

Catalyst Center API's begrijpen en gebruiken

Inhoud

[Inleiding](#)

[Achtergrondinformatie](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Achtergrondinformatie](#)

[Inleiding tot HTTP](#)

[Het HTTP-verzoek](#)

[Het HTTP antwoord](#)

[HTTP-methoden \(werkwoorden\)](#)

[HTTP-statuscodes](#)

[REST - Overzicht](#)

[Catalyst Center- en REST-API's](#)

[Een verificatietoken ophalen met Bruno](#)

[1. Identificeer het API-eindpunt](#)

[2. Bruno configureren voor verificatie](#)

[3. Een algemene koptekst instellen in Bruno](#)

[Voorbeeld 1: Maak een site vloer](#)

[Stap 1: De ouder-ID ophalen](#)

[Stap 2: Creëer de vloer](#)

[Voorbeeld 2: Interfacegegevens ophalen](#)

[Stap 1: Identificeer de apparaat-ID](#)

[Stap 2: Maak een geparametriseerd verzoek in Bruno](#)

[Conclusie](#)

Inleiding

In dit document wordt beschreven hoe CaTC Northbound (Intent) API's kunnen worden geopend en gebruikt met behulp van een API-client en de ingebouwde ontwikkelaarstools van het platform.

Achtergrondinformatie

Het behandelt fundamentele HTTP / REST-concepten en biedt praktische voorbeelden voor

automatisering en testen.

Voorwaarden

Vereisten

Cisco adviseert kennis van deze onderwerpen:

- HTTP/1.1-protocollen en statuscodes.
- REST architectonische stijl.
- Basiskennis van Cisco Catalyst Center (CaTC)
- Basiskennis van Bruno (of vergelijkbare API-clients zoals Postman).

Gebruikte componenten

De informatie in dit document is gebaseerd op de volgende hardware- en softwareversies:

- Cisco Catalyst Center versies 2.3.7.x en 3.x
- Bruno versie 4.1.0

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

Catalyst Center Northbound API's (Intent API's) stellen gebruikers in staat om buiten de grafische gebruikersinterface met de controller te communiceren. Dit maakt automatisering van workflows op maat, integratie met tools van derden en efficiënt netwerkbeheer op schaal mogelijk.

Inleiding tot HTTP

Hypertext Transfer Protocol (HTTP) is de basis van datacommunicatie voor het web. Het houdt zich aan een client-servermodel waarbij de client een verzoek verzendt en de server een antwoord

retourneert.

Het HTTP-verzoek

Een HTTP-verzoek bestaat uit vier delen:

1. Aanvraagregel: Hiermee geeft u de methode (inclusief GET en POST), de URI en de protocolversie op (zoals GET /api/v1/nodes HTTP/1.1).
2. Koppen: Geef metagegevens op, zoals verificatiegegevens en inhoudstype.
3. Lege regel: scheidt kopteksten van het lichaam.
4. Berichttekst (optioneel): Bevat gegevens die naar de server zijn verzonden (gebruikelijk in POST/PUT/PATCH).

Het HTTP antwoord

Een HTTP-antwoord weerspiegelt de structuur van de aanvraag:

1. Statusregel: bevat de statuscode (zoals 200 OK).
2. Headers: Details over de server en de responsgegevens.
3. Berichttekst: De daadwerkelijk gevraagde gegevens (meestal in JSON-indeling voor Catalyst Center).

HTTP-methoden (werkwoorden)

- GET: haalt gegevens op uit een resource.
- POST: Creëert een nieuwe resource.
- PUT: Vervangt een bestaande resource.
- PATCH: Hiermee worden gedeeltelijke wijzigingen op een resource toegepast.
- VERWIJDEREN: Verwijdert een resource.

HTTP-statuscodes

HTTP-statuscodes zijn gestandaardiseerde numerieke codes die door een server worden geretourneerd als onderdeel van een HTTP-respons. Ze geven het algemene resultaat van een klantverzoek aan en helpen te begrijpen of het verzoek succesvol was, extra actie vereist of een fout heeft ondervonden.

De codes zijn gegroepeerd in vijf hoofdcategorieën op basis van hun eerste cijfer.

- 1xx codes zijn informatief en geeft aan dat het verzoek is ontvangen en de verwerking wordt voortgezet.
- 2xx codes geven aan dat de aanvraag succesvol is, wat betekent dat de server het verzoek met succes heeft ontvangen, begrepen en verwerkt.
- 3xx-codes worden gebruikt voor omleiding en geven aan dat aanvullende actie kan worden vereist om het verzoek in te vullen.
- 4xx-codes vertegenwoordigen fouten van de client, wat betekent dat er een probleem is met het verzoek dat door de client is verzonden.
- 5xx-codes vertegenwoordigen serverfouten, wat betekent dat de server een probleem heeft ondervonden tijdens het verwerken van een verzoek.

REST - Overzicht

REST (Representational State Transfer) is een architectonische stijl voor het ontwerpen van netwerktoepassingen. RESTful API's gebruiken HTTP als het transportmechanisme en gebruiken JSON meestal voor gegevensuitwisseling vanwege het lichte karakter ervan.

Wanneer een client interageert met een REST API, geeft de server een weergave van de huidige status van een resource. Deze representatie wordt overgedragen aan de klant als onderdeel van de reactie en kan in een ander formaat worden uitgedrukt.

REST is geen protocol. In plaats daarvan is het een hiërarchische stijl die een reeks principes en beperkingen definieert voor het bouwen van gedistribueerde systemen en API's. In moderne API's is JSON een van de meest voorkomende formaten omdat het licht en gemakkelijk te verwerken is.

Een typische REST-interactie houdt vast aan een clientgebaseerd model (dit werd eerder vermeld in het artikel). De client verzendt een HTTP-verzoek met methoden zoals GET, POST, PUT, PATCH of DELETE. De server verwerkt en retourneert een HTTP-antwoord met de gevraagde informatie of het resultaat van de bewerking.

RESTful API's gebruiken HTTP vaak als het communicatiemechanisme voor het uitvoeren van bewerkingen op resources. In dit model identificeert de client een resource/eindpunt en gebruikt vervolgens de HTTP-methode om aan te geven welke actie moet worden uitgevoerd op die resource. Een REST API kan worden beschreven met behulp van verschillende hoofdelementen. Ten eerste wordt URI gebruikt om de bron of het eindpunt te identificeren waartoe de client toegang wil hebben. De URI geeft de locatie of identificatie van die bron binnen de API. Ten tweede gebruikt de client standaard HTTP-methoden zoals GET, POST, PUT, PATCH en DELETE om de aanvraagbewerking te definiëren. Bijvoorbeeld, GET kan worden gebruikt om een bron op te halen, POST om er een te maken en VERWIJDEREN om er een volledig te verwijderen. Ten slotte wordt de resource uitgewisseld via een representatie, die wordt gedefinieerd door het mediatype. Deze representatie is de vorm waarin de brongegevens worden overgedragen tussen de client en de server, zoals JSON of XML.

Catalyst Center- en REST-API's

CaTC helpt eenvoudigweg de netwerkactiviteiten op de campus door automatisering, configuratie, beleidsbeheer, bewaking en analyse te centraliseren. Ik sta netwerkbeheerders toe om apparaten efficiënter te implementeren en te beheren, consistent beleid te handhaven, netwerkgedrag te bewaken en de algehele operationele zichtbaarheid te verbeteren. Deze mogelijkheden zijn toegankelijk via de grafische gebruikersinterface (GUI) en via API's en externe integraties.

Het CaTC-platform biedt ook API's die kunnen worden gebruikt om aangepaste integraties te bouwen en netwerkactiviteiten te automatiseren. Door CaTC API's te combineren met bestaande Cisco Solutions of applicaties van derden, kunnen organisaties de functionaliteit van het platform uitbreiden en op maat gemaakte workflows maken voor specifieke activiteiten of zakelijke vereisten.

Na het behandelen van de basisconcepten, richt dit document zich op Northbound API's, ook wel Intent API's genoemd, binnen Cisco Catalyst Center. Deze API's stellen specifieke Catalyst Center-mogelijkheden bloot door middel van een beleidsgebaseerde abstractie van zakelijke intenties, waardoor beheerders en ontwikkelaars zich kunnen concentreren op het gewenste resultaat in plaats van de individuele configuratiestappen die nodig zijn om dit te bereiken.

Catalyst Center Intent API's zijn RESTful API's die het JSON-gegevensformaat gebruiken om met het platform te communiceren. Ze kunnen worden gebruikt om netwerkinformatie te ontdekken, operationele gegevens op te halen en ondersteunde besturings- en automatiseringstaken in het netwerk uit te voeren.

Een verificatietoken ophalen met Bruno

Voordat u API-oproepen uitvoert, moet u zich verifiëren om een token te ontvangen.

1. Identificeer het API-eindpunt

1. Log in bij Catalyst Center.
2. Navigeer naar Menu > Platform > Developer Toolkit.
3. Selecteer in het linkermenu de optie Verificatie > Verificatie-API.

The screenshot shows the Catalyst Center Developer Toolkit interface. On the left is a sidebar with a search bar and a list of categories: Appliance, Authentication (highlighted with a red box), No Subdomain, Cisco DNA Center System, Connectivity, Ecosystem Integrations, Event Management, Integrations, Know Your Network, Operational Tasks, Policy, Site Management, System, and System Settings. The main area is titled 'Authentication' and contains a table of APIs. The table has columns for Method, Name, Description, URL, and Actions. Three APIs are listed: 'importCertificate' (POST, /dna/intent/api/v1/certificate), 'importCertificateP12' (POST, /dna/intent/api/v1/certificate-p12), and 'Authentication API' (POST, /dna/system/api/v1/auth/token). The 'Authentication API' row is highlighted with a red box. Below the table, it says 'Cisco DNA Center System'.

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. Kopieer de URL (meestal POST /dna/system/api/v1/auth/token).

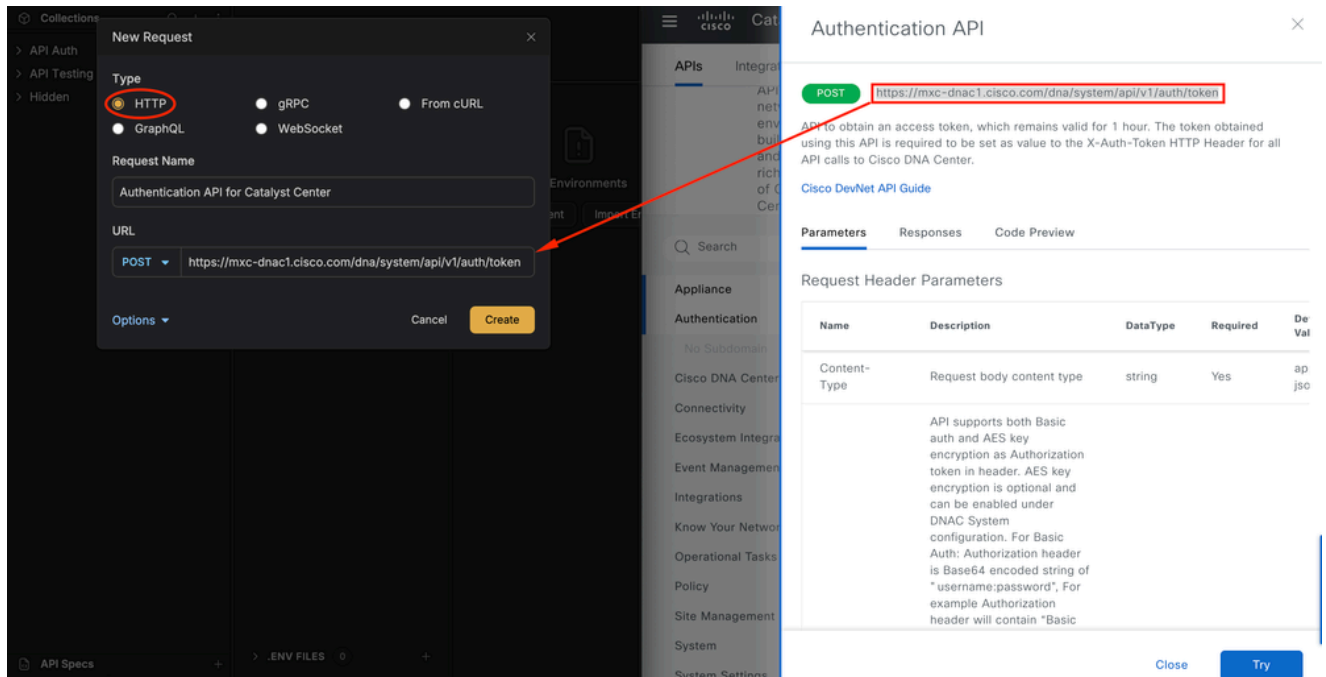
2. Bruno configureren voor verificatie

1. Maak in Bruno een nieuw verzoek.
2. Stel het type in op HTTP en de methode op POST.

The screenshot shows the Bruno API client interface. On the left is a sidebar with a search bar and a list of collections: API Auth, API Testing, and Hidden. The main area is titled 'My Workspace' and contains a menu with options: New Request (highlighted with a red box), 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 with a red box.

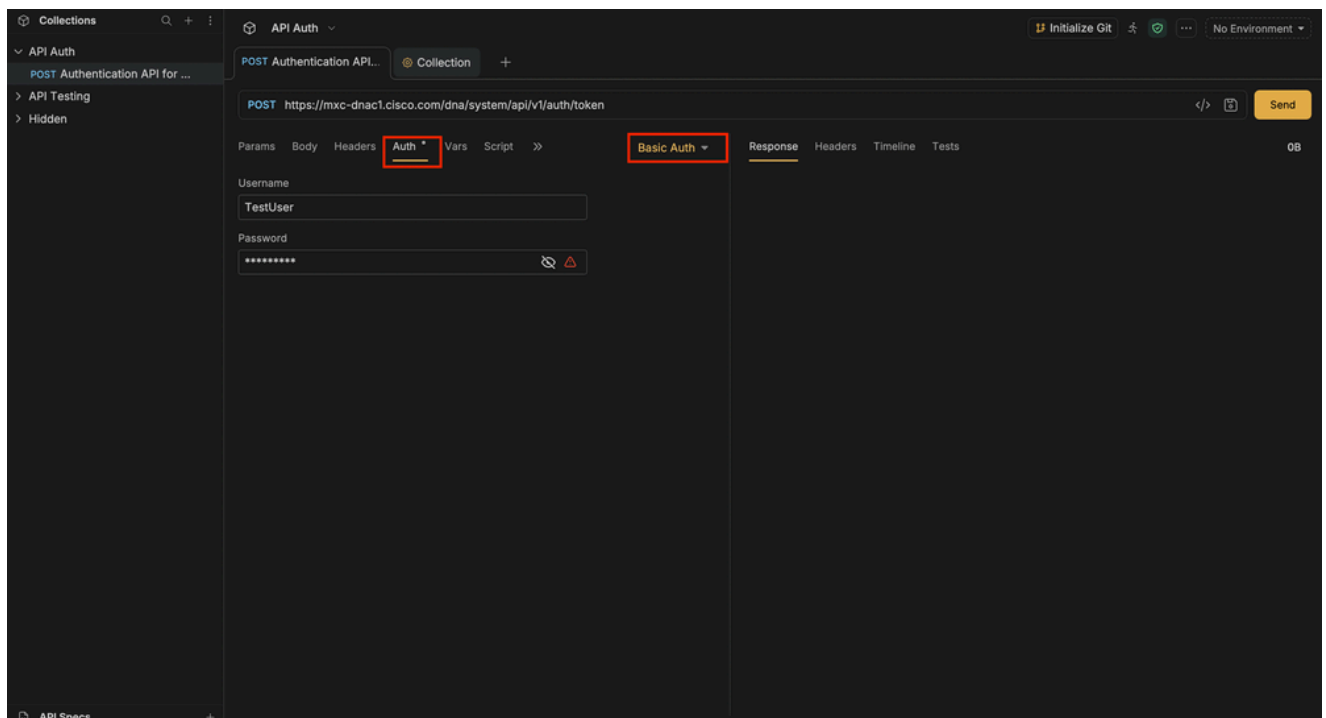
Een nieuw verzoek indienen in Bruno

3. Voer de URL voor de verificatie-API in.



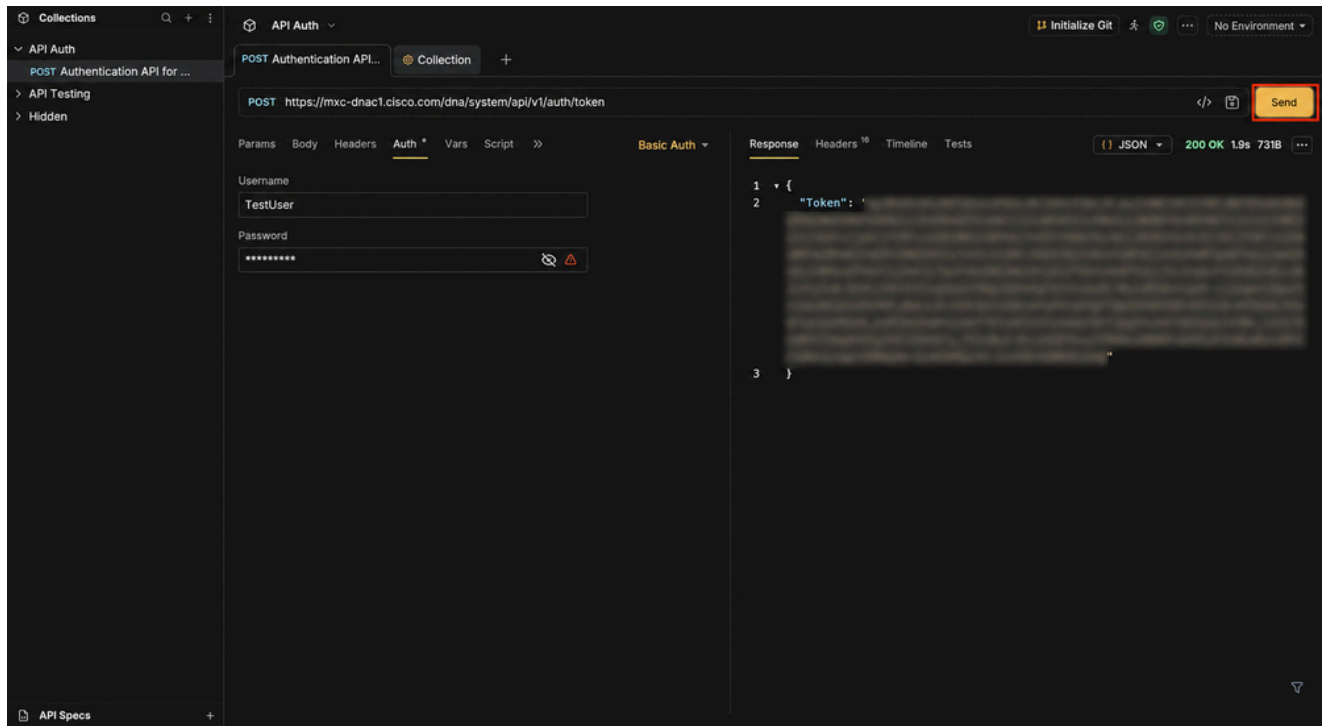
Een nieuw verzoek indienen in Bruno

4. Navigeer naar het tabblad Auth en selecteer Basic Auth.



Basisverificatie voor verificatie-API

5. Voer uw Catalyst Center SUPER ADMIN-referenties in.
6. Klik op Verzenden.



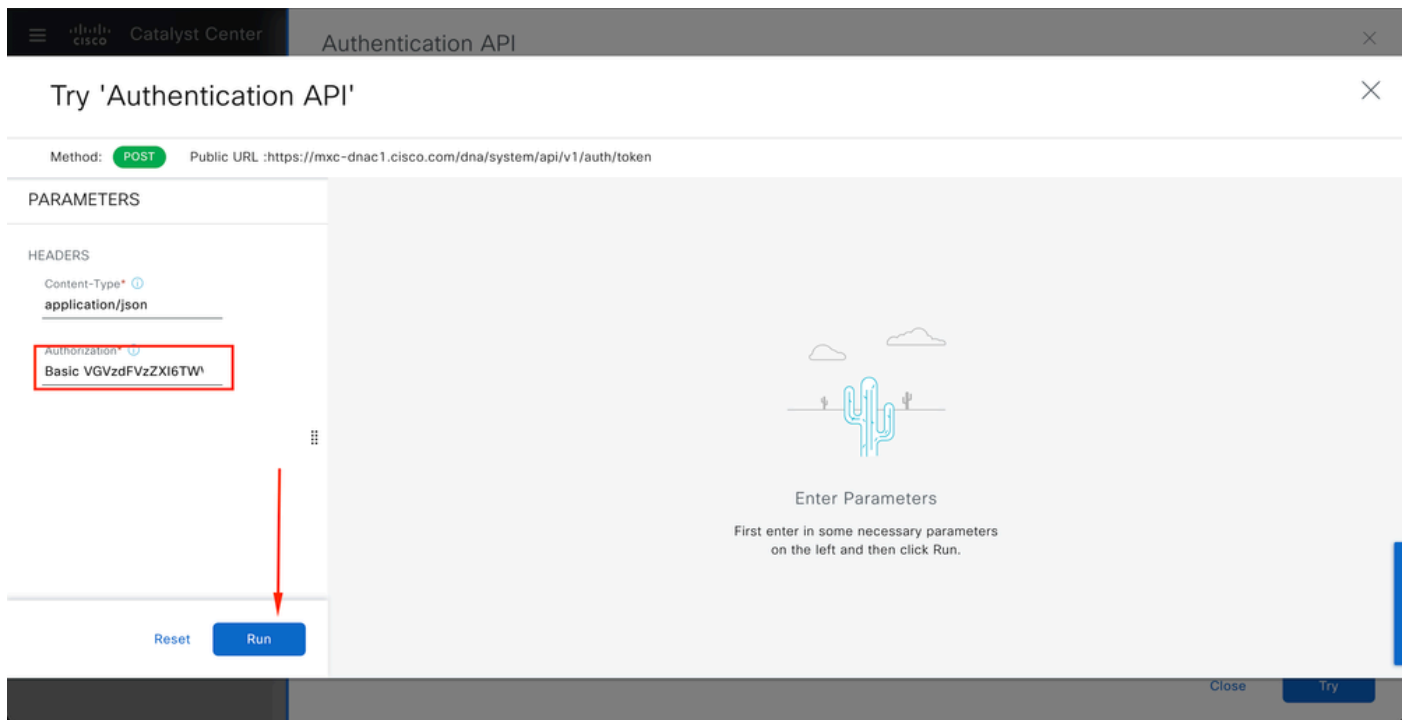
De verificatie-API uitvoeren

De server geeft een 200 OK-status en een JSON-payload met het token.

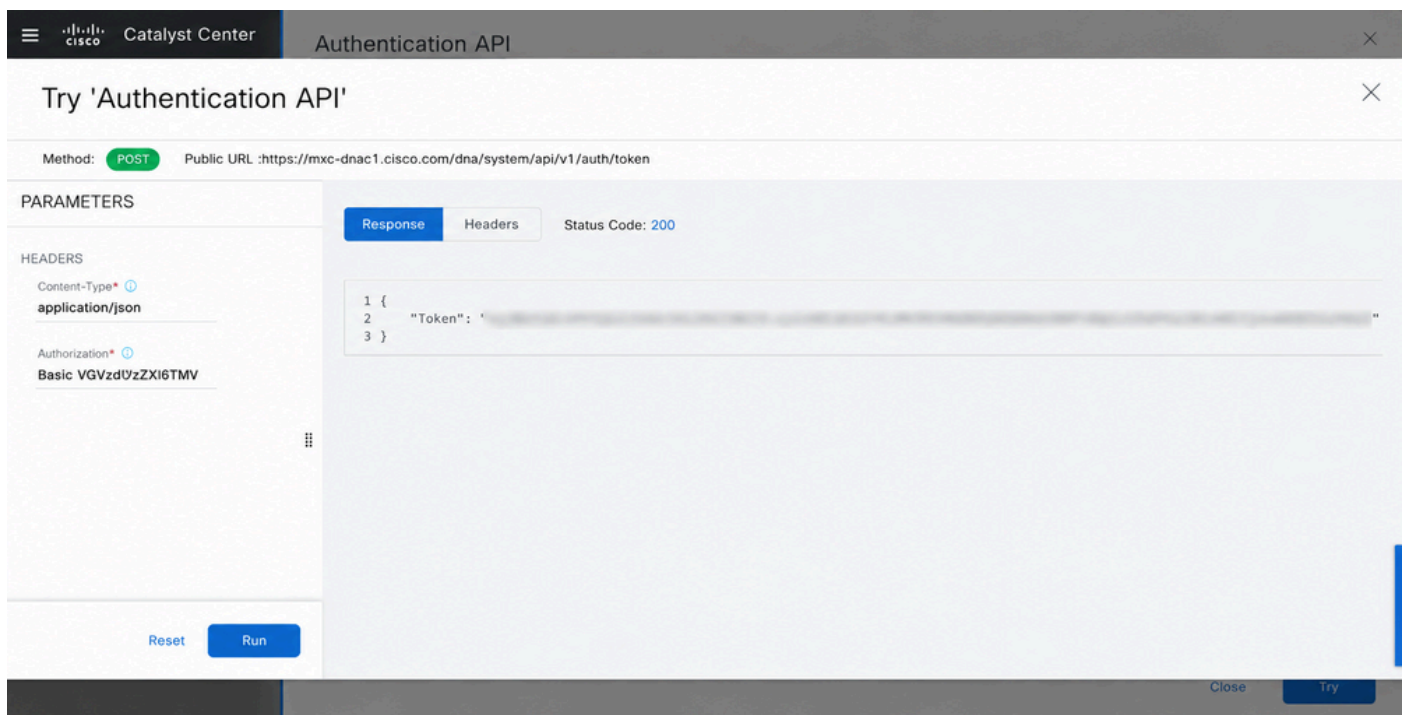
Alternative (Terminal): You can also generate the token by base64-encoding your `username:password` and

```
[root@localhost ~]# % echo -n 'TestUser:VGvzdFVzZXI6VGvzdDEyMyEhIQ==' | base64
VGvzdFVzZXI6VGvzdDEyMyEhIQ==
```

De base64-codering voor de gebruikersnaam en het wachtwoord ophalen



Invoeren van Basic Auth met gecodeerde base64 gebruikersnaam en wachtwoord



Verificatietoken verkrijgen van Catalyst Center GUI.

Op dit punt hebt u het verificatietoken, hetzij op de Bruno-reactie of op de reactie die is uitgevoerd vanuit de Catalyst Center Developer Toolkit.

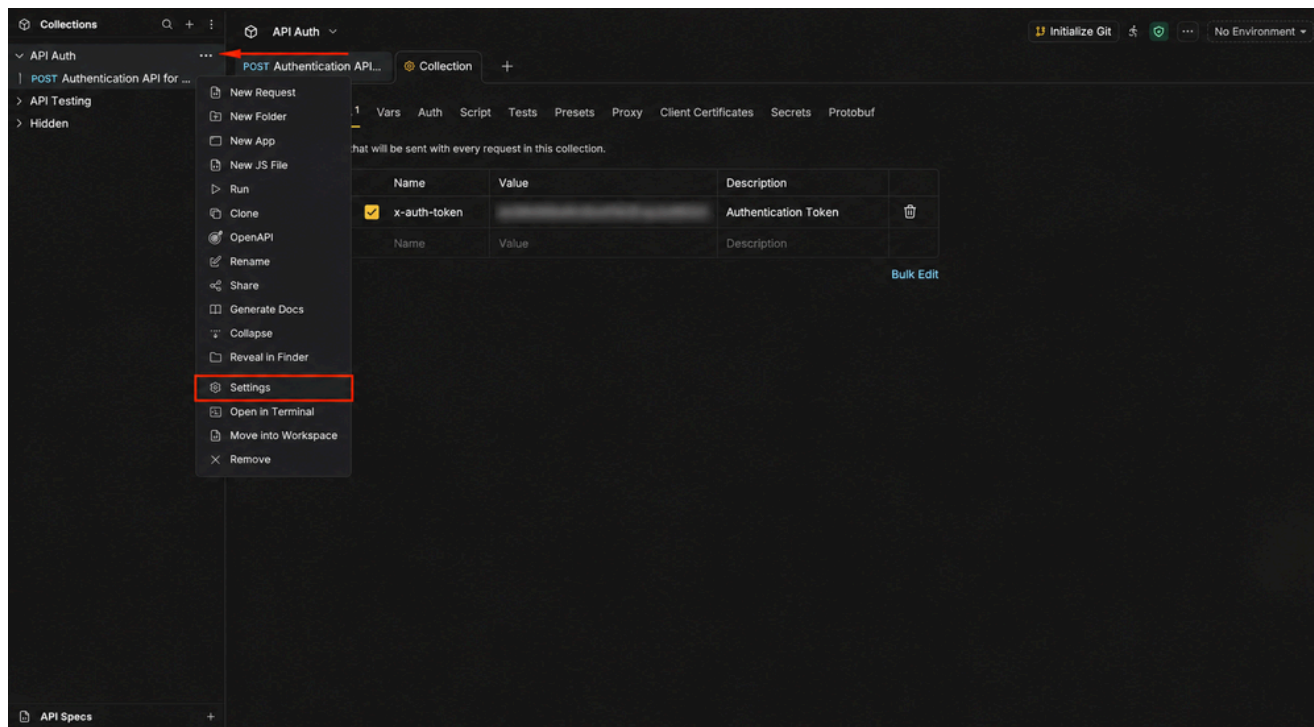
Voeg nu het token toe als een header op de verzamelinstellingen in Bruno, dit vergemakkelijkt elk ander gemaakt verzoek binnen de verzameling om het token te gebruiken, zodat u het token niet

handmatig hoeft in te voeren als een header voor elk verzoek.

3. Een algemene koptekst instellen in Bruno

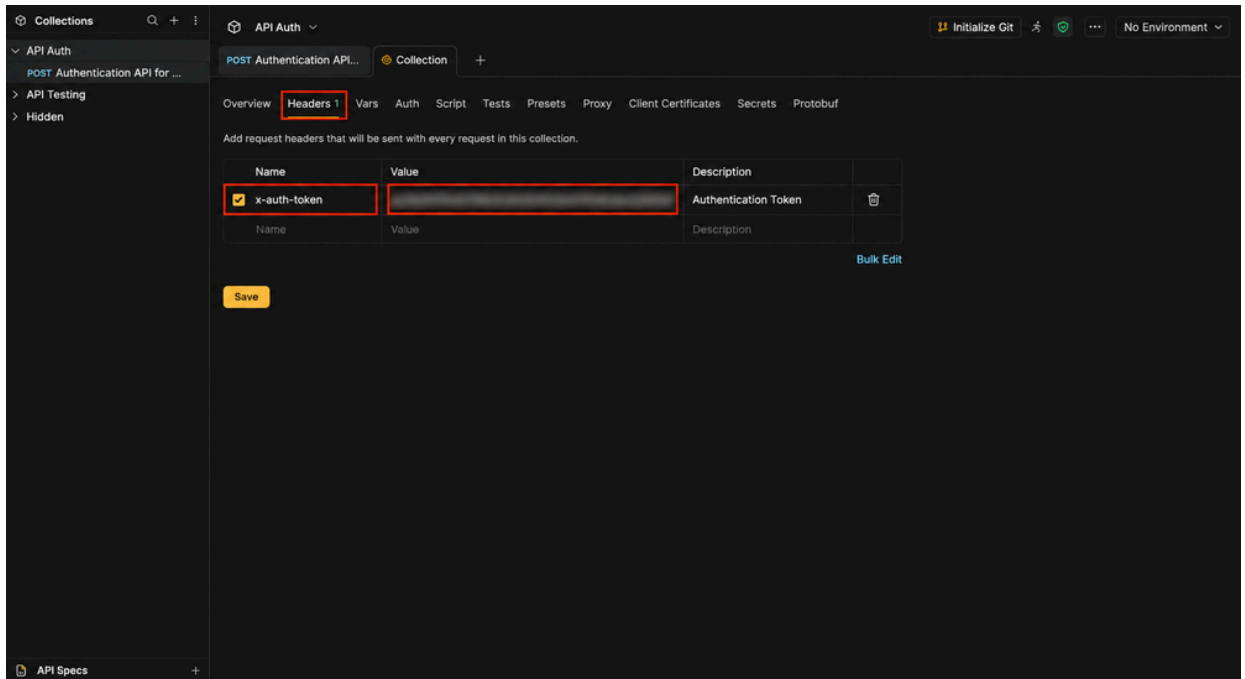
Om herverificatie voor elk verzoek te voorkomen, stelt u het token in als een koptekst op collectieniveau:

1. Klik met de rechtermuisknop op uw verzameling en selecteer Instellingen.



Verzamelinstellingen

2. Ga naar het tabblad Headers.
3. Een nieuwe header toevoegen:
 - Naam: x-auth-token
 - Value: [Plak je token hier]



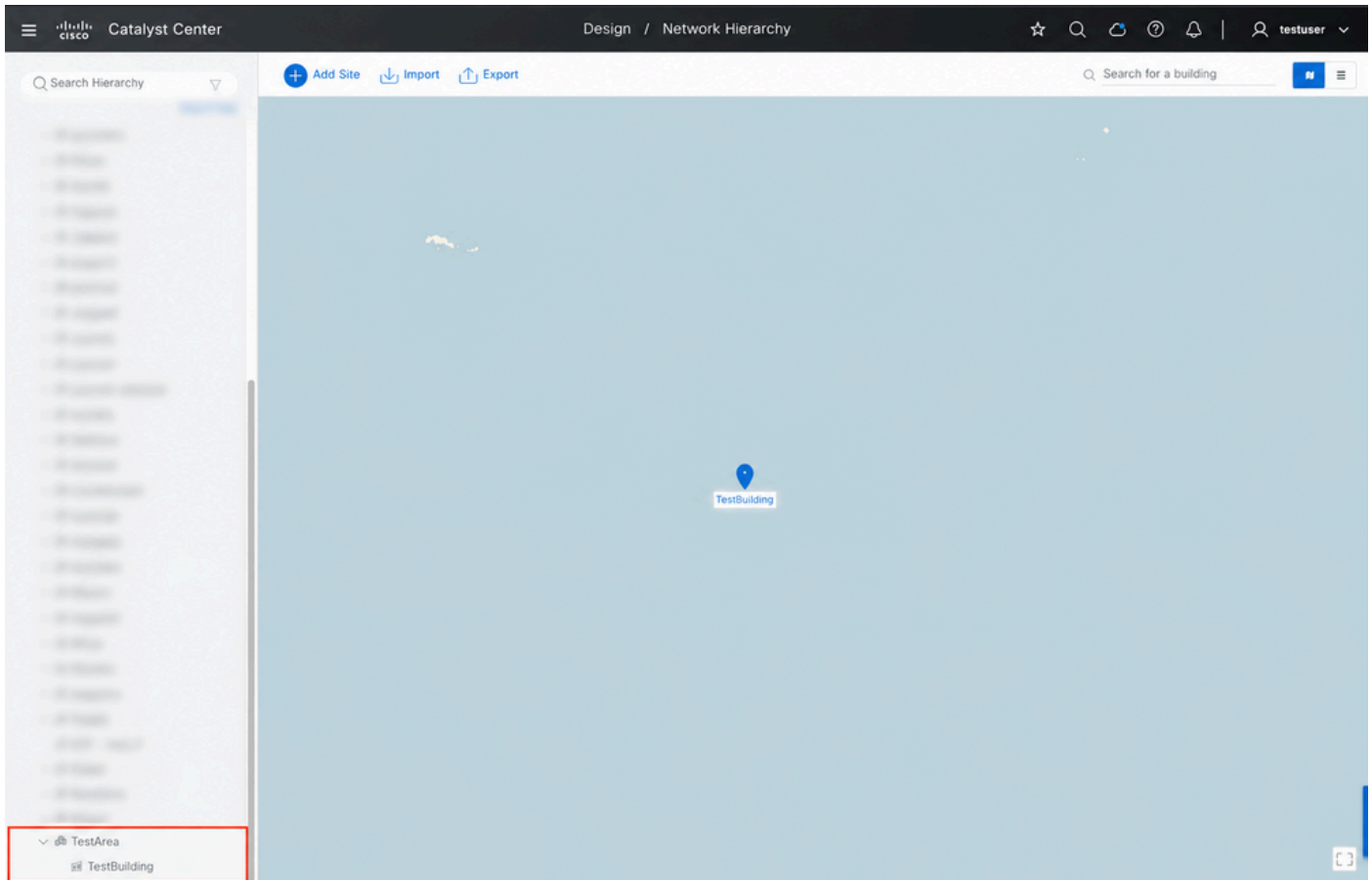
Het invoeren van globale headers voor de API Collection in Bruno.

4. Klik op Save (Opslaan).

Op dit punt hebt u het verificatietoken gegenereerd om andere Catalyst Center API-verzoeken uit te voeren.

Voorbeeld 1: Maak een site vloer

In dit voorbeeld wordt getoond hoe u een vloer maakt binnen de netwerkhierarchie (Globaal > TestArea > TestBuilding).

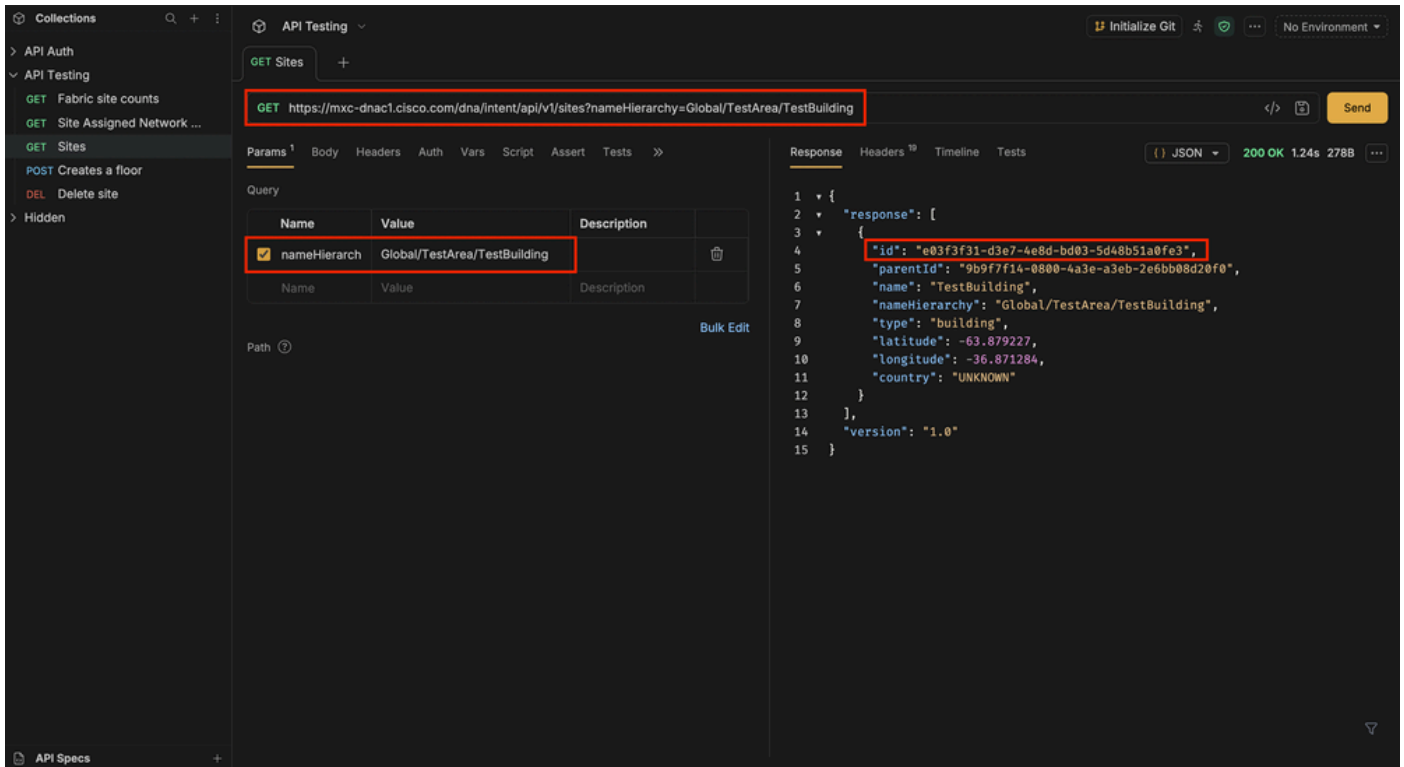


netwerkhierarchie

Stap 1: De ouder-ID ophalen

Gebruik de API GET `/dna/intent/api/v1/sites` om de ID van TestBuilding te vinden.

- Queryparameter: `nameHierarchy=Global/TestArea/TestBuilding`
- Kopieer de ID uit het antwoord; dit is je ouder-ID.



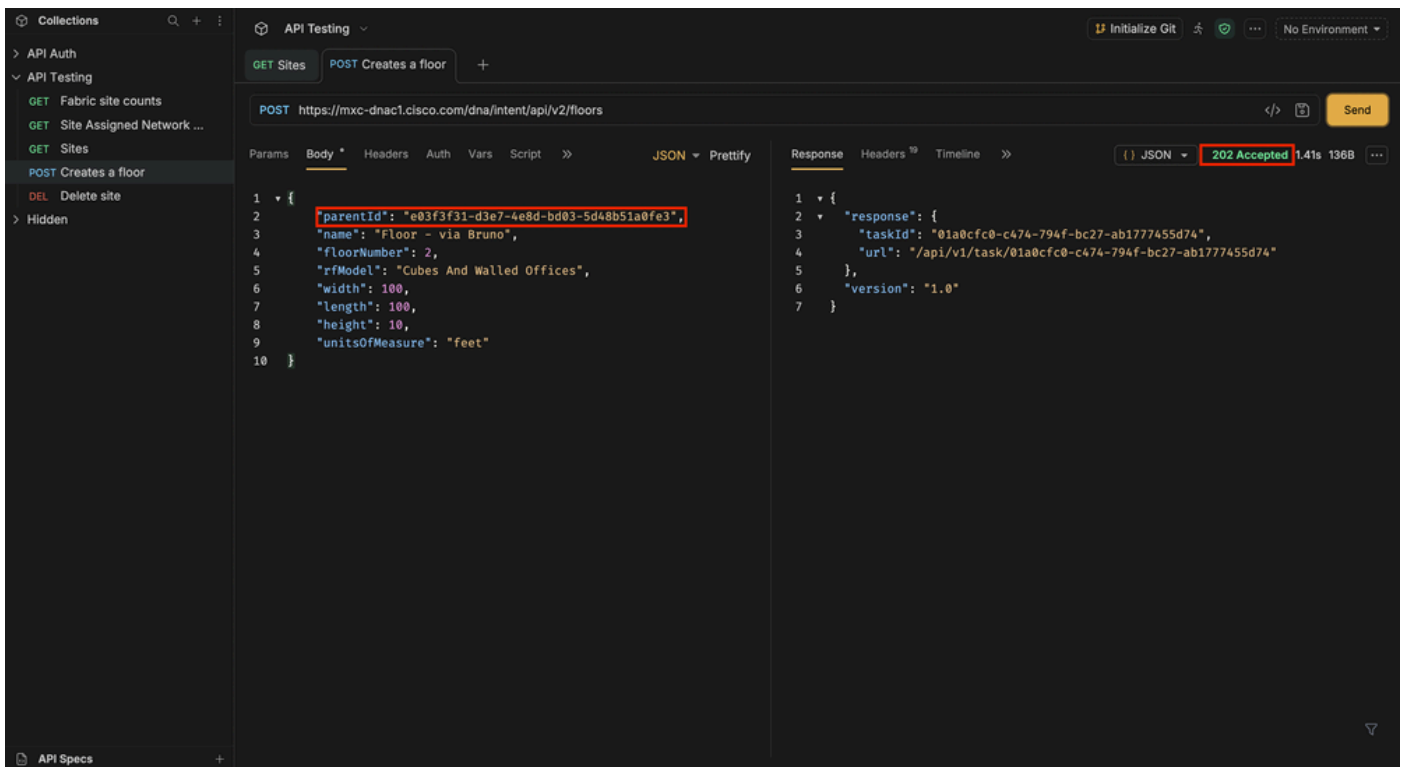
VRAAG NAAR /dna/intent/api/v1/sites?nameHierarchy=Global/TestArea/TestBuilding

Stap 2: Creëer de vloer

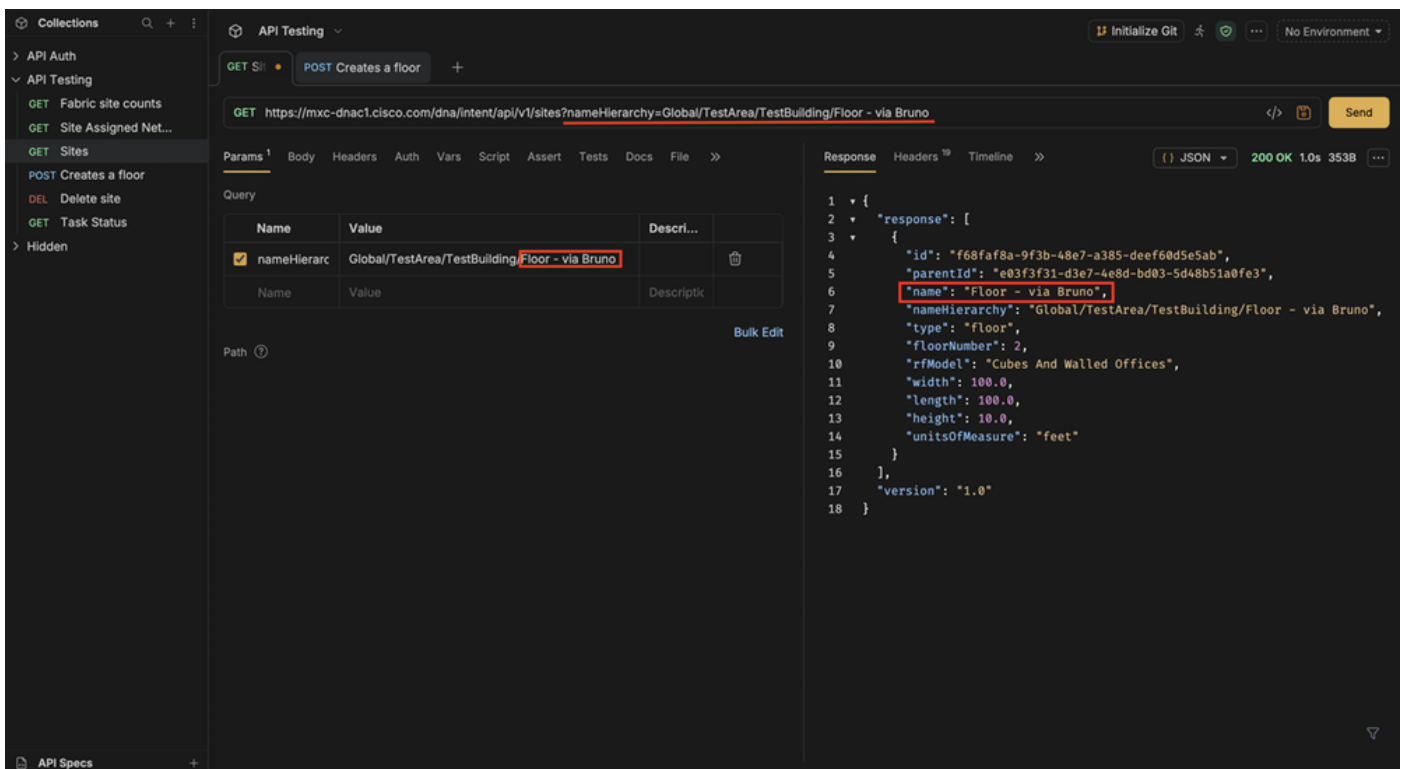
1. Maak een POST-aanvraag voor /dna/intent/api/v2/vloeren.
2. Voer op het tabblad Body (JSON) de vereiste parameters in:

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

3. Klik op Verzenden. Een succesvolle reactie bevestigt de vloercreatie.



POST verzoek om de vloer te creëren



KRIJG API om de creatie van de vloer te valideren via een POST-aanvraag.

Ten slotte, om te valideren dat de vloer met succes is gemaakt, kunt u de vorige API uitvoeren en de query wijzigen om het volledige hiërarchische pad te matchen, zoals hierboven wordt weergegeven.

Voorbeeld 2: Interfacegegevens ophalen

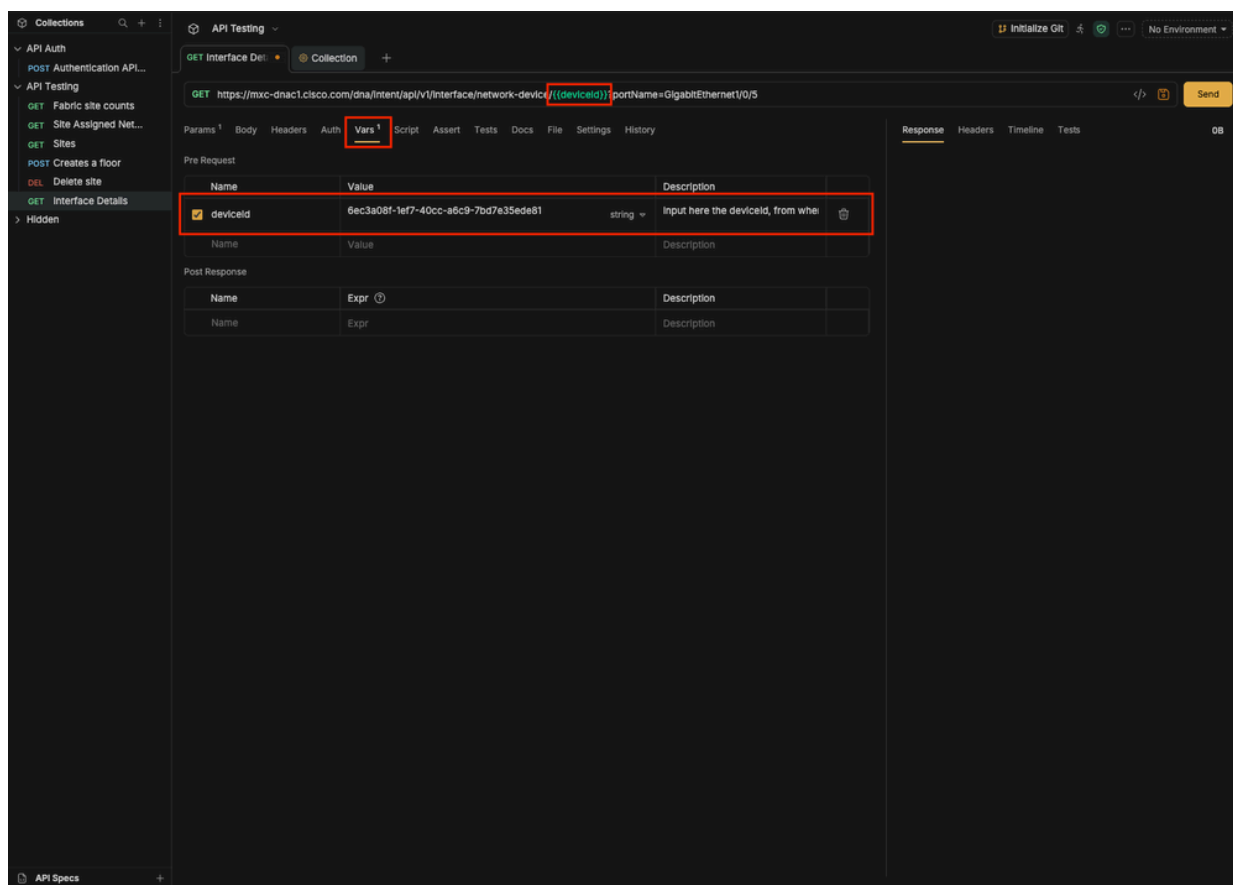
Dit voorbeeld laat zien hoe u de status van een specifieke interface op een netwerkapparaat kunt ophalen.

Stap 1: Identificeer de apparaat-ID

Ga in de Catalyst Center GUI naar Inventory, klik op uw apparaat en bekijk de URL in uw browser. Het laatste segment is de Device ID.

Stap 2: Maak een geparametriseerd verzoek in Bruno

1. Maak een GET-verzoek voor `/dna/intent/api/v1/interface/network-device/{{deviceid}}`.
2. Voeg op het tabblad Vars van het verzoek een variabele met de naam `deviceid` toe en plak uw ID.
3. Als u wilt filteren op een specifieke poort, voegt u een queryparameter toe:
 - Sleutel: naam
 - Waarde: Gigabit Ethernet1/0/5



Variabelen maken in Bruno

4. Klik op Verzenden.

The screenshot displays the API Testing tool interface. On the left, a sidebar shows a collection of API endpoints under 'API Testing', with 'GET Interface Details' selected. The main panel shows the details of this endpoint, including the URL: `https://mxc-dnac1.cisco.com/dna/intent/api/v1/interface/network-device/{{deviceId}}?portName=GigabitEthernet1/0/5`. The 'Params' tab is active, showing a table with one parameter:

Name	Value	Description
portName	GigabitEthernet1/0/5	

Below the table is a 'Bulk Edit' button. The 'Response' tab shows the JSON output of the request, which is a 200 OK status. The response body is a JSON object containing detailed information about the network interface, including its status, type, and various identifiers.

```
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": "66571d0b1efef061f865c6d5",
38      "id": "18208b55-383c-4970-ab8b-db942252974d"
39    }
40  ],
41  "version": "1.0"
42 }
```

AANVRAAG ONTVANGEN Een query toepassen

Conclusie

Catalyst Center Northbound API's bieden krachtige flexibiliteit voor netwerkbeheerders. Door over te schakelen van GUI-gebaseerd beheer naar API-gestuurde workflows, kunt u repetitieve taken automatiseren, de consistentie van de configuratie garanderen en netwerkintelligentie integreren in uw bredere IT-ecosysteem.

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document ([link](#)) te raadplegen.