これらのアクセス権限は、通常、ユーザに割り当てられたロールに基づいています。ロールには、ユーザがデータ変更に関するすべての権限をもつ管理者ロールや、読み取り専用のアクセス権限をもつユーザ ロールなどがあります。

1. **Basic HTTP**：ユーザ名とパスワードがエンコードされた文字列でサーバに渡されます。詳細については、Basic 認証 [英語] を参照してください。
2. **OAuth**：HTTP 認証およびセッション管理のオープン スタンダード。特定のユーザに関連付けられたアクセス トークンを作成し、ユーザ権限も指定します。アクセスと制御の確認に API コールを実行する際、ユーザと権限を特定するためにトークンが使用されます。詳細については、OAuth [英語] を参照してください。
3. **トークン**：OAuth と同様にトークンが生成され、API コールごとに渡されます。ただし、サーバとクライアントとのやりとりをシンプルにするため、クライアントのセッションの管理や追跡は行われません。APIC-EM は認証管理にこの設計を使用します。詳細については、トークン ベース認証 [英語] を参照してください。

APIC-EM はトークン ベースの認証を使用します。したがって、最初に作成する必要のあるリクエストは、APIC-EM でサービス チケットと呼ばれるトークンを作成するリクエストです。サービス チケットは、API へのアクセスの許可と制御の両方に使用され、チケット作成のために実行されるコール以外のすべての API コールに必要になります。

APIC-EM の認証トークンを使用する手順は次のとおりです。

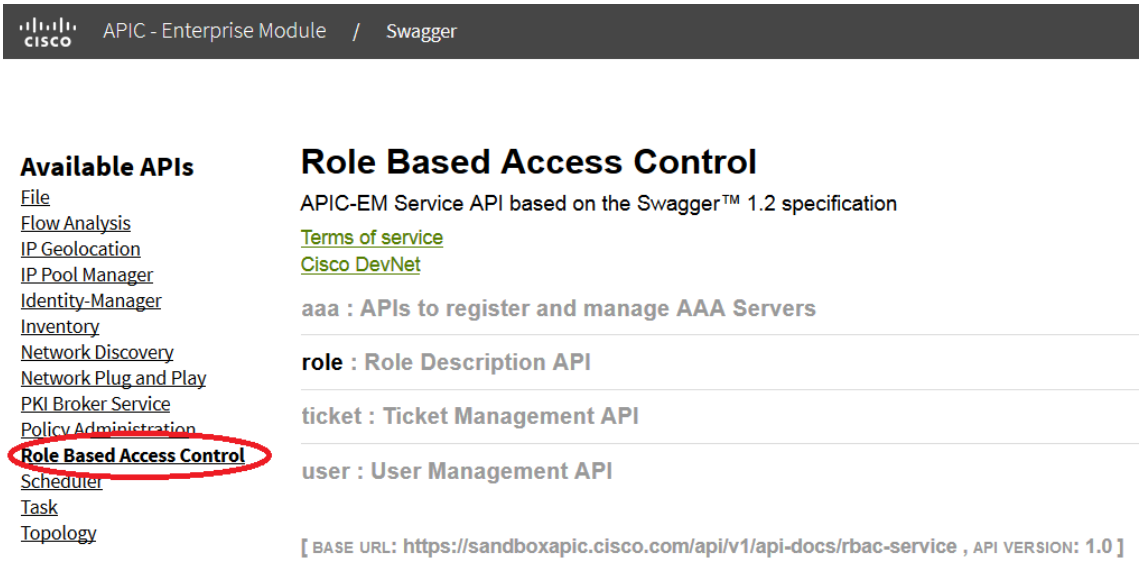
1. チケットを作成します。
2. チケット（トークン）がレスポンスのボディで返されます。
3. これ以降のすべてのリクエストの「X-Auth-Token」ヘッダーに、このトークンを含めます。

API リファレンス ドキュメントの使用

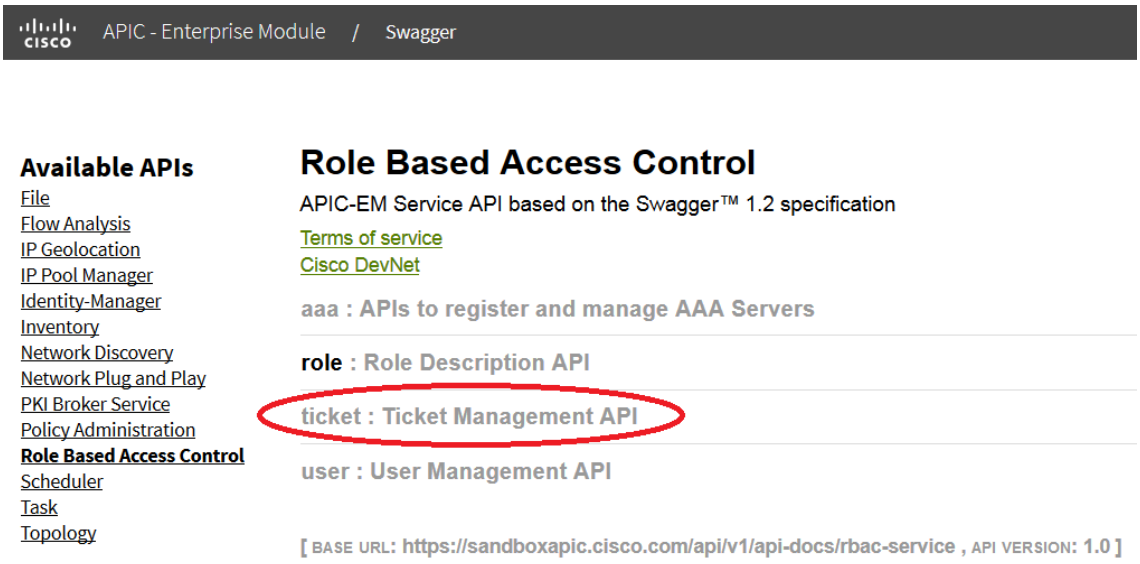
API リファレンス ドキュメントでは、すべての API メソッドの一覧と、各リクエストの作成方法の詳細について確認できます。新しい API を使用する際には、API リファレンスが最も重要な情報源の 1 つとなります。

ではここで、APIC-EM のチケット API について確認してみましょう。

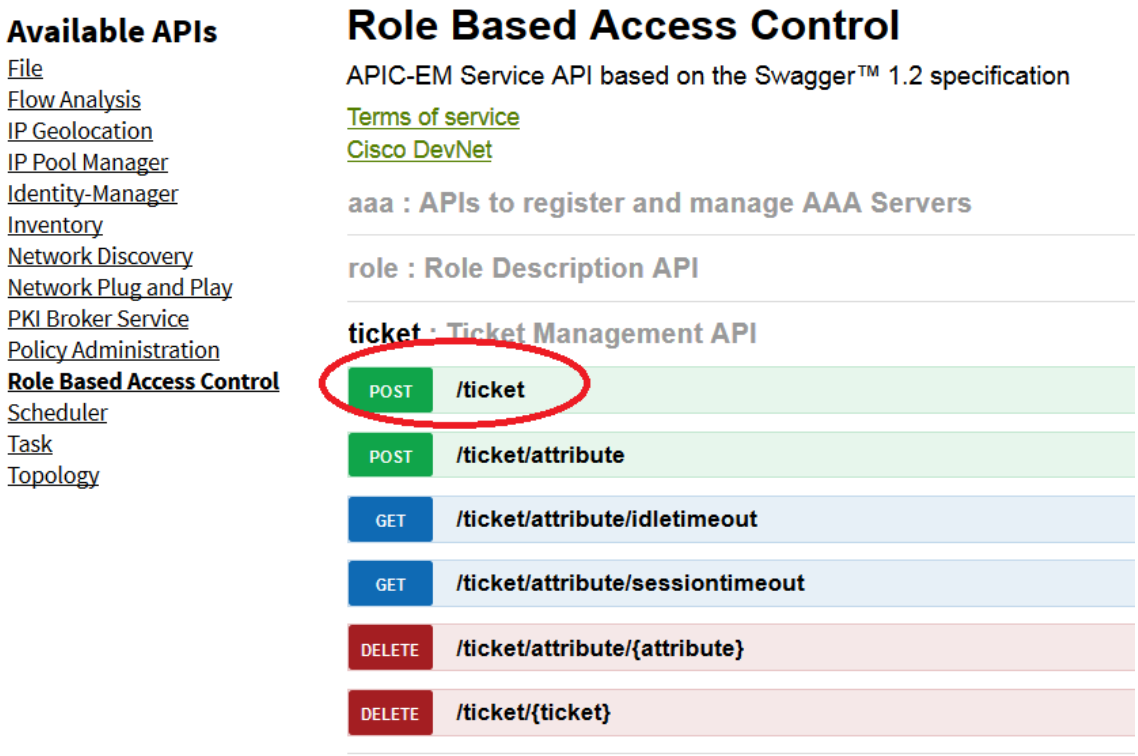
- APIC-EM リファレンス ドキュメントを開くと、すべての API がスクロール可能な形式で表示されます。



- [ロール ベース アクセス制御（Roll Based Access Control）] をクリックし、[チケット（ticket）] をクリックします。



- [チケット（ticket）] で [POST] の [/ticket] をクリックします。この API がチケット作成機能を持っています。

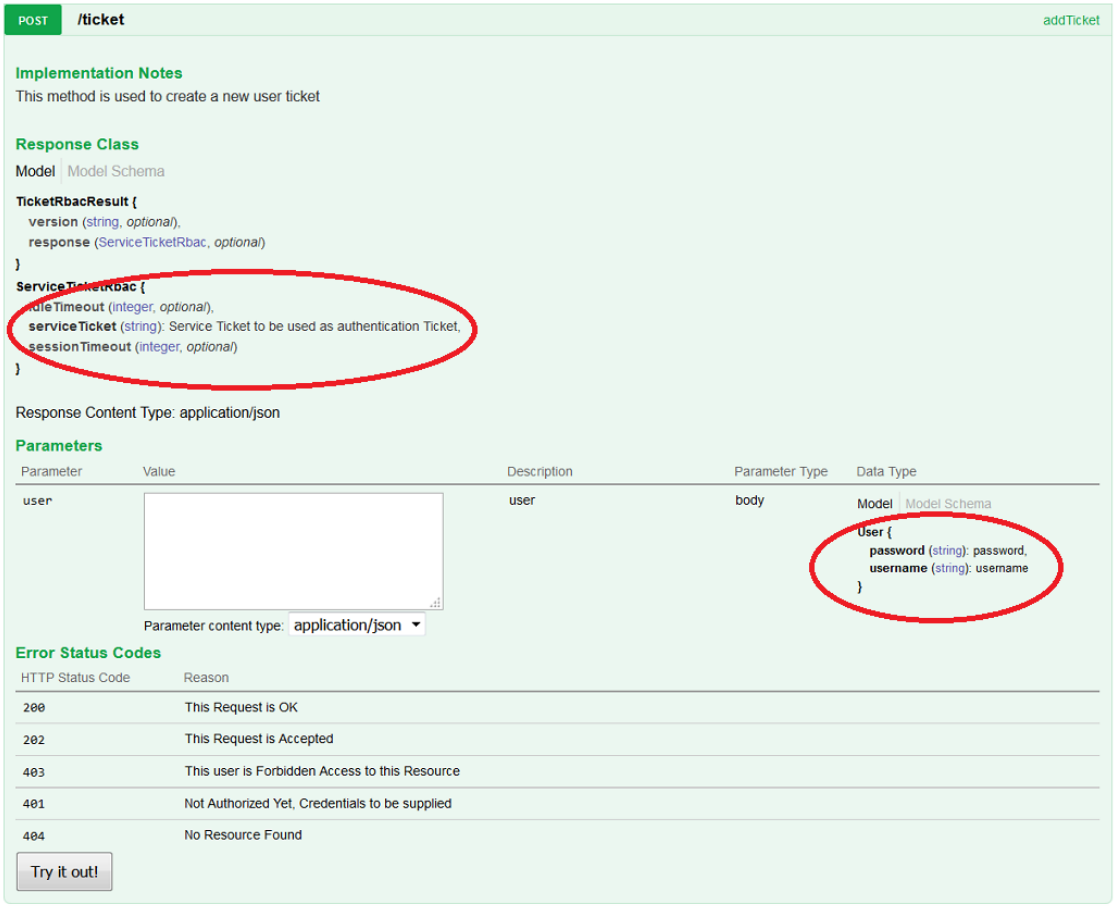


レスポンスの内容

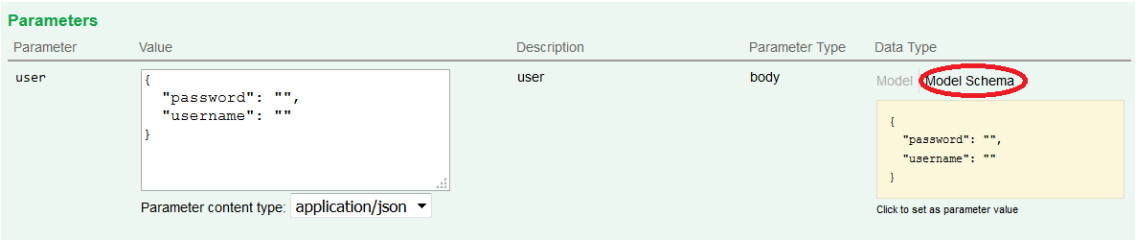
API リファレンスには、送受信される情報のデータ属性が含まれています。受信するデータはレスポンスの部分で定義されており、データ形式や属性とともに HTTP ステータス コードが含まれています。

- HTTP ステータス コード
  - HTTP ステータス コードは、成功、失敗、または他のステータスを返すために使用されます。
    http://www.w3.org/Protocols/HTTP/HTRESP.html
  - 次のような例が一般的です。
    - 200 OK
    - 202 Accepted/Processing
    - 401 Not Authorized

- 内容を確認し、サーバに送信されるデータやアプリケーションに返されるデータを把握します。両方のデータを丸で囲っています。チケット作成リクエストのレスポンスの例では、レスポンスの文字列に、文字列型の値として「serviceTicket」属性が含まれていることがわかります。これが、他のすべての API エンドポイントへの API コールの作成に必要なトークンの値になります。



- [パラメータ（Parameters）] セクションで、[モデル スキーマ（Model Schema）] をクリックします。API を試す場合は、このボックスをクリックして、パラメータを編集可能なボックスに移します。



確認事項

- API リファレンス ガイドの他のチケット API をチェックし、どこが違うかを確認します。

ステップ 4：REST の実行：実際に試す

APIC-EM の例：POST でのチケットの作成

API を使用して認証用チケットを作成する方法について詳しく説明します。

リクエストを作成してサービス チケットを取得する方法を知るためには、API のドキュメントを使用して次の内容を決定する必要があります。

- メソッド
- URL
- ヘッダー
- 認証
- ボディ

API-EM 用の URL はすべて次のように始まります。

```
https://{APIC-EM-Server}/api/v1
```

チケットの API コールの説明については、APIC-EM API ドキュメント [英語] を参照してください。

`POST /api/v1/ticket`：新しいユーザ チケットの作成には、このメソッドが使用されます

HTTP コールでのサービス チケットの作成は非常に簡単です。

```
https://{APIC-EM-Server}/api/v1/ticket
```

サービス チケットを入手するリクエストは、次のようになります。

- **メソッド**：POST
- **URL**：
    - アドレス部分の {APIC-EM-Server} は、使用する APIC-EM のアドレスで置き換えます。
        - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com/）を使用します。

- **ヘッダー**
  - POST コールの場合は、「Content-Type」に、ヘッダーで「application/json」と指定する必要があります。すべての REST メソッド（GET、POST、PUT、DELETE 等）コールで、このヘッダーを含めるのが一般的です。
- **認証**
  - この API コールでは、認証データはボディに入れられて送信されます。
- **ボディ**
  - APIC-EM コントローラへのログインに使用するユーザ名とパスワードは、JSON 形式でボディに指定する必要があります。

ステップ 5：REST API コール

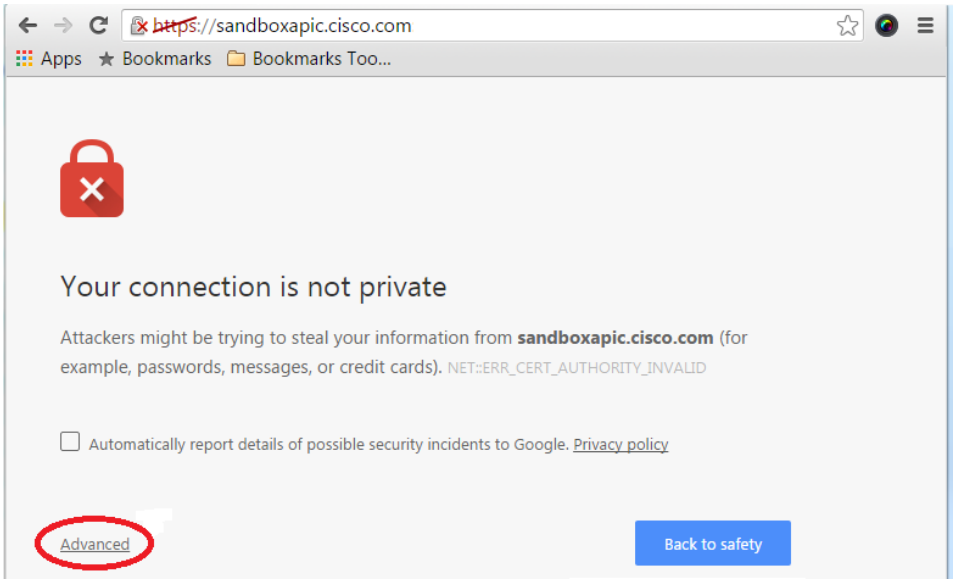
Web サービスを手軽にテストする上で、役に立つ HTTP クライアントが多数あります。これらのツールを使用すれば、簡単に、REST API のリクエストを作成して送信し、レスポンスを確認できます。

- Postman：http://www.getpostman.com/
- Firefox RestClient：https://addons.mozilla.org/en-US/firefox/addon/restclient/
- curl を使用したコマンド ライン：http://curl.haxx.se/docs/httpscripting.html#POST
- SOAPUI
- REST サービスのテスト機能が組み込まれた各種 IDE コンソール

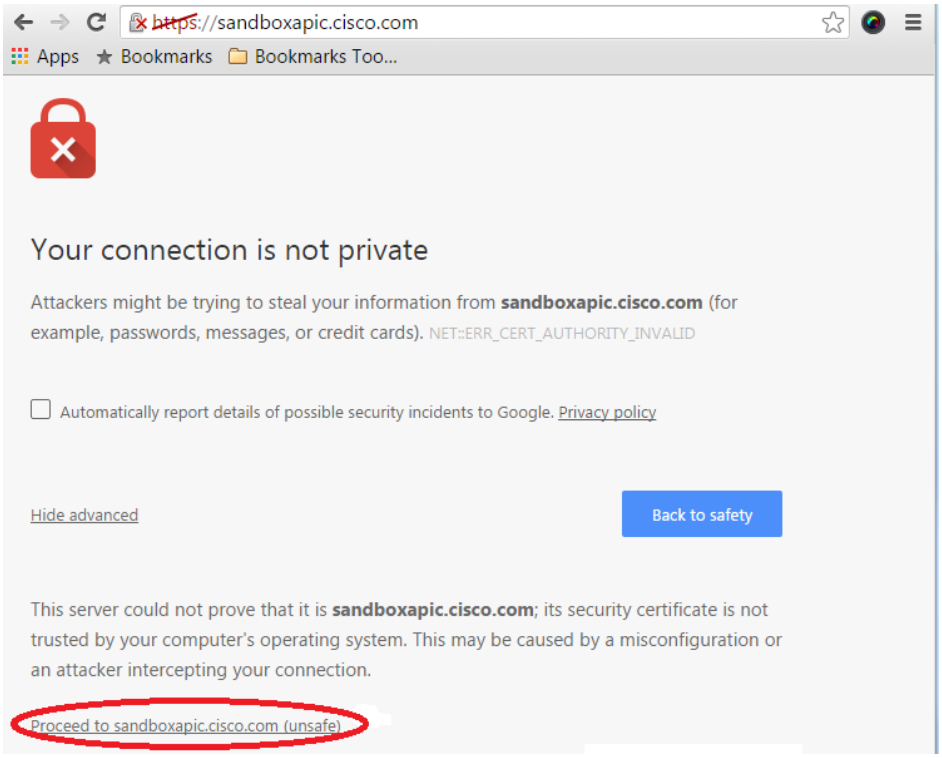
このラボでは、REST クライアントの例として、Postman を使用します。

使用前に SSL 証明書を受け入れます。Postman で API をコールする前に SSL 証明書を受け入れる必要があります。証明書の警告が表示される場合は、サンドボックスに移動し、次の手順に従います。警告が表示されない場合は、省略できます。

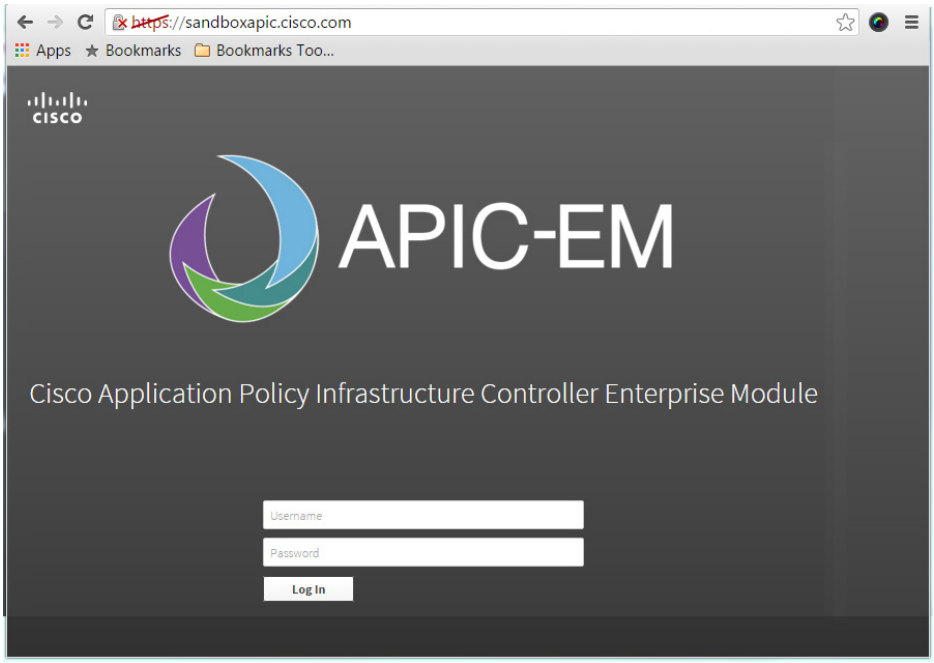
1. Chrome を開きます。
2. DevNet の Always-On APIC-EM サンドボックスを使用している場合は、https://sandboxapic.cisco.com に移動します。
3. 証明書の問題を示すメッセージが表示された場合は、[詳細（Advanced）] リンクをクリックします。メッセージが表示されず、APIC-EM UI のログイン画面に遷移する場合は、「Postman を使用して REST API コールを発信する」の項にスキップします。



4. [進む（Proceed to）] のリンクをクリックします。

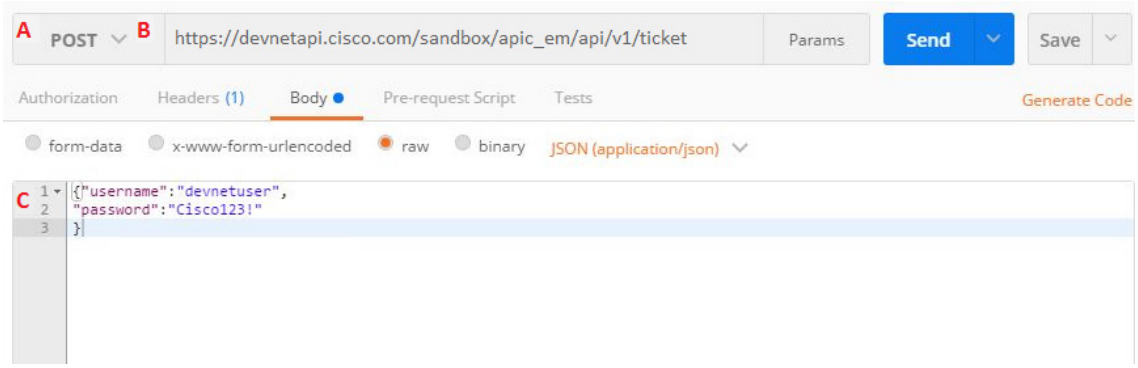


5. APIC-EM のログイン画面が表示されます。これで、Postman で作業を開始できるようになりました。



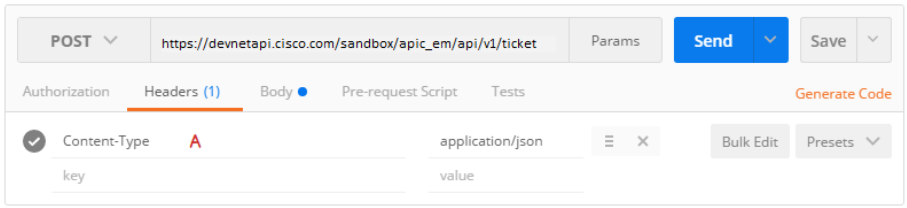
Postman を使用して REST API コールを発信する

1. Chrome を起動し、Postman を開きます。
    - DevNet Zone のステーションにいる場合は、すでに Postman がインストールされています。Google App Launcher から起動するか、Chrome から Postman をクリックして Launch App を実行し、起動させることができます。それ以外の場合は、Chrome ブラウザでこのページを表示していることを確認した上で、上記の Postman のリンクをクリックし、指示にしたがって Postman をインストール後、起動してください。
2. 次を参照して、Postman の該当するフィールドに、ホスト一覧を取得するリクエスト用の情報を入力します。



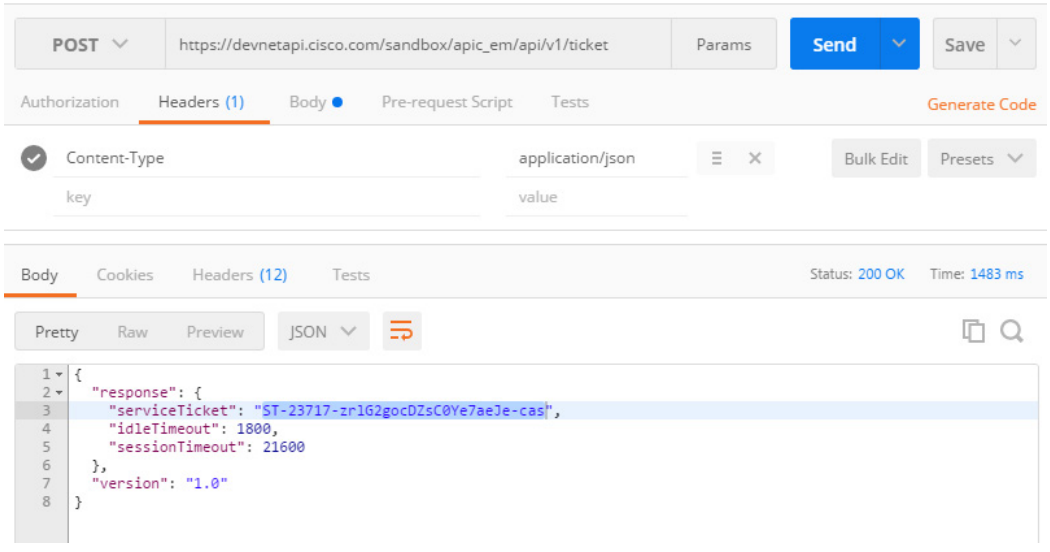
    - **メソッド**
        - A: メソッドのドロップ ダウンから [POST] を選択します。
    - **URL**
        - B: **http://{APIC-EMController}/api/v1/ticket** と入力します。
            - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com）を使用します。
    - **ボディ**
        - C: ユーザ名とパスワードを JSON 形式で入力します。これらのクレデンシャルは、APIC-EM コントローラへのログインに使用されます。DevNet の APIC-EM Always-On ラボにアクセスしている場合は、上記のユーザ名とパスワードを入力します。そうでない場合は、自分の APIC-EM コントローラに必要なユーザ名とパスワードを入力します。

- **ヘッダー**
  - A:ヘッダーに「Content-Type」、「application/json」と入力します。



- **認証**
  - 上記のようにボディに指定された認証情報が使用されます。

3. [送信（Send）] をクリックします。
4. Postman はサーバにリクエストを送信し、レスポンスを表示します。
   - [レスポンス コード（Response Code）] が戻され、[ステータス（Status）] フィールドに表示されるのを確認します。この場合は「200」です。
   - 「serviceTicket」属性を含む [JSON] レスポンスデータが表示されます。サービス チケットの値をハイライトで示しています。実際に返される値は示されているものとは異なります。**次のステップで使用するために、この値をテキスト ファイルにコピー アンド ペーストします。**



おめでとうございます。最初の REST API コールを実行できました。

ステップ 6：サービス チケットを使用した APIC-EM REST API コール

APIC-EM の例：ホストの取得

ホストを取得する REST API コールについて説明します。ホストはワークステーションなどのエンド デバイスです。ネットワーク ケーブルでスイッチなどのネットワーク デバイスに接続されたり、ワイヤレス デバイスに接続されたりします。ここでの目的は、ホストを見つけ、その情報を表示することです。

ホスト一覧を取得するためのリクエスト方法を知るためには、API のドキュメントを使用して次の内容を決定する必要があります。

- メソッド
- URL
- ヘッダー
- 認証
- ボディ

API のドキュメントで記述されているように、API-EM 用の URL はすべて次のように始まります。

```
https://{APIC-EM-Server}/api/v1
```

ホスト取得の API の詳細については、APIC-EM API ドキュメント [英語] を参照してください。

下のコールによって、指定されたサーバ上のすべてのホストが取得されます。

```
GET https://{APIC-EM-Server}/api/v1/host
```

一方、表示上の理由から、ホストの一部だけを部分的に取得したい場合があります。その場合は、たとえば、受け取るホストの最大数を指定する「limit」パラメータを渡すことができます。また、必要に応じて、ホストの一覧をどこから始めるかを指定する「offset」パラメータを渡すこともできます。下のコールでは、最初のホストから開始し（offset=1）、受け取るホストの最大数が 5 になるように指定しています（limit=5）。

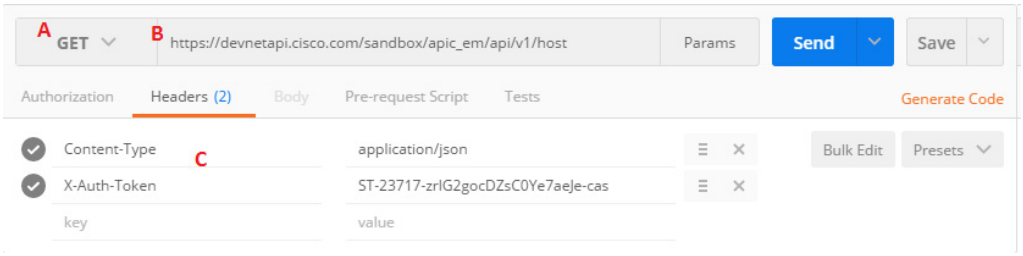
```
GET https://{APIC-EM-Server}/api/v1/host?limit=5&offset=1
```

ホストの一覧をすべて取得する場合、リクエストは次のようになります。

- **メソッド**：GET
- **URL**：https://{APIC-EM-Server}/api/v1/host
  - アドレスは使用する APIC-EM コントローラのアドレスで置き換えます。
    - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com/）を使用します。
- **ヘッダー**
  - 認証のため、以前に作成したサービス チケットを追加する必要があります。ヘッダーの左側で、「X-Auth-Token」というテキストを追加します。右側には、エンドポイントへ Create Ticket をコールして戻ってきたサービス チケットの値を追加します。
  - GET を使用する際には、リクエストにコンテンツ ヘッダーは必要ありませんが、実際には、コンテンツ ヘッダーで「Content-Type」に「application/json」を指定することを推奨します。
- **認証**
  - 上記のヘッダーで行います。
- **ボディ**
  - このリクエストには不要です。

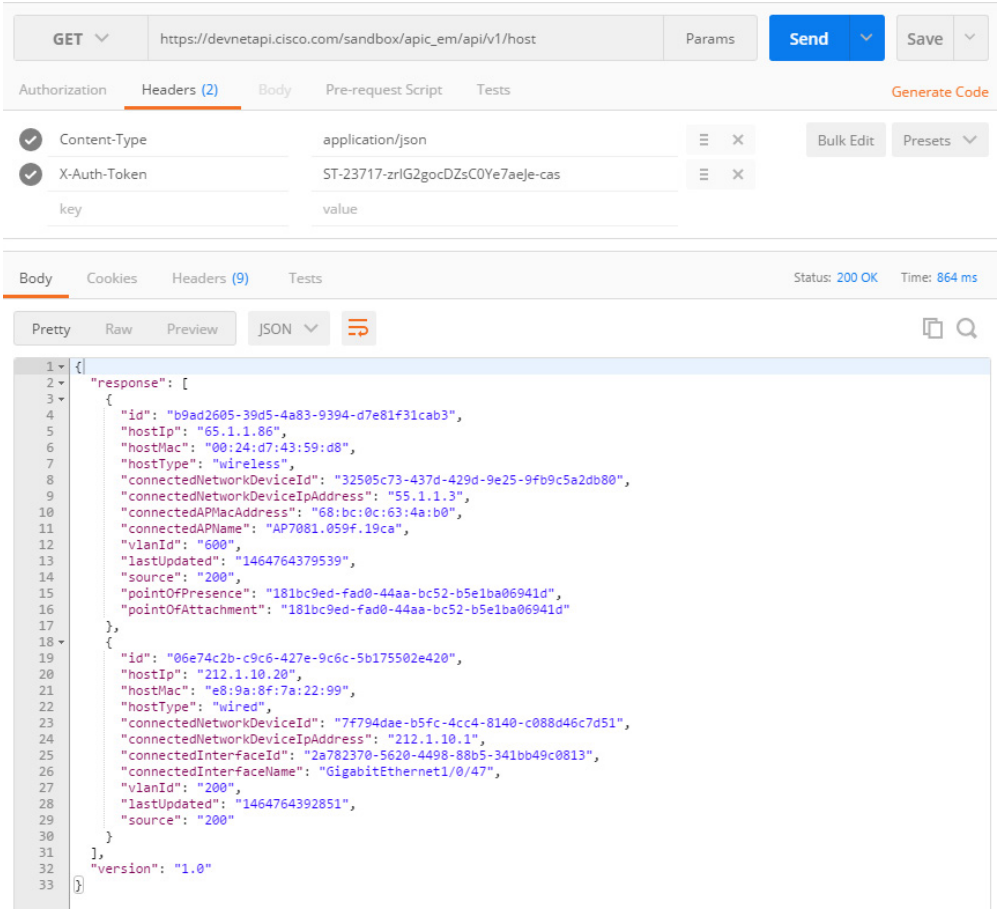
ステップ 7：ネットワーク ホストの取得

1. Postman で API コールを設定します。
    - ここでは、コントローラからホストを取得するリクエストの認可にサービス チケットを使用します。



2. 次を参照して、Postman の該当するフィールドに、ホスト一覧を取得するリクエスト用の情報を入力します。
    - **メソッド**
        - A：メソッドのドロップ ダウンから [GET] を選択します。
    - **URL**
        - B：http://{APIC-EMController}/api/v1/host と入力します。
            - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com/）を使用します。
    - **ヘッダー**
        - C：「Content-Type」と入力し、値に「application/json」を指定します。
        - C：「X-Auth-Token」と入力し、以前作成したサービス チケットの値を指定します。「X-Auth-Token」は、APIC-EM コントローラのすべての API コールに必要です。
    - **認証**
        - ヘッダーで行います。
    - **ボディ**
        - このリクエストには不要です。
3. [送信（Send）] をクリックします。
4. Postman はサーバにリクエストを送信し、レスポンスを表示します。

- [レスポンス コード（Response Code）] が返され、[ステータス（Status）] フィールドに「200 OK」と示されているのを確認できます。
- 返されたホストの一覧を含む **JSON** を確認できます。



確認事項

1. Postman で、http://{APIC-EMController}/api/v1/host の URL に「?limit=1&offset=1」を追加し、http://{APIC-EMController}/api/v1/host?limit=1&offset=1 として、[送信（Send）] ボタンを押します。この場合、返ってくるデータはどこが違うでしょうか。ホスト取得 API コールの詳細については、API リファレンス ガイドを確認してください。

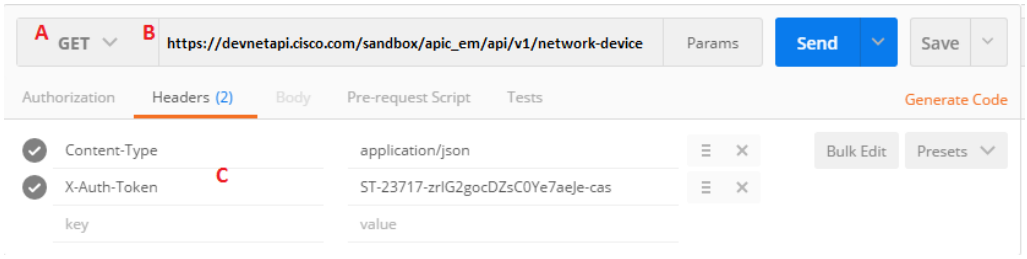
おめでとうございます。APIC-EM サービス チケットを使用する最初の REST API コールを実行できました。

ステップ 8：ネットワーク デバイスの取得

ネットワーク デバイスを取得する REST API コールについて説明します。ネットワーク デバイスは、ホスト デバイスや他のネットワーク デバイスとの間でデータを送受信します。通常、ネットワーク デバイスはハブ、スイッチ、ルータで、イーサネット ケーブルで接続するか、ワイヤレスで接続することができます。ここでの目的は、ネットワーク デバイスを見つけ、その情報を表示することです。

ネットワーク デバイスを取得する API の詳細については、APIC-EM API ドキュメント [英語] を参照してください。

1. Postman で API コールを設定します。
    - ここでは、コントローラからネットワーク デバイスを取得するリクエストの認可にサービス チケットを使用します。

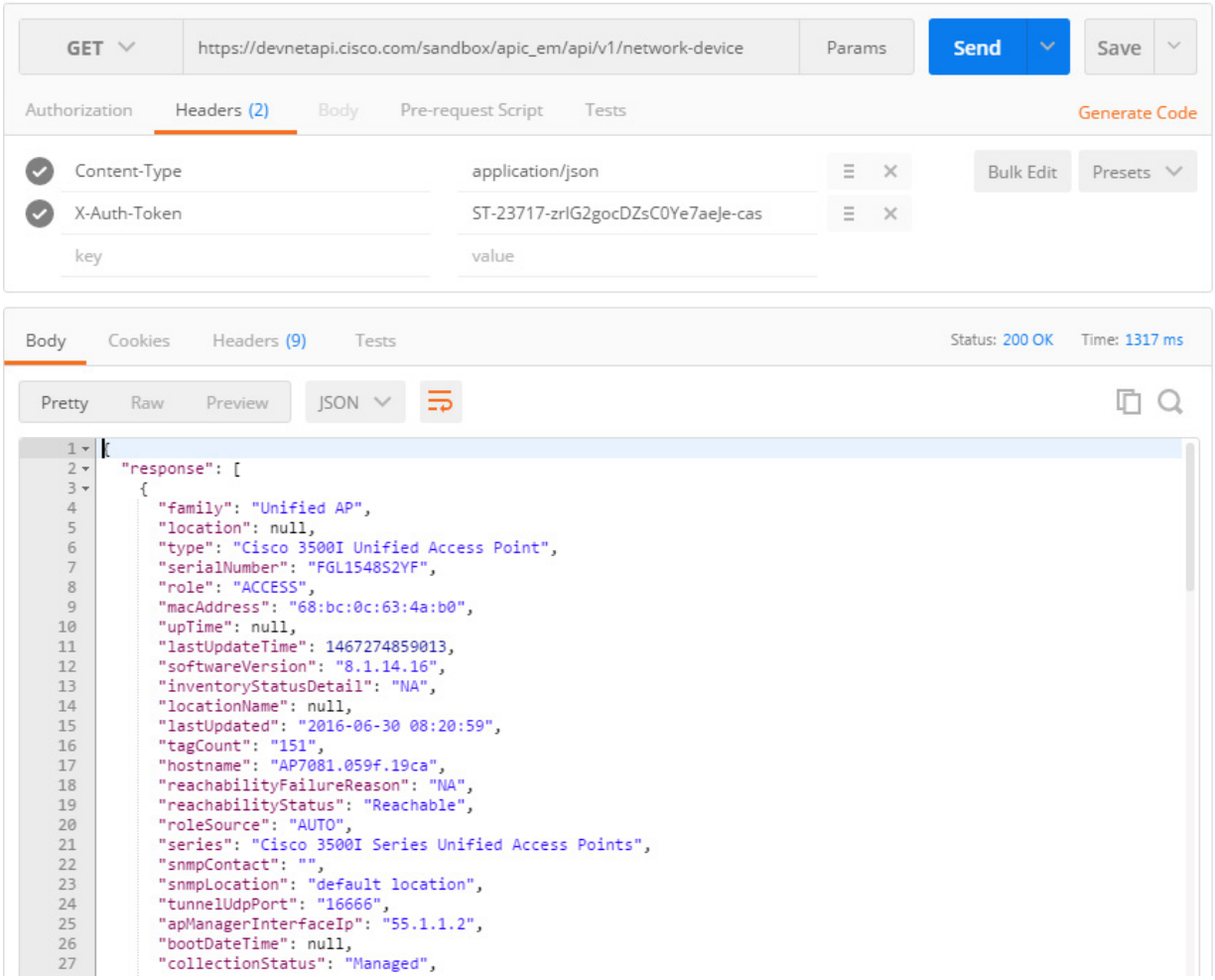


2. 次を参照して、Postman の該当するフィールドに、ネットワーク デバイス一覧を取得するリクエスト用の情報を入力します。
    - **メソッド**
        - A：メソッドのドロップ ダウンから [GET] を選択します。
    - **URL**
        - B：**http://{APIC-EMController}/api/v1/network-device** と入力します。
            - 自分の APIC-EM コントローラを使用しない場合は、Always-On APIC-EM ラボ（https://sandboxapic.cisco.com/）を使用します。
    - **ヘッダー**
        - C：「Content-Type」と入力し、値に「application/json」を指定します。
        - C：「X-Auth-Token」と入力し、以前作成したサービス チケットの値を指定します。「X-Auth-Token」は、APIC-EM コントローラのすべての API コールに必要です。

- **認証**
  - ヘッダーで行います。
- **ボディ**
  - このリクエストには不要です。

3. [送信（Send）] をクリックします。
4. Postman はサーバにリクエストを送信し、レスポンスを表示します。
   - [レスポンス コード（Response Code）] が返され、[ステータス（Status）] フィールドに「200 OK」と示されているのを確認できます。
   - 返されたホストの一覧を含む **JSON** を確認できます。

下のような出力が表示されます。表示上の理由で、何行かのレコードは削除されています。



確認事項

- Postman で、http://{APIC-EMController}/api/v1/network-device の URL に「1/2」を追加し、http://{APIC-EMController}/api/v1/network-device/1/2 として、[送信（Send）] ボタンをクリックします。この場合、出力はどこが違うでしょうか。
- 今実施したネットワーク デバイスを取得する API コールでの違いと、ステップ 7 の演習で実施した、ホストを取得するコールでの違いを比較します。どのように異なっているでしょうか。またそれはなぜでしょうか。ヒント：これらの API コールについてそれぞれ API リファレンス ガイドを確認します。
- Postman で API コールを http://{APIC-EMController}/api/v1/network-device/1/2 から http://{APIC-EMController}/api/v1/user に変更し、[送信（Send）] ボタンを押します。返ってくるデータがどのように変わり、何を表していますか。ヒント：API リファレンス ガイドを確認します。