

Comprensione e uso delle API di Catalyst Center

Sommario

[Introduzione](#)

[Premesse](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Introduzione a HTTP](#)

[La richiesta HTTP](#)

[Risposta HTTP](#)

[Metodi HTTP \(verbi\)](#)

[Codici di stato HTTP](#)

[Panoramica di REST](#)

[API Catalyst Center e REST](#)

[Recuperare un token di autenticazione tramite Bruno](#)

[1. Identificare l'endpoint API](#)

[2. Configurare Bruno per l'autenticazione](#)

[3. Impostare un'intestazione globale in Bruno](#)

[Esempio 1: Creazione di un piano del sito](#)

[Passaggio 1: Ottieni ID padre](#)

[Passaggio 2: Creazione della base](#)

[Esempio 2: Recupera informazioni interfaccia](#)

[Passaggio 1: Identificare l'ID del dispositivo](#)

[Passaggio 2: Creare una richiesta con parametri in Bruno](#)

[Conclusioni](#)

Introduzione

In questo documento viene descritto come accedere e utilizzare le API CaTC Northbound (Intent) utilizzando un client API e gli strumenti di sviluppo incorporati della piattaforma.

Premesse

Illustra i concetti fondamentali di HTTP/REST e fornisce esempi pratici per l'automazione e il

testing.

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- protocolli HTTP/1.1 e codici di stato.
- Stile architettonico REST.
- Conoscenze base di Cisco Catalyst Center (CaTC)
- Conoscenza di base di Bruno (o client API simili come Postman).

Componenti usati

Le informazioni di questo documento si basano sulle seguenti versioni hardware e software:

- Cisco Catalyst Center versioni 2.3.7.x e 3.x
- Bruno versione 4.1.0

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Le API di tipo Intent (Intent API) di Catalyst Center Northbound consentono agli utenti di interagire con il controller all'esterno dell'interfaccia utente grafica. Ciò consente l'automazione di flussi di lavoro personalizzati, l'integrazione con strumenti di terze parti e una gestione di rete efficiente e scalabile.

Introduzione a HTTP

Il protocollo HTTP (Hypertext Transfer Protocol) rappresenta la base della comunicazione dei dati

per il Web. Aderisce a un modello client-server in cui il client invia una richiesta e il server restituisce una risposta.

La richiesta HTTP

Una richiesta HTTP è composta da quattro parti:

1. Riga richiesta: Specifica il metodo (inclusi GET e POST), l'URI e la versione del protocollo (ad esempio GET /api/v1/nodes HTTP/1.1).
2. Intestazioni: Fornire metadati quali credenziali di autenticazione e tipo di contenuto.
3. Riga vuota: Separa le intestazioni dal corpo.
4. Corpo del messaggio (facoltativo): Contiene i dati inviati al server (comune in POST/PUT/PATCH).

Risposta HTTP

Una risposta HTTP rispecchia la struttura della richiesta:

1. Riga di stato: Include il codice di stato (ad esempio 200 OK).
2. Intestazioni: Dettagli sul server e sui dati di risposta.
3. Corpo del messaggio: Dati effettivi richiesti (in genere in formato JSON per Catalyst Center).

Metodi HTTP (verbi)

- SCARICA: Recupera i dati da una risorsa.
- POST: Crea una nuova risorsa.
- PUT: Sostituisce una risorsa esistente.
- PATCH: Applica modifiche parziali a una risorsa.
- ELIMINA: Rimuove una risorsa.

Codici di stato HTTP

I codici di stato HTTP sono codici numerici standardizzati restituiti da un server come parte di una risposta HTTP. Indicano il risultato generale di una richiesta client e consentono di determinare se la richiesta è stata eseguita correttamente, richiede un'azione aggiuntiva o ha rilevato un errore.

I codici sono raggruppati in cinque categorie principali a seconda della loro prima cifra.

- I codici 1xx sono informativi e indicano che la richiesta è stata ricevuta e l'elaborazione continua.
- I codici 2xx indicano che la richiesta ha avuto esito positivo, ovvero che il server ha ricevuto, capito ed elaborato la richiesta.
- I codici 3xx vengono utilizzati per il reindirizzamento e indicano che può essere necessaria un'azione aggiuntiva per completare la richiesta.
- I codici 4xx rappresentano errori client, ovvero si è verificato un problema con la richiesta inviata dal client.
- I codici 5xx rappresentano errori del server, ovvero il server ha rilevato un problema durante il tentativo di elaborare una richiesta.

Panoramica di REST

REST (Representative State Transfer) è uno stile architettonico per la progettazione di applicazioni in rete. Le API RESTful utilizzano HTTP come meccanismo di trasporto e in genere utilizzano JSON per lo scambio di dati a causa della sua natura leggera.

Quando un client interagisce con un'API REST, il server restituisce una rappresentazione dello stato corrente di una risorsa. Questa rappresentazione viene trasferita al client come parte della risposta e può essere espressa in un formato diverso.

REST non è un protocollo. Si tratta invece di uno stile gerarchico che definisce un insieme di principi e vincoli per la creazione di API e sistemi distribuiti. Nelle API moderne, JSON è uno dei formati più comuni perché è leggero e facile da elaborare.

Una tipica interazione REST aderisce a un modello basato su client (questo è stato menzionato in precedenza nell'articolo). Il client invia una richiesta HTTP utilizzando metodi quali GET, POST, PUT, PATCH o DELETE. Il processo server restituisce una risposta HTTP contenente le informazioni richieste o il risultato dell'operazione.

Le API RESTful in genere utilizzano HTTP come meccanismo di comunicazione per l'esecuzione di operazioni sulle risorse. In questo modello, il client identifica una risorsa o un endpoint e quindi utilizza il metodo HTTP per indicare l'azione da eseguire sulla risorsa. È possibile descrivere un'API REST utilizzando diversi elementi principali. In primo luogo, l'URI viene utilizzato per identificare la risorsa o l'endpoint a cui il client desidera accedere. L'URI fornisce la posizione o l'identificativo della risorsa all'interno dell'API. In secondo luogo, il client utilizza metodi HTTP standard come GET, POST, PUT, PATCH e DELETE per definire l'operazione di richiesta. Ad esempio, è possibile utilizzare GET per recuperare una risorsa, POST per crearne una e DELETE per rimuoverla completamente. Infine, la risorsa viene scambiata tramite una rappresentazione, definita dal relativo tipo di supporto. Questa rappresentazione è il formato in cui i dati delle risorse vengono trasferiti tra il client e il server, ad esempio JSON o XML.

API Catalyst Center e REST

CaTC semplifica le operazioni di rete del campus centralizzando automazione, configurazione, gestione delle policy, monitoraggio e analisi. Consente agli amministratori di rete di installare e gestire i dispositivi in modo più efficiente, mantenere regole coerenti, monitorare il comportamento della rete e migliorare la visibilità operativa complessiva. L'accesso a queste funzionalità è possibile tramite l'interfaccia grafica dell'utente (GUI), nonché tramite API e integrazioni esterne.

La piattaforma CaTC fornisce anche API che possono essere utilizzate per creare integrazioni personalizzate e automatizzare le operazioni di rete. Combinando le API CaTC con le soluzioni Cisco esistenti o con applicazioni di terze parti, le organizzazioni possono estendere le funzionalità della piattaforma e creare flussi di lavoro personalizzati per esigenze operative o di business specifiche.

Dopo aver illustrato i concetti di base, questo documento si concentra sulle API Northbound, note anche come API intent, all'interno di Cisco Catalyst Center. Queste API espongono funzionalità specifiche di Catalyst Center tramite un'astrazione basata su regole delle finalità aziendali, consentendo ad amministratori e sviluppatori di concentrarsi sul risultato desiderato piuttosto che sui singoli passaggi di configurazione richiesti per raggiungerlo.

Le API intent Catalyst Center sono API RESTful che utilizzano il formato dati JSON per interagire con la piattaforma. Possono essere utilizzati per individuare informazioni sulla rete, recuperare dati operativi ed eseguire attività di controllo e automazione supportate in tutta la rete.

Recuperare un token di autenticazione tramite Bruno

Prima di effettuare chiamate API, è necessario eseguire l'autenticazione per ricevere un token.

1. Identificare l'endpoint API

1. Accedere a Catalyst Center.
2. Selezionare Menu > Piattaforma > Developer Toolkit.
3. Nel menu a sinistra, selezionare Authentication > Authentication API (Autenticazione > API di autenticazione).

The screenshot shows the Catalyst Center Developer Toolkit interface. On the left, a sidebar lists various categories, with 'Authentication' highlighted in red. The main area displays the 'Authentication' section, which includes a table of APIs. The table has columns for Method, Name, Description, URL, and Actions. The 'Authentication API' is highlighted in red, showing a POST method, the name 'Authentication API', a description about obtaining an access token, and the URL '/dna/system/api/v1/auth/token'.

Method	Name	Description	URL	Actions
POST	importCertificate	This API enables a user to import a PEM certificate and its key for the controller and/or disaster recovery.	/dna/intent/api/v1/certificate	...
POST	importCertificateP12	This API enables a user to import a PKCS12 certificate bundle for the controller and/or disaster recovery.	/dna/intent/api/v1/certificate-p12	...
POST	Authentication API	API to obtain an access token, which remains valid for 1 hour. The token obtained using this API is required to be set as value to the X-Auth-Token HTTP...	/dna/system/api/v1/auth/token	...

Catalyst Center Developer Toolkit

4. Copiare l'URL (generalmente POST /dna/system/api/v1/auth/token).

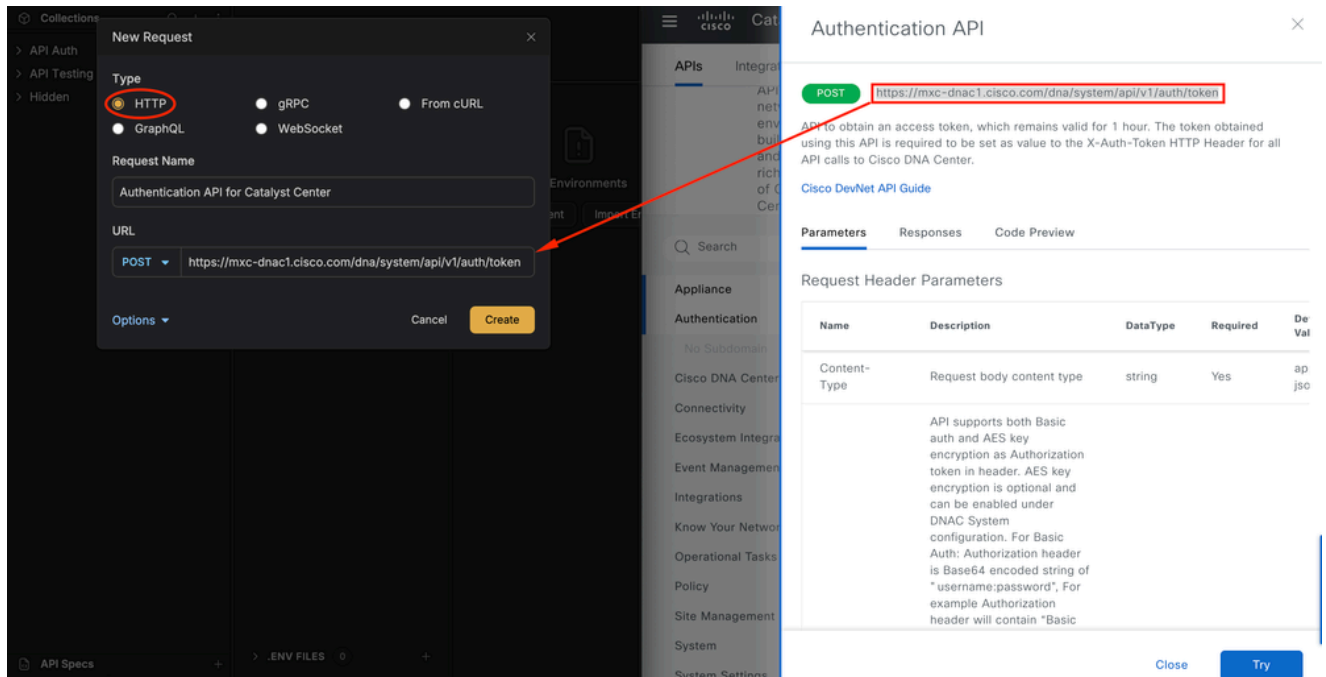
2. Configurare Bruno per l'autenticazione

1. In Bruno creare una nuova richiesta.
2. Impostare Type su HTTP e Method su POST.

The screenshot shows the Bruno application interface. On the left, a sidebar lists various collections, with 'API Auth' highlighted. The main area displays the 'New Request' menu, which is open. The menu options include 'New Request', 'New Folder', 'New App', 'New JS File', 'Run', 'Clone', 'OpenAPI', 'Rename', 'Share', 'Generate Docs', 'Collapse', 'Reveal in Finder', 'Settings', 'Open in Terminal', 'Move into Workspace', and 'Remove'. The 'New Request' option is highlighted in red.

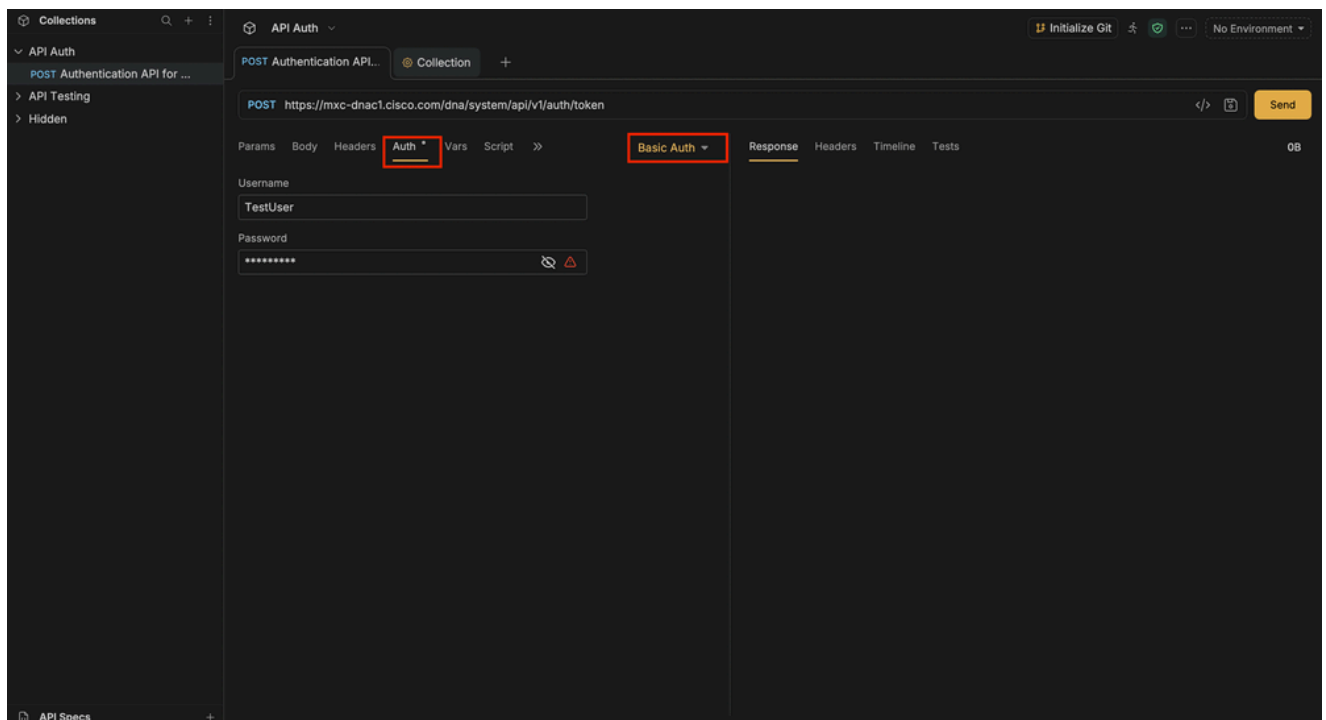
Creazione di una nuova richiesta in Bruno

3. Immettere l'URL per l'API di autenticazione.



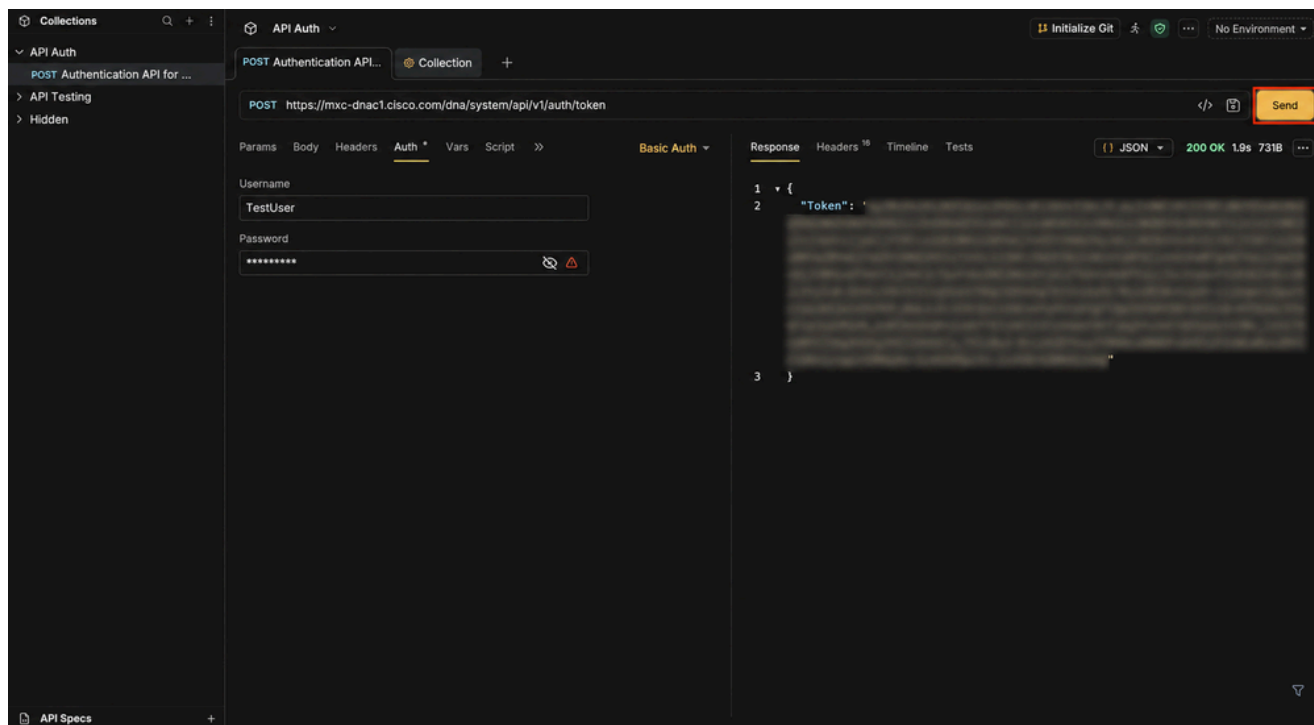
Creazione di una nuova richiesta in Bruno

4. Passare alla scheda Autenticazione e selezionare Autenticazione di base.



Autenticazione di base per l'API di autenticazione

5. Immettere le credenziali Catalyst Center SUPER ADMIN.
6. Fare clic su Invia.



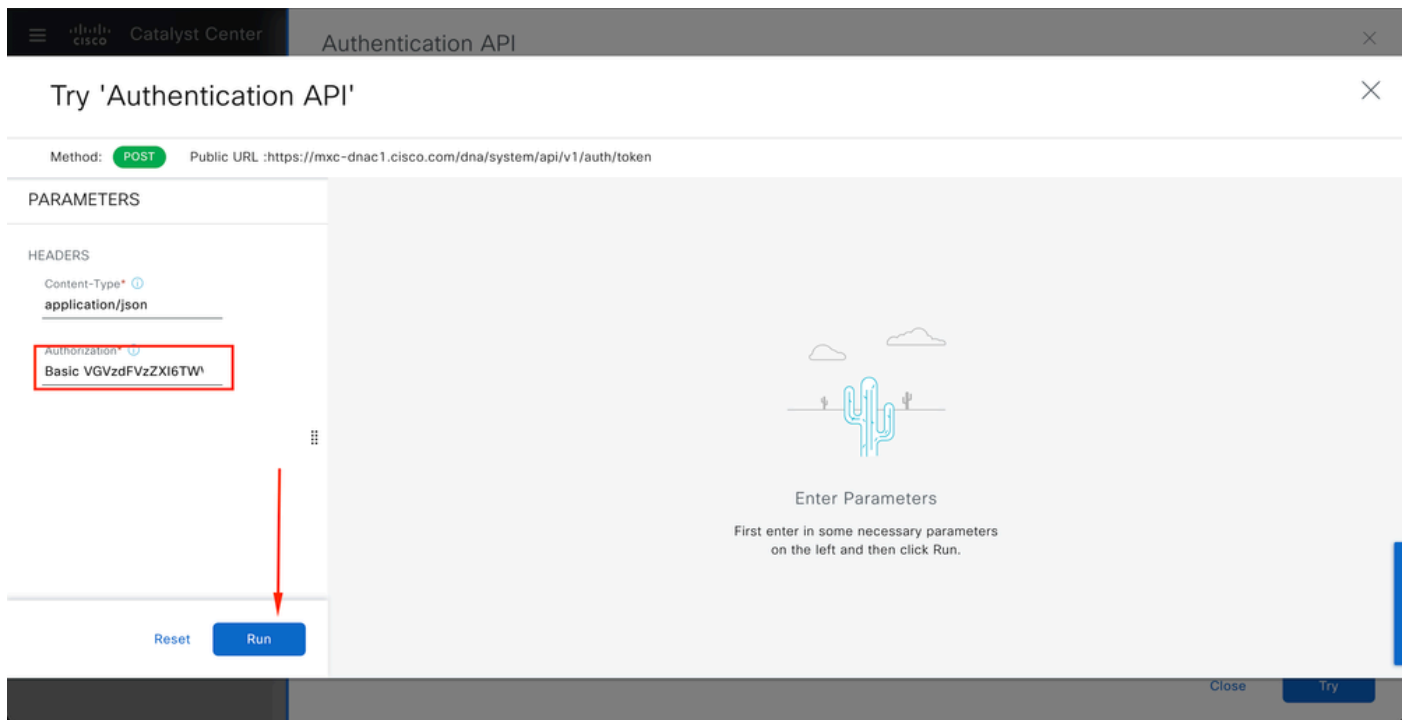
Esecuzione dell'API di autenticazione

Il server restituisce uno stato 200 OK e un payload JSON contenente il token.

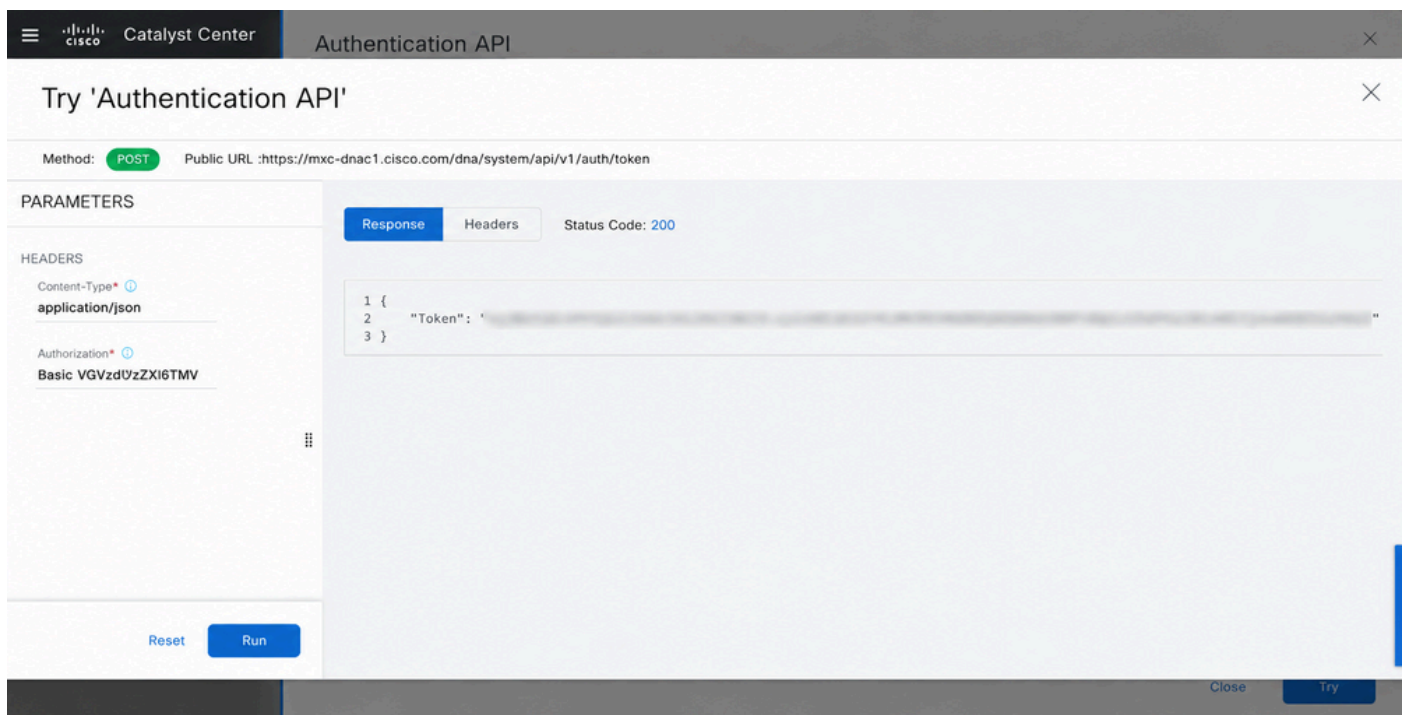
Alternative (Terminal): You can also generate the token by base64-encoding your `username:password` and

```
[redacted] ~ % echo -n 'TestUser:VGvzdFVzZXI6VGvzdDEyMyEhIQ==' | base64
VGvzdFVzZXI6VGvzdDEyMyEhIQ==
```

Recupero della codifica base64 per il nome utente e la password



Inserimento di Basic Auth con nome utente e password base64 codificati



Recupero del token di autenticazione dall'interfaccia utente di Catalyst Center.

A questo punto, il token di autenticazione si trova sulla risposta Bruno o sulla risposta fornita da Catalyst Center Developer Toolkit.

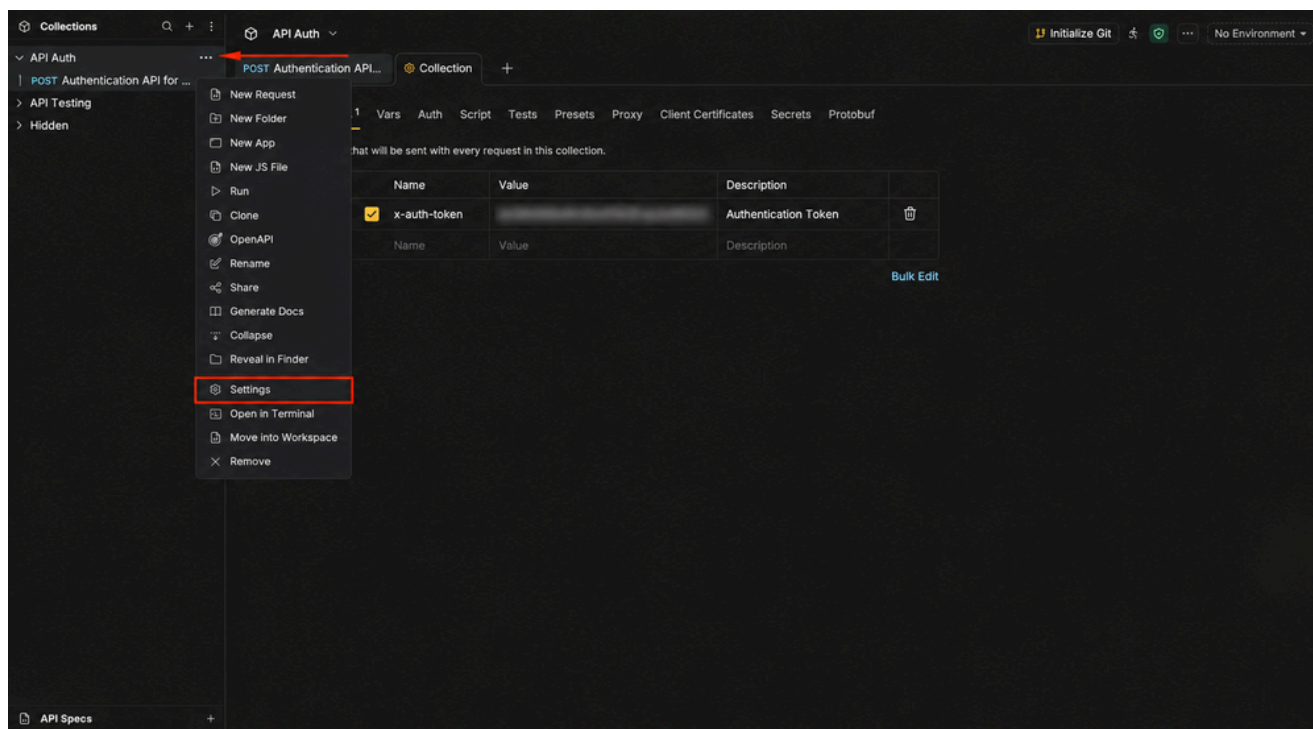
A questo punto, aggiungere il token come intestazione nelle impostazioni di raccolta in Bruno, facilitando qualsiasi altra richiesta creata all'interno della raccolta per utilizzare il token, in modo

che non sia necessario immettere manualmente il token come intestazione per ogni richiesta.

3. Impostare un'intestazione globale in Bruno

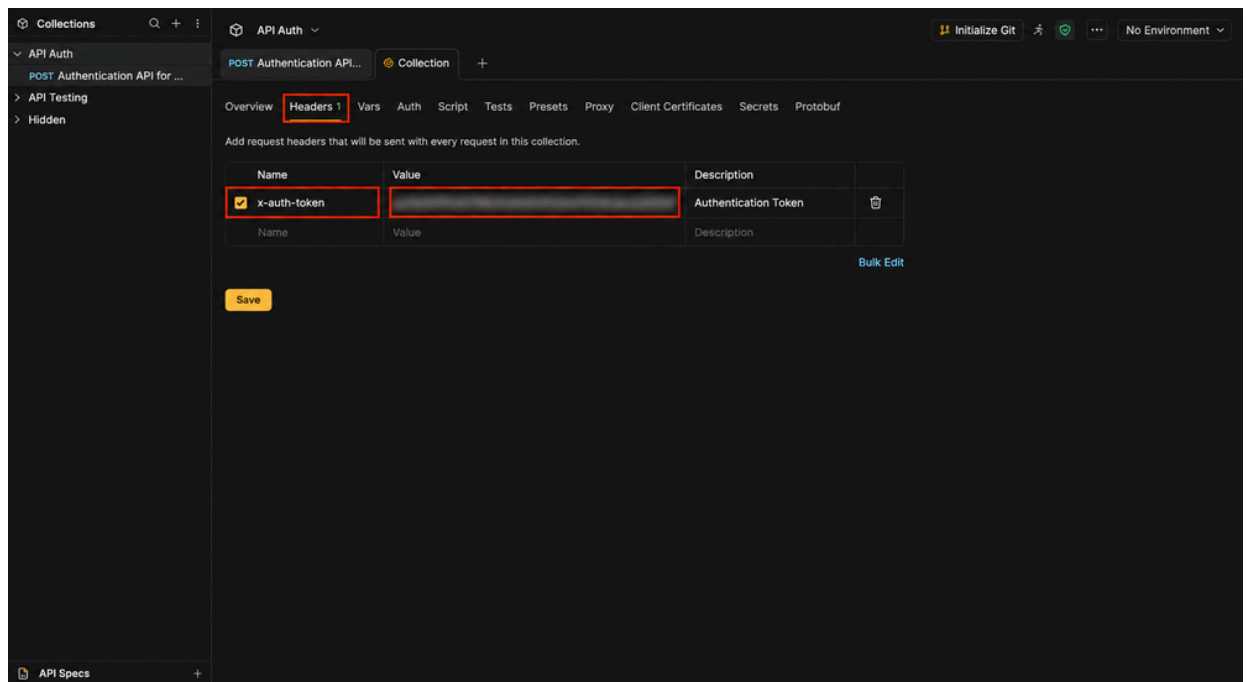
Per evitare la riautenticazione per ogni richiesta, impostare il token come intestazione a livello di raccolta:

1. Fare clic con il pulsante destro del mouse sulla raccolta e selezionare Impostazioni.



Impostazioni raccolta

2. Passare alla scheda Intestazioni.
3. Aggiungere nuova intestazione:
 - Nome: x-auth-token
 - Valore: [Incollare il token qui]



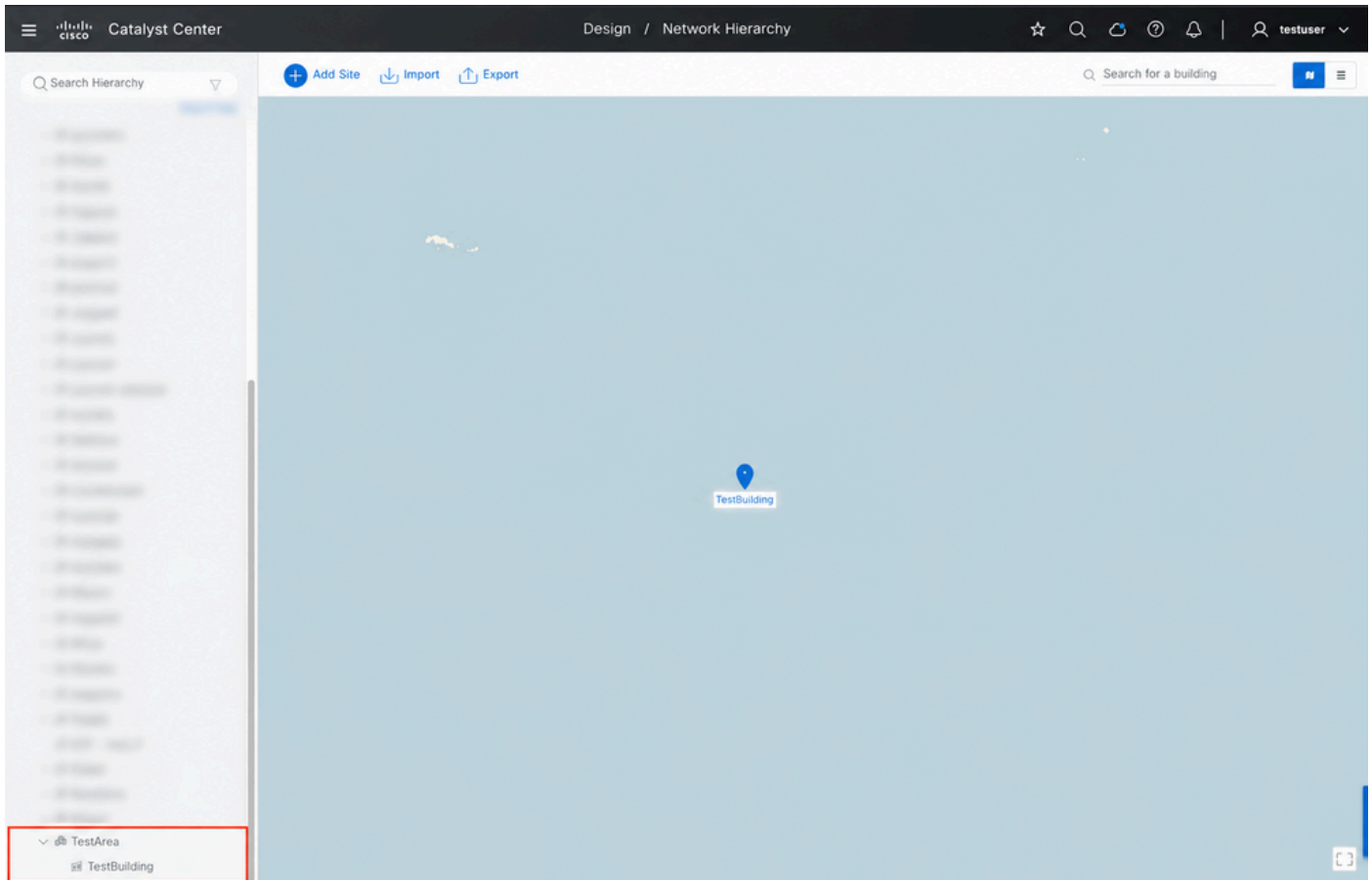
Inserimento delle intestazioni globali per la raccolta API in Bruno.

4. Fare clic su Save (Salva).

A questo punto, il token di autenticazione è stato generato per eseguire qualsiasi altra richiesta API di Catalyst Center.

Esempio 1: Creazione di un piano del sito

In questo esempio viene illustrato come creare un piano all'interno della gerarchia di rete (Globale > AreaTest > TestBuilding).

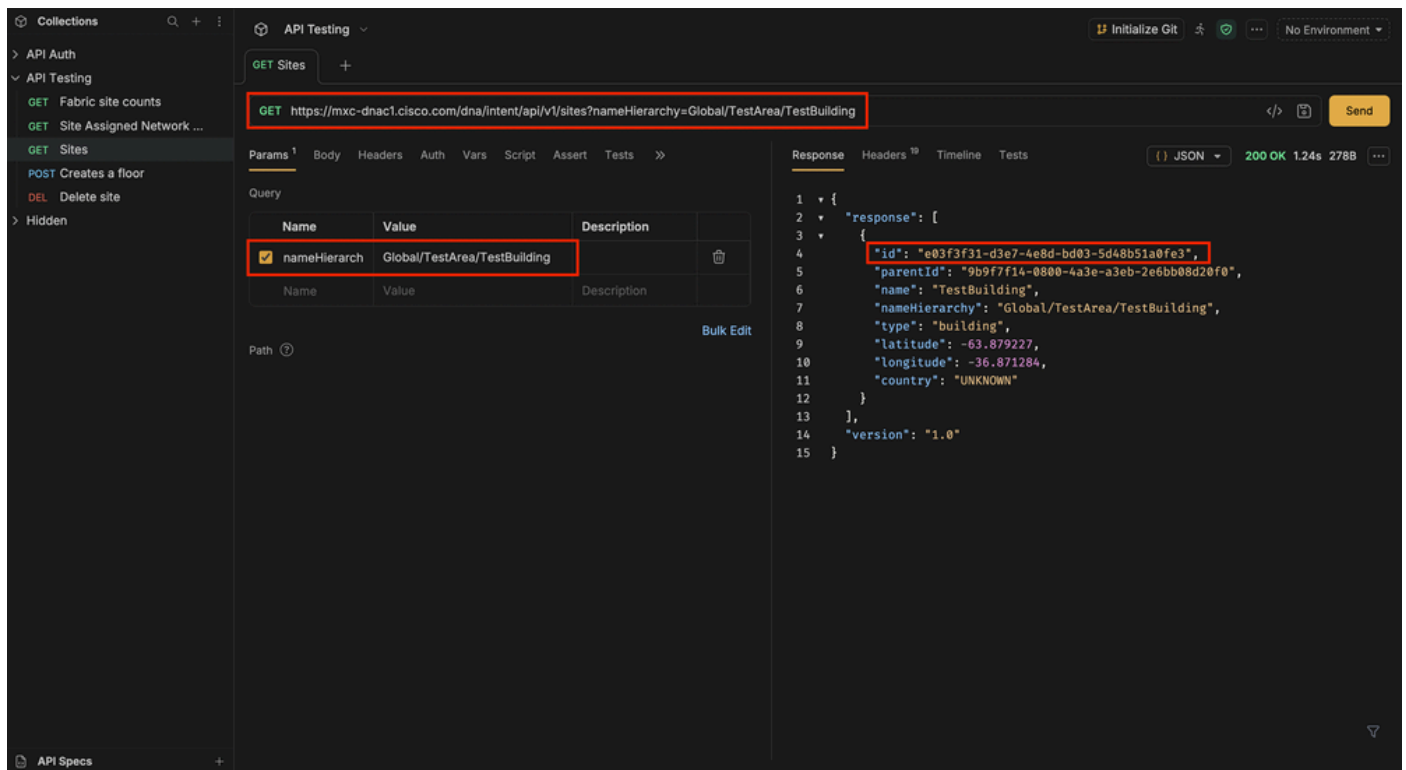


Gerarchia di rete

Passaggio 1: Ottieni ID padre

Utilizzare l'API GET `/dna/intent/api/v1/sites` per trovare l'ID di TestBuilding.

- Parametro query: `nameHierarchy=Globale/AreaTest/GenerazioneTest`
- copiare l'ID dalla risposta; questo è il tuo `parentId`.



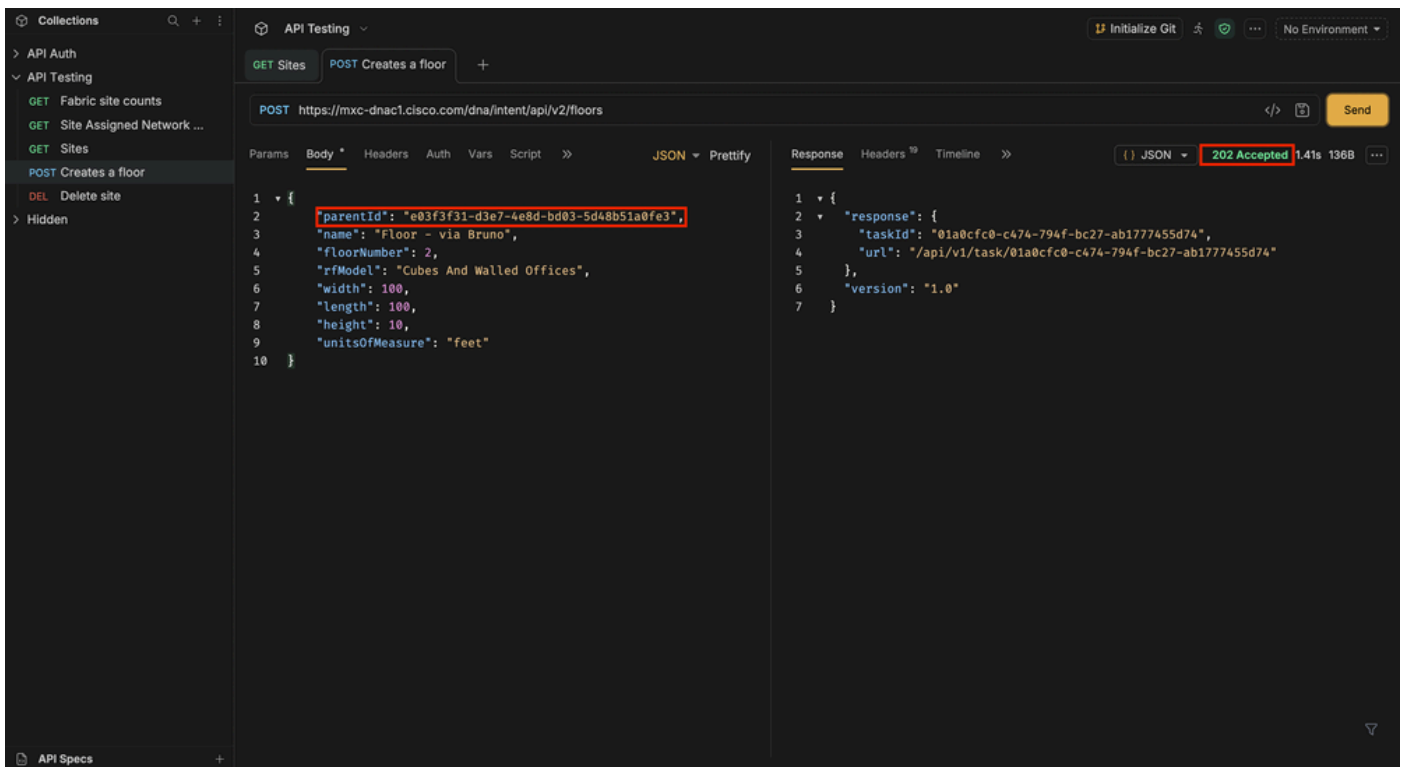
Richiesta GET per `/dna/intent/api/v1/sites?nameHierarchy=Global/TestArea/TestBuilding`

Passaggio 2: Creazione della base

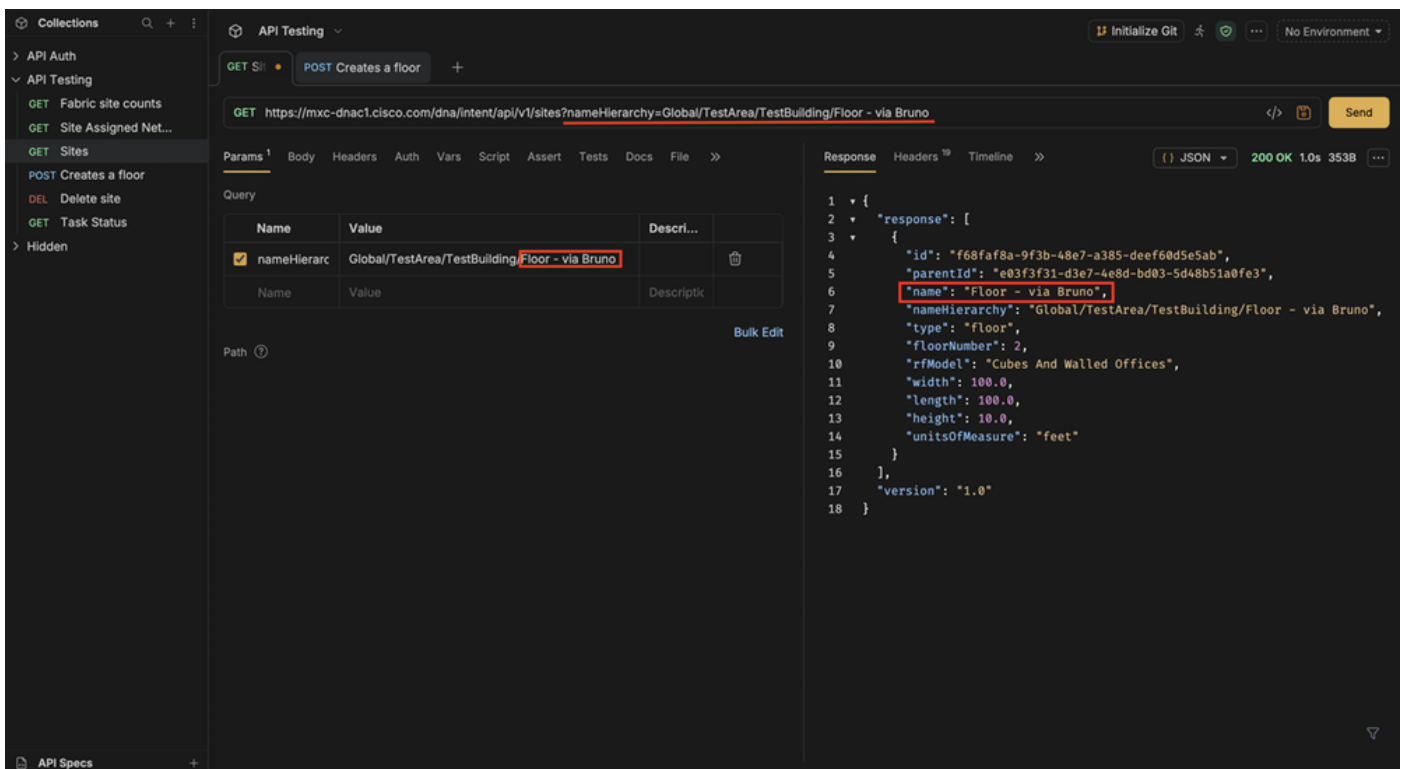
1. Creare una richiesta POST per `/dna/intent/api/v2/floors`.
2. Nella scheda Body (JSON), immettere i parametri richiesti:

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

3. Fare clic su Invia. Una risposta di successo conferma la creazione del pavimento.



Richiesta POST per creare la base



API GET per convalidare la creazione di pavimenti tramite una richiesta POST.

Infine, per verificare che la creazione della base sia stata completata correttamente, è possibile eseguire l'API precedente e modificare la query in modo che corrisponda al percorso gerarchico completo, come illustrato in precedenza.

Esempio 2: Recupera informazioni interfaccia

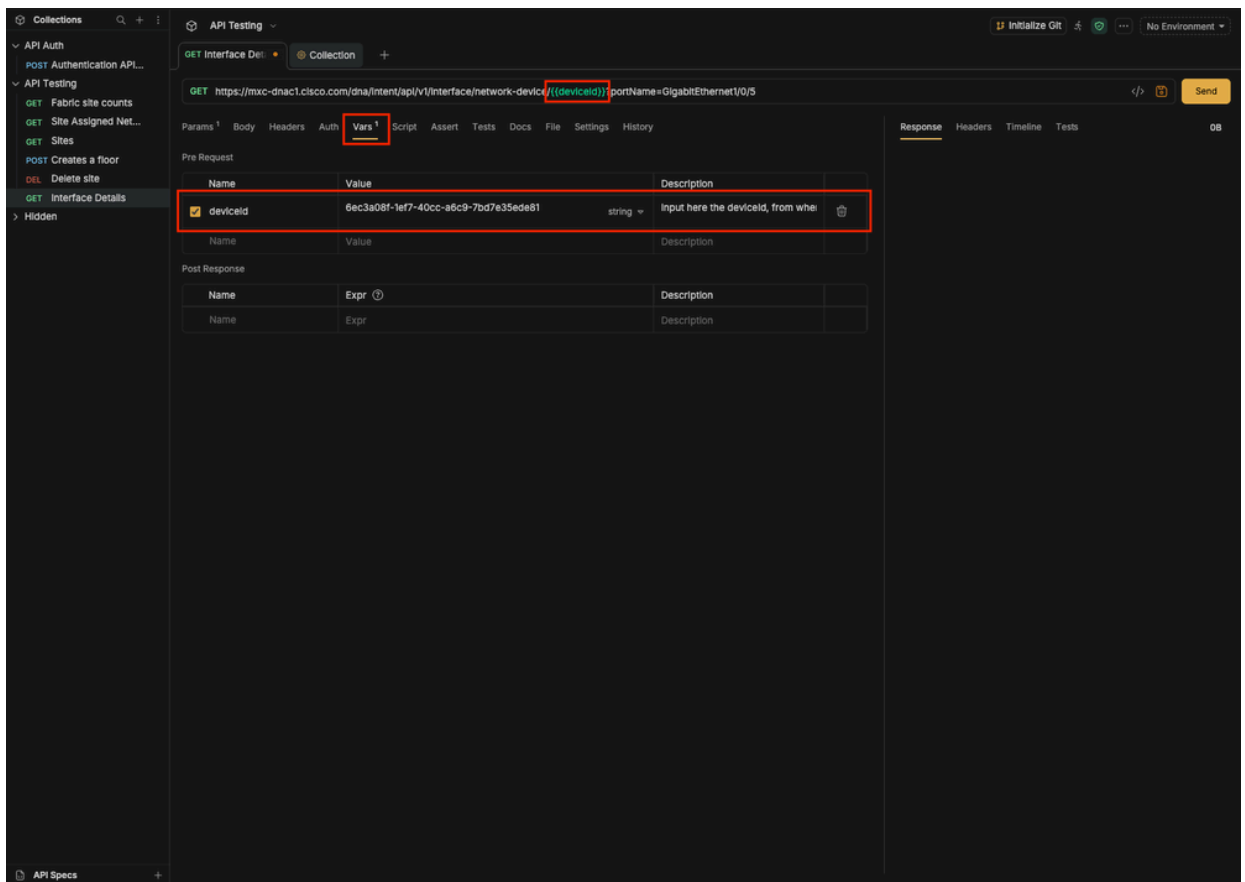
Nell'esempio viene mostrato come recuperare lo stato di un'interfaccia specifica su un dispositivo di rete.

Passaggio 1: Identificare l'ID del dispositivo

Nell'interfaccia utente di Catalyst Center, andare a Inventario, fare clic sul dispositivo e controllare l'URL nel browser. L'ultimo segmento è l'ID del dispositivo.

Passaggio 2: Creare una richiesta con parametri in Bruno

1. Creare una richiesta GET per `/dna/intent/api/v1/interface/network-device/{{deviceId}}`.
2. Nella scheda Vars della richiesta, aggiungere una variabile denominata `deviceId` e incollare l'ID.
3. Per filtrare una porta specifica, aggiungere un parametro Query:
 - Chiave: nome
 - Valore: Gigabit Ethernet 1/0/5



Creazione di variabili in Bruno

4. Fare clic su Invia.

The screenshot shows the API Testing interface with a GET request to the Cisco DNA Intent API. The request URL is `https://mx-c-dnac1.cisco.com/dna/intent/api/v1/interface/network-device/{deviceId}?portName=GigabitEthernet1/0/5`. The response is a JSON object containing details about the network interface.

Name	Value	Description
portName	GigabitEthernet1/0/5	

```
1 {
2   "response": [
3     {
4       "addresses": [],
5       "adminStatus": "up",
6       "className": null,
7       "deviceId": "6ce3a08f-1ef7-40cc-a6c9-7bd7e358de81",
8       "duplex": "FullDuplex",
9       "ifIndex": "12",
10      "interfaceType": "Physical",
11      "ipv4Address": null,
12      "ipv4Mask": null,
13      "isisSupport": "false",
14      "lastIncomingPacketTime": 1786685566000,
15      "lastOutgoingPacketTime": 1790265622000,
16      "lastUpdated": null,
17      "macAddress": "000000000000",
18      "mappedPhysicalInterfaceId": null,
19      "mappedPhysicalInterfaceName": null,
20      "mediaType": null,
21      "natu": "1500",
22      "nativeVlanId": "1",
23      "ospfSupport": "false",
24      "pid": "20",
25      "portMode": "access",
26      "portName": "GigabitEthernet1/0/5",
27      "portType": "Ethernet Port",
28      "serialNo": "000000000000",
29      "series": "000000000000",
30      "speed": "1000000",
31      "status": "up",
32      "vlanId": "1",
33      "voiceVlan": "",
34      "description": "",
35      "name": null,
36      "instanceUuid": "18208b55-383c-4970-ab8b-db942252974d",
37      "instanceTenantId": "66571d0b1efef061f865cd5",
38      "id": "18208b55-383c-4970-ab8b-db942252974d"
39    }
40  ],
41  "version": "1.0"
42 }
```

Richiesta GET applicazione di una query

Conclusioni

Le API Catalyst Center Northbound offrono agli amministratori di rete una flessibilità straordinaria. Passando dalla gestione basata su GUI ai workflow basati su API, è possibile automatizzare le attività ripetitive, garantire la coerenza della configurazione e integrare le funzionalità di rete nell'ecosistema IT più ampio.

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).