

瞭解和使用Catalyst Center API

目錄

[簡介](#)

[背景資訊](#)

[必要條件](#)

[需求](#)

[採用元件](#)

[背景資訊](#)

[HTTP簡介](#)

[HTTP請求](#)

[HTTP響應](#)

[HTTP方法 \(動詞\)](#)

[HTTP狀態代碼](#)

[REST概述](#)

[Catalyst中心和REST API](#)

[使用Bruno檢索身份驗證令牌](#)

[1.確定API終結點](#)

[2.配置Bruno進行身份驗證](#)

[3.在Bruno中設定全域性標題](#)

[範例 1：建立現場工作台](#)

[步驟 1:獲取父ID](#)

[步驟 2:建立樓層](#)

[範例 2：檢索介面資訊](#)

[步驟 1:識別裝置ID](#)

[步驟 2:在Bruno中建立引數化請求](#)

[結論](#)

簡介

本文檔介紹如何使用API客戶端和平台的內建開發人員工具訪問和使用CaTC Northbound(Intent)API。

背景資訊

它涵蓋了基礎HTTP/REST概念，並為自動化和測試提供了實用示例。

必要條件

需求

思科建議瞭解以下主題：

- HTTP/1.1協定和狀態代碼。
- REST架構風格。
- 對Cisco Catalyst Center(CaTC)的基本瞭解
- 對Bruno (或類似Postman的API客戶端) 的基本瞭解。

採用元件

本文件中的資訊是以下列硬體與軟體版本為依據：

- Cisco Catalyst Center 2.3.7.x和3.x版
- Bruno 4.1.0版

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除 (預設) 的組態來啟動。如果您的網路運作中，請確保您瞭解任何指令可能造成的影響。

背景資訊

Catalyst中心北向API (意圖API) 允許使用者與圖形使用者介面之外的控制器進行互動。這實現了定製工作流的自動化、與第三方工具的整合，以及大規模高效的網路管理。

HTTP簡介

超文本傳輸協定(HTTP)是Web資料通訊的基礎。它遵循客戶端 — 伺服器模型，在該模型中，客戶端傳送請求並且伺服器返回響應。

HTTP請求

HTTP請求由四部分組成：

1. 請求行：指定方法(包括GET和POST)、URI和協定版本(例如GET /api/v1/nodes HTTP/1.1)。
2. 標頭：提供後設資料，如身份驗證憑據和內容型別。
3. 空行：將標題從正文分離。
4. 消息正文 (可選)：包含傳送到伺服器的資料(常見於POST/PUT/PATCH)。

HTTP響應

HTTP響應將映象請求結構：

1. 狀態行：包括狀態代碼(例如200 OK)。
2. 標頭：有關伺服器和響應資料的詳細資訊。
3. 郵件正文：請求的實際資料 (對於Catalyst Center通常採用JSON格式) 。

HTTP方法 (動詞)

- GET:從資源中檢索資料。
- POST:建立新資源。
- PUT:替換現有資源。
- 修補程式：對資源應用部分修改。
- 刪除：刪除資源。

HTTP狀態代碼

HTTP狀態代碼是由伺服器作為HTTP響應的一部分返回的標準化數字代碼。它們指示客戶端請求的一般結果，並有助於瞭解請求是否成功、是否需要執行其他操作或遇到錯誤。

這些代碼按其第一位數字分為五個主要類別。

- 1xx代碼是資訊性的，表示已收到請求並且正在繼續處理。
- 2xx代碼表示請求成功，表示伺服器成功接收、理解並處理了請求。
- 3xx代碼用於重定向，指示可能需要執行其他操作才能完成請求。
- 4xx代碼表示客戶端錯誤，表示客戶端傳送的請求存在問題。
- 5xx代碼代表伺服器錯誤，表示伺服器在嘗試處理請求時遇到問題。

REST概述

REST (表示性狀態傳輸) 是設計聯網應用程式的體系結構風格。RESTful API使用HTTP作為傳輸機制，並且通常使用JSON進行資料交換，因為其輕量級。

當客戶端與REST API互動時，伺服器返回資源的當前狀態的表示形式。這種表示作為響應的一部分傳送給客戶端，並且可以用不同的格式表示。

REST不是協定。相反，它是一種分層式樣式，定義了一套用於構建分散式系統和API的原則和約束。在現代API中，JSON是最常見的格式之一，因為它重量輕且易於處理。

典型的REST互動遵循基於客戶端的模型（在文章前面提到過）。使用者端使用GET、POST、PUT、PATCH或DELETE等方法傳送HTTP要求。伺服器處理並返回包含請求的資訊或操作結果的HTTP響應。

RESTful API通常使用HTTP作為通訊機制，對資源執行操作。在此模型中，客戶端標識資源/終結點，然後使用HTTP方法指示要對該資源執行的操作。REST API可以使用幾個主要元素進行描述。首先，URI用於標識客戶端要訪問的資源或端點。URI提供該資源在API中的位置或識別符號。第二，客戶端使用標準HTTP方法(如GET、POST、PUT、PATCH和DELETE)來定義請求操作。例如，GET可用於檢索資源，POST用於建立資源，而DELETE用於完全刪除資源。最後，資源通過表示形式進行交換，表示形式由資源的介質型別定義。這種表示法是在客戶端和伺服器之間傳輸資源資料的形式，如JSON或XML。

Catalyst中心和REST API

CaTC通過集中自動化、配置、策略管理、監控和分析，幫助簡化園區網路運營。它允許網路管理員更有效地部署和管理裝置、維護一致的策略、監控網路行為以及提高整體運營可見性。這些功能可以通過圖形使用者介面(GUI)以及通過API和外部整合來訪問。

CaTC平台還提供可用於構建自定義整合和網路操作自動化的API。通過將CaTC API與現有思科解決方案或第三方應用相結合，組織可以擴展平台功能並建立針對特定運營或業務需求的定製工作流程。

在介紹了基本概念之後，本文檔重點介紹了Cisco Catalyst Center中的北向API，也稱為目的API。這些API通過基於策略的業務意圖抽象來展示特定的Catalyst Center功能，使管理員和開發人員能夠專注於預期結果，而不是實現結果所需的單個配置步驟。

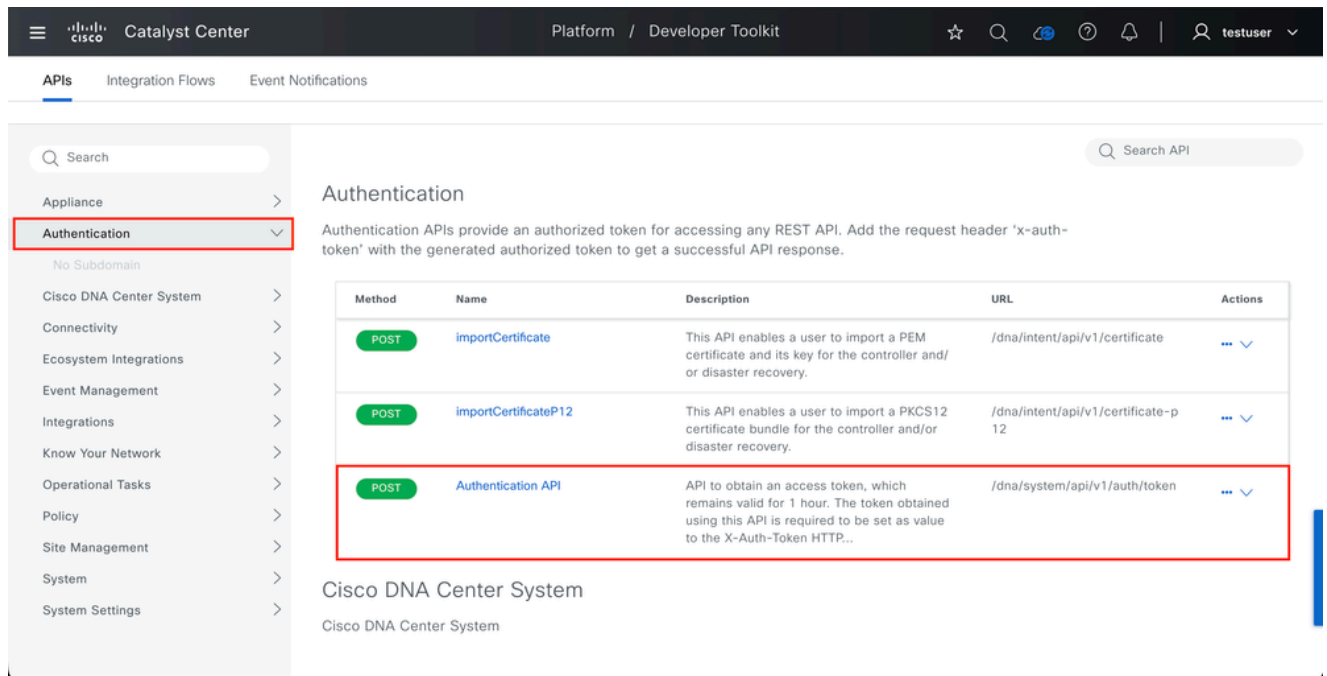
Catalyst中心意圖API是使用JSON資料格式與平台互動的RESTful API。它們可用於在整個網路中發現網路資訊、檢索運算元據並執行受支援的控制和自動化任務。

使用Bruno檢索身份驗證令牌

進行API呼叫之前，您必須進行身份驗證才能接收令牌。

1. 確定API終結點

1. 登入Catalyst Center。
2. 導航至選單>平台>開發人員工具包。
3. 在左側選單中，選擇Authentication > Authentication API。

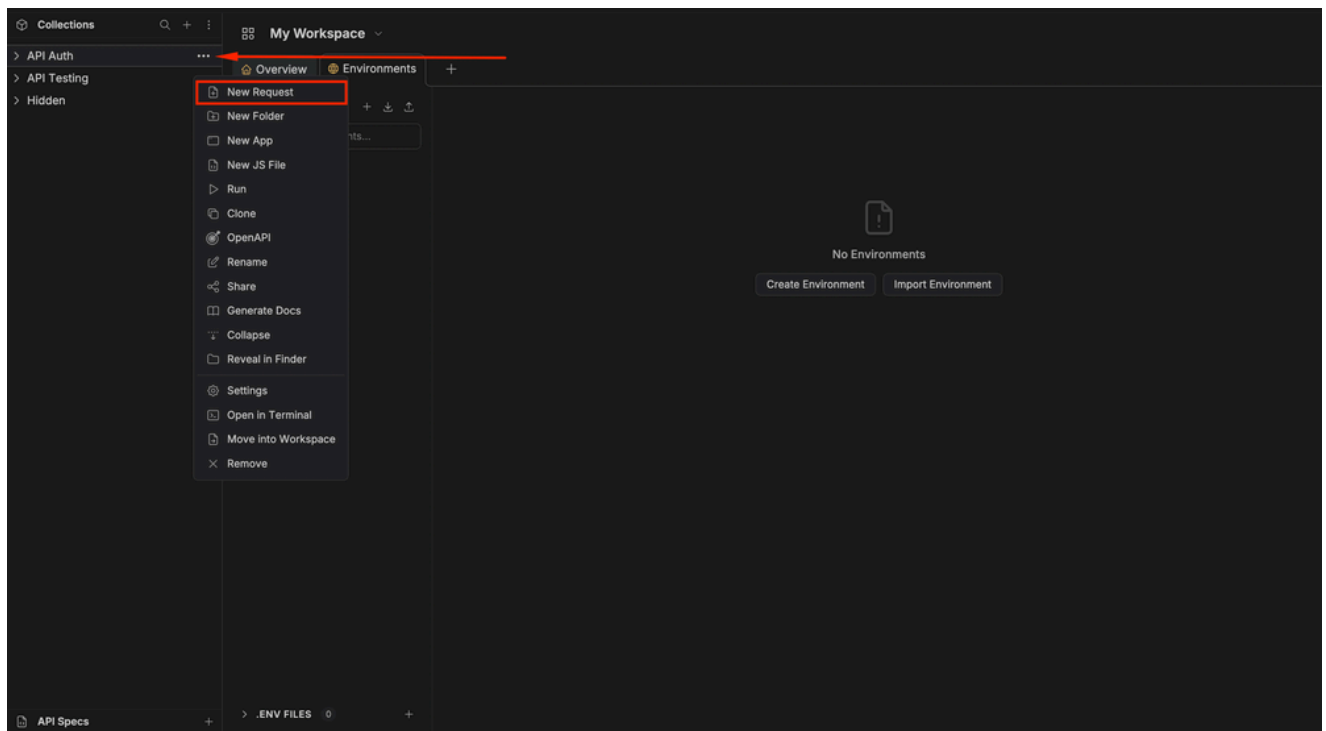


Catalyst Center開發人員工具包

4. 複製URL(通常為POST /dna/system/api/v1/auth/token)。

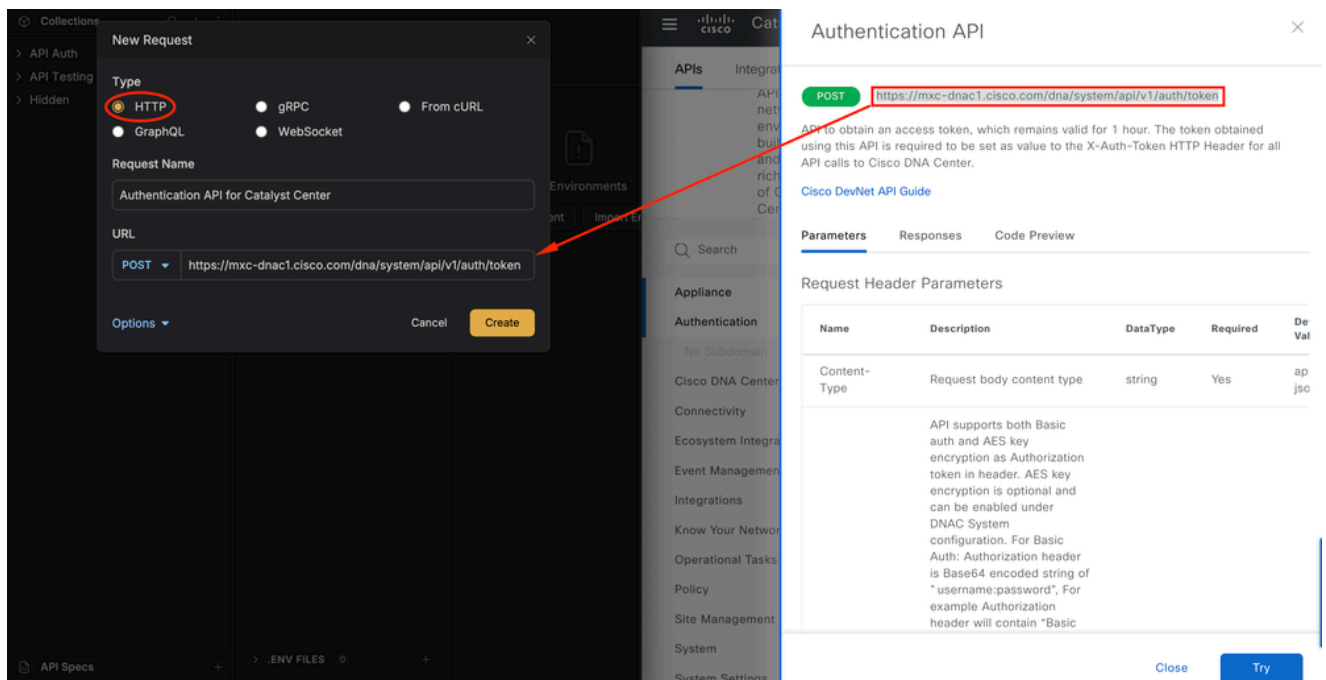
2. 配置Bruno進行身份驗證

1. 在Bruno中，建立新請求。
2. 將Type設定為HTTP，並將Method設定為POST。



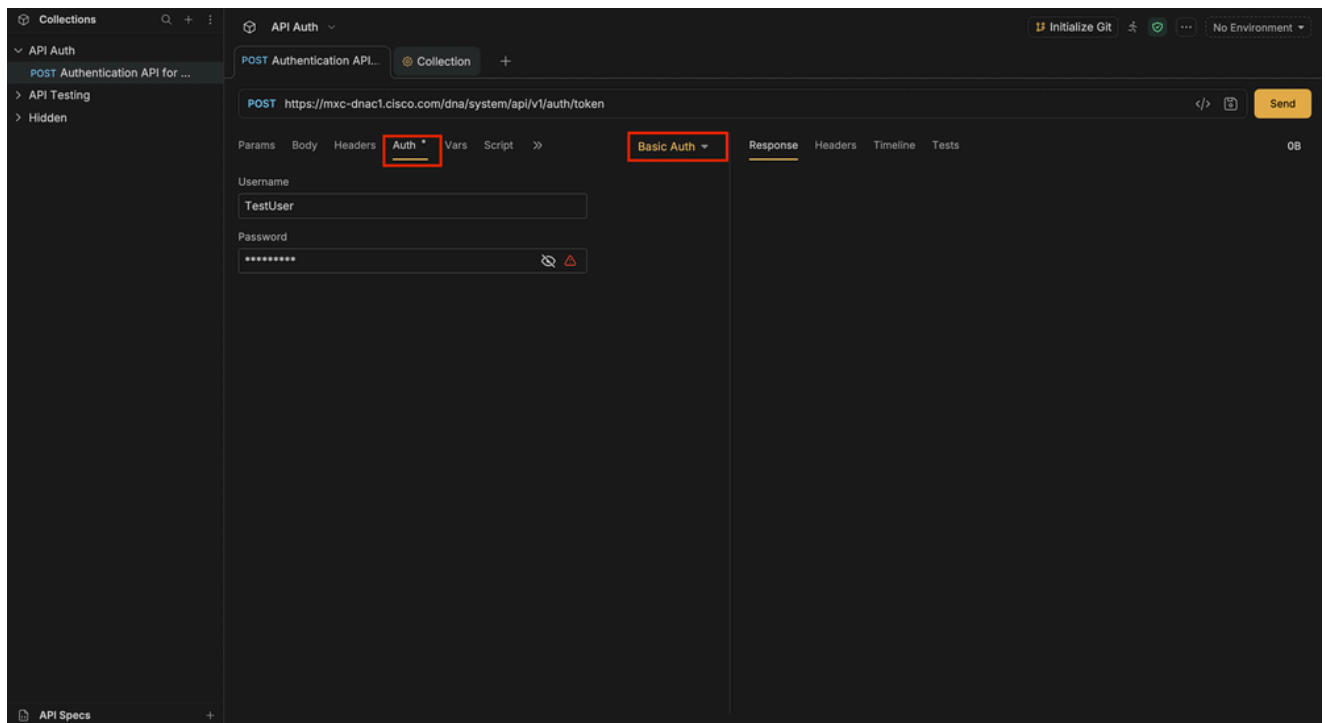
在Bruno建立新請求

3. 輸入身份驗證API的URL。



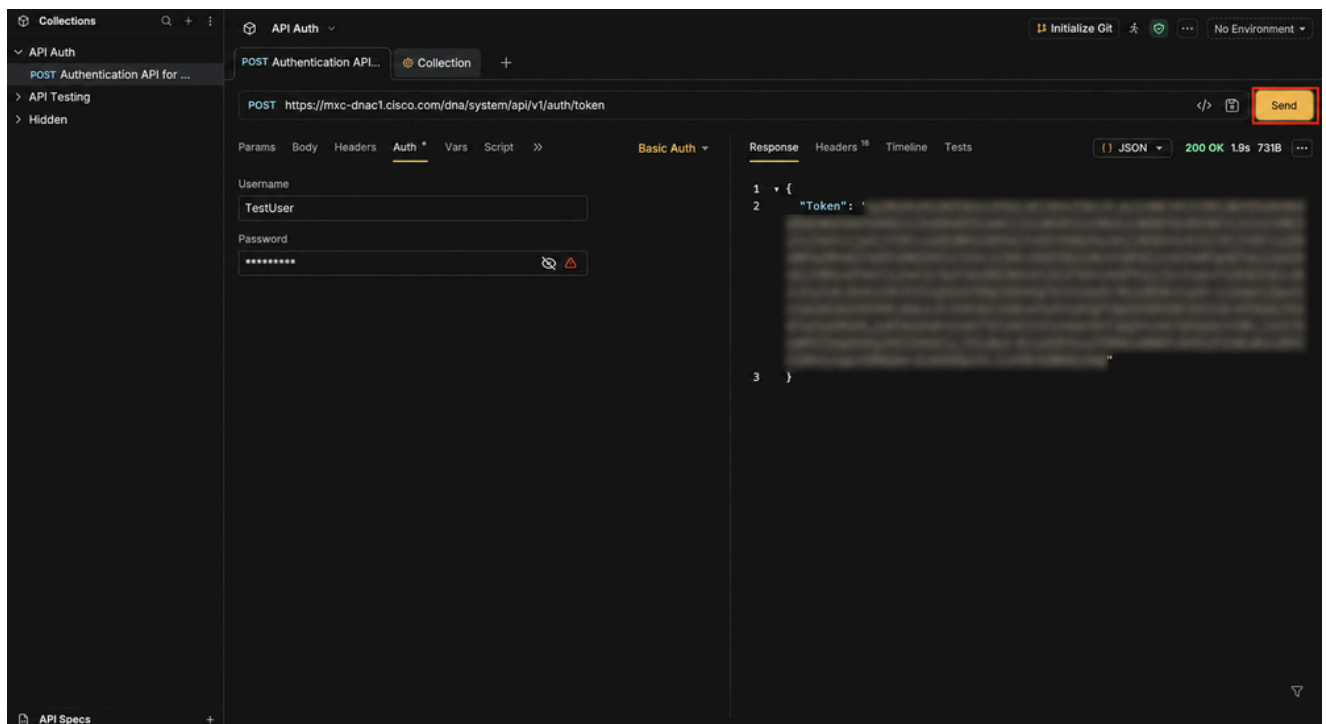
在Bruno建立新請求

4. 導覽至Auth索引標籤，然後選擇Basic Auth。



驗證API的基本驗證

5. 輸入您的Catalyst Center SUPER ADMIN憑證。
6. 按一下「Send」。



運行身份驗證API

伺服器返回200 OK狀態和包含令牌的JSON負載。

Alternative (Terminal): You can also generate the token by base64-encoding your `username:password` and

```
[ ~ % echo -n 'TestUser:Test12345678' | base64
VGZzdFVzZXI6VGZzdDEyMyEhIQ==
```

獲取使用者名稱和密碼的base64編碼

Try 'Authentication API'

Method: **POST** Public URL :https://mxc-dnac1.cisco.com/dna/system/api/v1/auth/token

PARAMETERS

HEADERS

Content-Type* application/json

Authorization* Basic VGZzdFVzZXI6VGZzdDEyMyEhIQ==

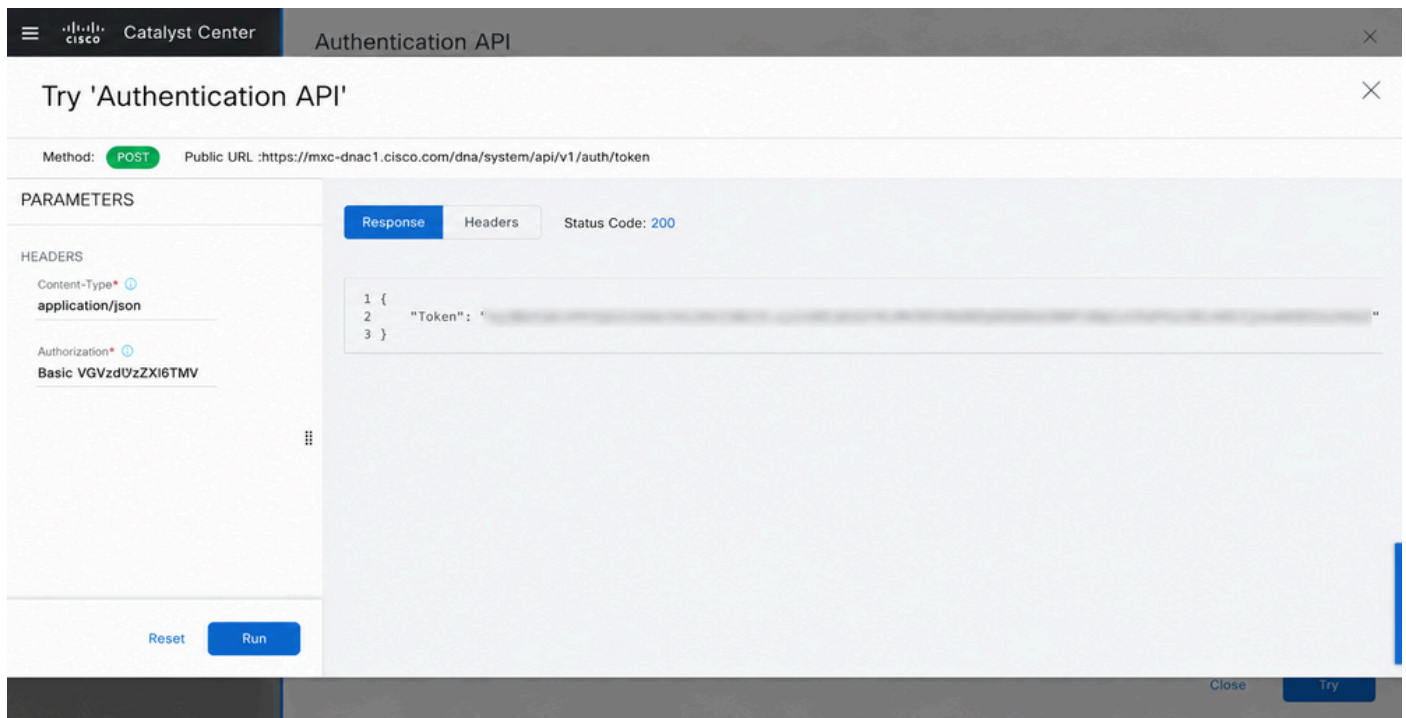
Reset Run

Enter Parameters

First enter in some necessary parameters on the left and then click Run.

Close Try

使用編碼的base64使用者名稱和密碼輸入基本身份驗證



正在從Catalyst Center GUI獲取身份驗證令牌。

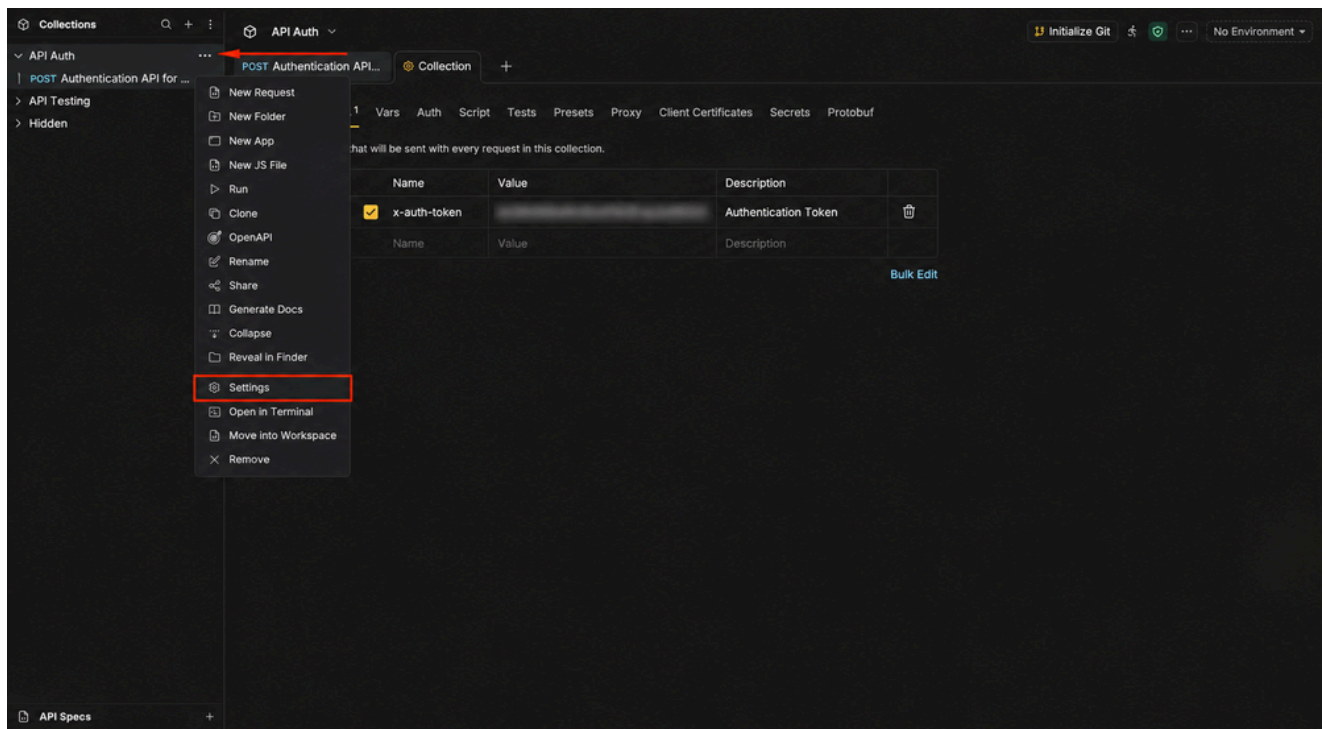
此時，您在Bruno響應或通過Catalyst Center Developer Toolkit完成的響應上擁有身份驗證令牌。

現在，在Bruno中將令牌新增為標頭，這將便於集合內建立的任何其他請求使用令牌，因此您無需手動將令牌輸入為每個請求的標頭。

3.在Bruno中設定全域性標題

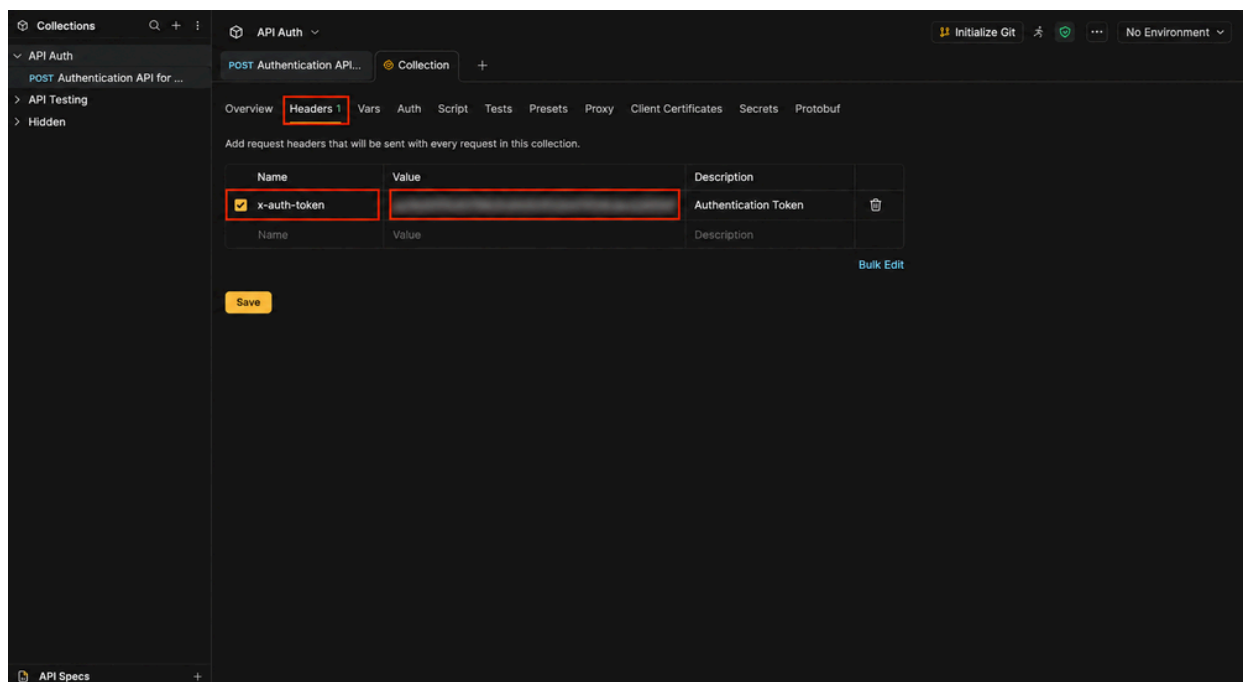
要避免對每個請求進行重新身份驗證，請將令牌設定為集合級別標頭：

1. 按一下右鍵您的收藏並選擇設定。



集合設定

2. 轉到標頭頁籤。
3. 新增新標頭：
 - 名稱:x-auth-token
 - 值： [在此處貼上您的令牌]



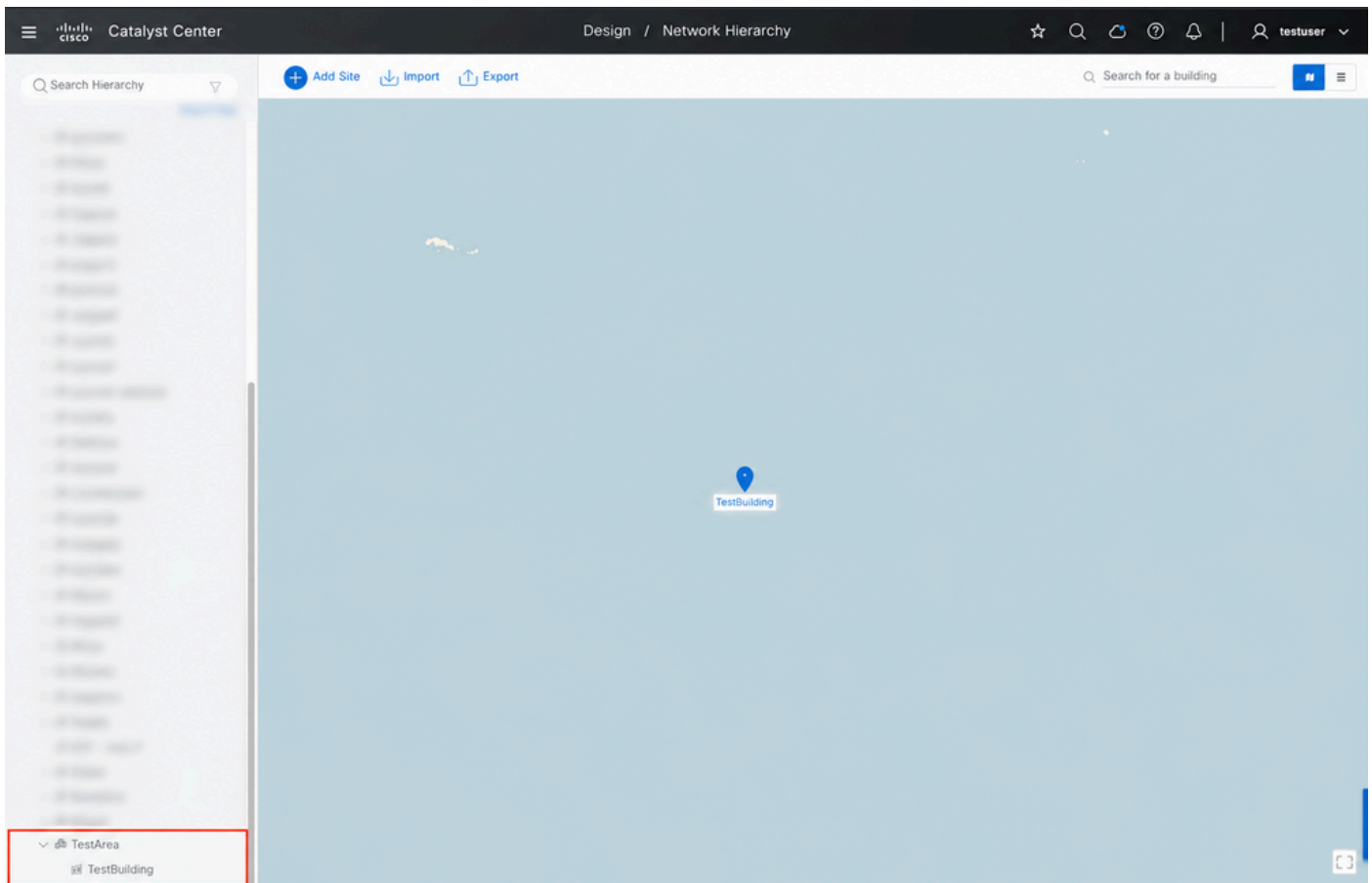
在Bruno中輸入API集合的全域性標頭。

4. 按一下「Save」。

此時，您已經生成了身份驗證令牌，以便執行任何其他Catalyst Center API請求。

範例 1：建立現場工作台

此示例演示如何在網路分層結構(Global > TestArea > TestBuilding)中建立樓層。

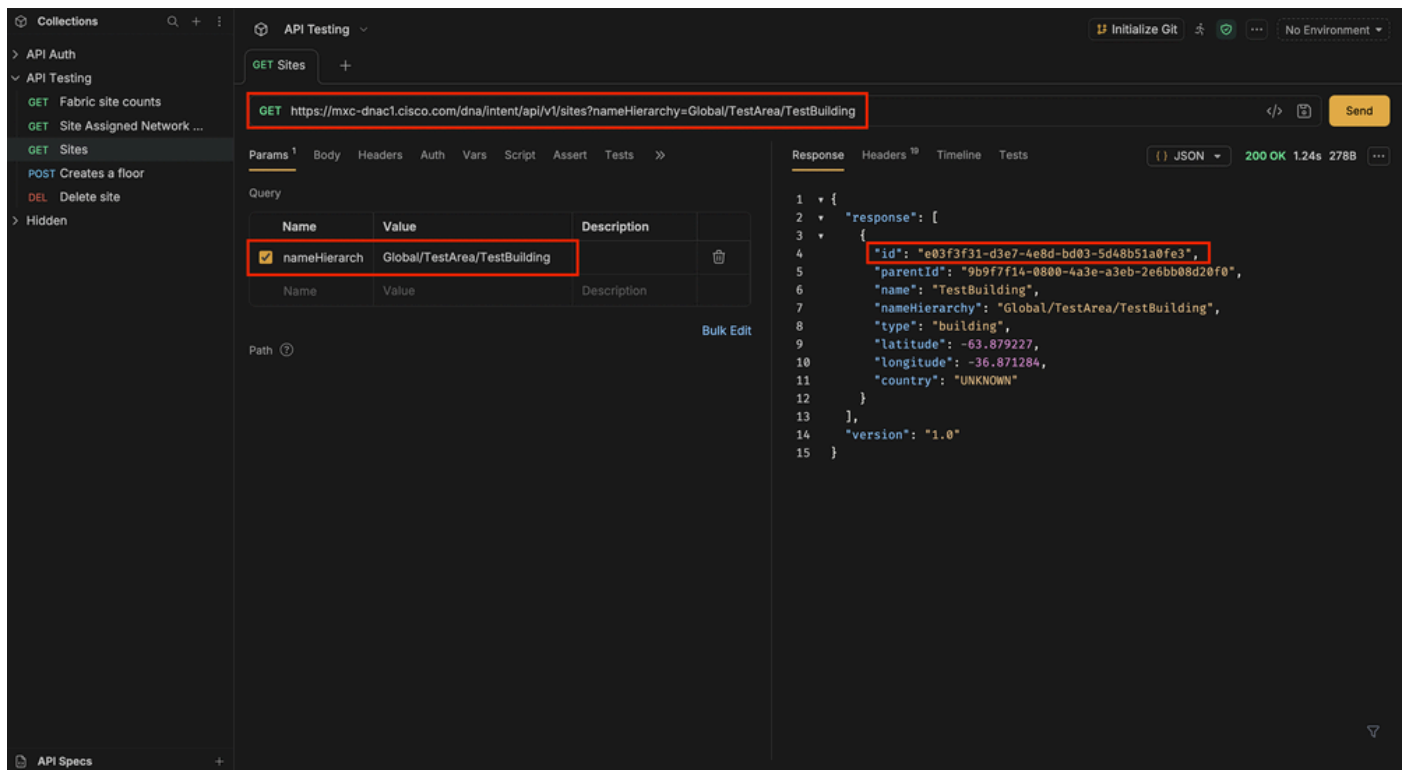


網路層次結構

步驟 1:獲取父ID

使用GET /dna/intent/api/v1/sites API查詢TestBuilding的ID。

- 查詢引數：nameHierarchy=Global/TestArea/TestBuilding
- 從響應中複製ID;這是你的parentId。



GET 請求/dna/intent/api/v1/sites?nameHierarchy=Global/TestArea/TestBuilding

步驟 2: 建立樓層

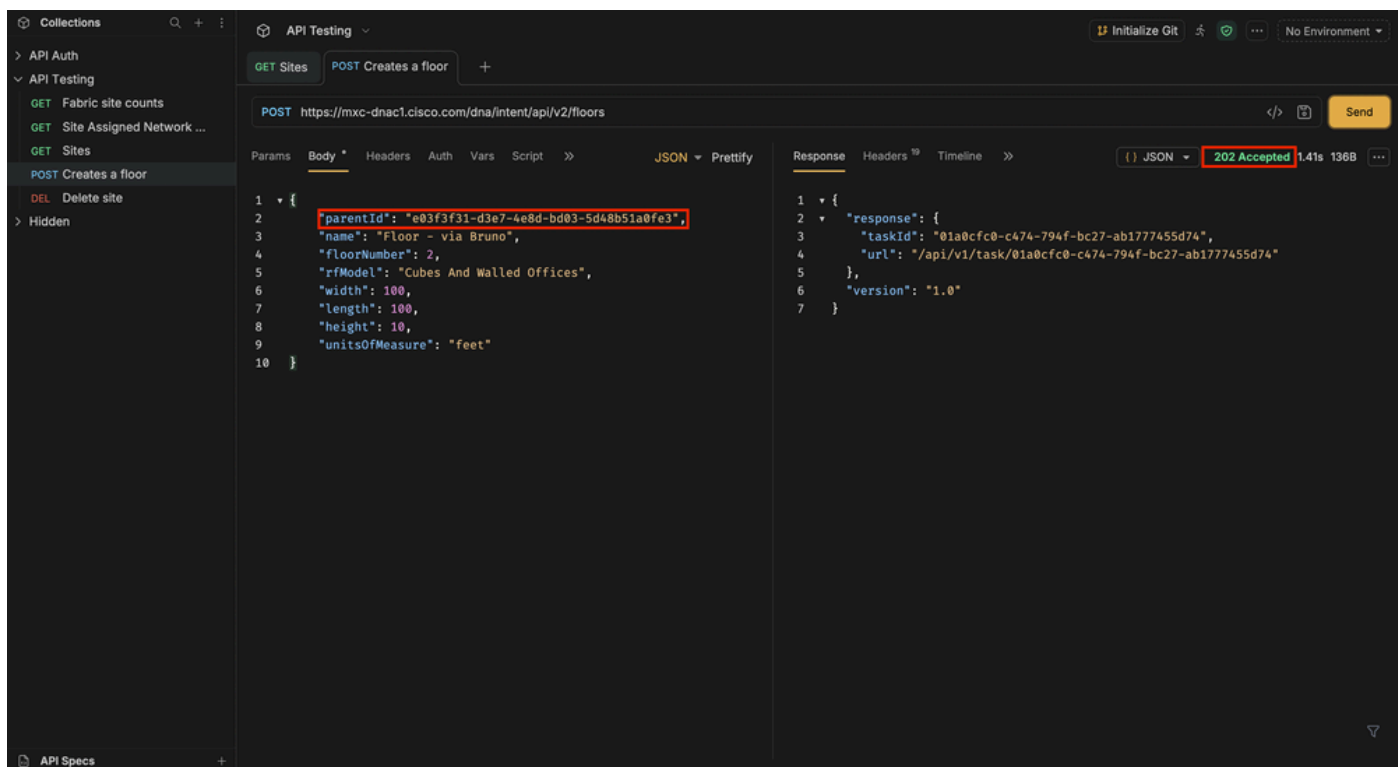
1. 建立POST請求/dna/intent/api/v2/floors。
2. 在Body頁籤(JSON)中，輸入所需的引數：

```

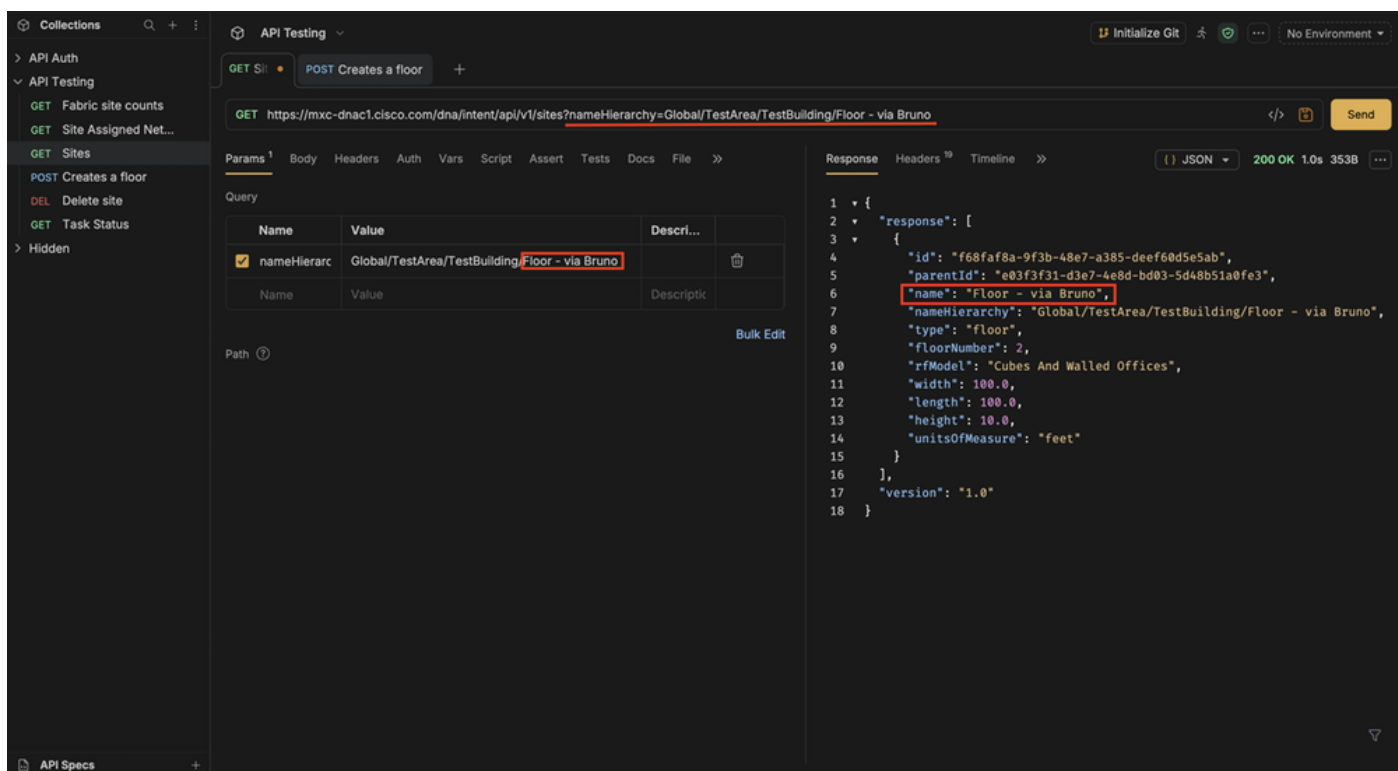
{
  "name": "Floor 1",
  "parentId": "your-building-id-here",
  "floorNumber": 1,
  "rfModel": "Cubic-Office",
  "width": 100,
  "length": 100,
  "height": 10,
  "unitsOfMeasure": "feet"
}

```

3. 按一下「Send」。成功響應確認已建立樓層。



建立樓層的POST請求



通過POST請求驗證Floor建立的GET API。

最後，要驗證已成功建立樓層，可以運行先前的API並修改查詢以匹配完整的層次結構路徑，如上文所示。

範例 2：檢索介面資訊

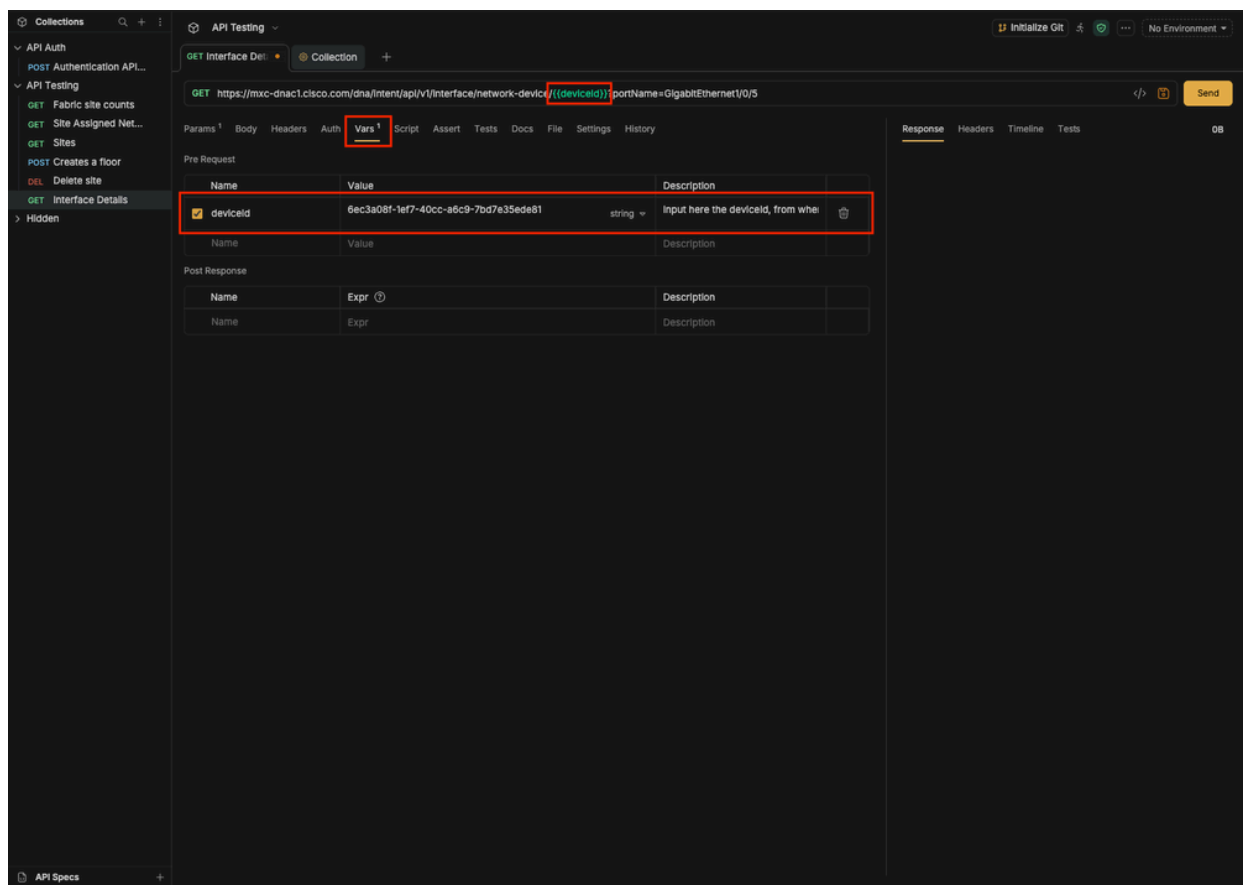
此範例顯示如何取得網路裝置上特定介面的狀態。

步驟 1:識別裝置ID

在Catalyst Center GUI中，轉到清單，按一下您的裝置，然後在瀏覽器中檢視URL。最後一個資料段是裝置ID。

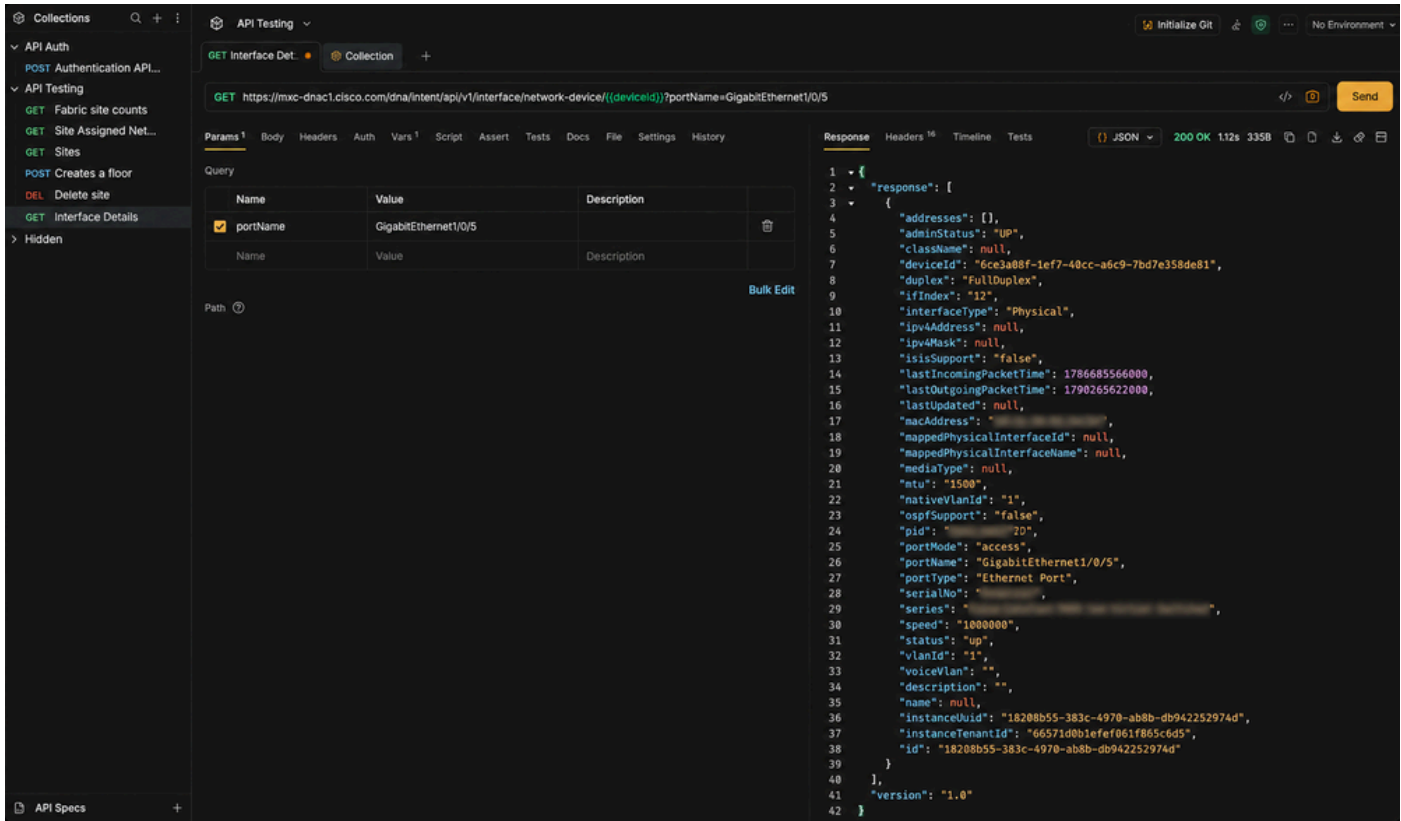
步驟 2:在Bruno中建立引數化請求

1. 建立/dna/intent/api/v1/interface/network-device/{{deviceId}}的GET請求。
2. 在請求的Vars頁籤中，新增名為deviceId的變數並貼上您的ID。
3. 要為特定埠進行過濾，請新增查詢引數：
 - 主要:名稱
 - 值：GigabitEthernet1/0/5



在Bruno中建立變數

4. 按一下「Send」。



GET請求應用查詢

結論

Catalyst中心北向API為網路管理員提供了強大的靈活性。通過從基於GUI的管理過渡到由API驅動的工作流程，您可以自動執行重複性任務、確保配置一致性，並將網路智慧整合到更廣闊的IT生態系統中。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。