Secure Access to Use REST API configureren met Python

Inhoud

Inleiding

Voorwaarden

- Vereisten
- Gebruikte componenten

Configureren

- Een API-sleutel maken
- Python-code
  - Script 1:
  - Script 2:

Problemen oplossen

Gerelateerde informatie

Inleiding

Dit document beschrijft de stappen om API-toegang te configureren en te gebruiken om resources informatie uit de Secure Access te halen.

Voorwaarden

Cisco raadt kennis van de volgende onderwerpen aan:

1. Python 3.x
2. REST API
3. Cisco beveiligde toegang

Vereisten

Aan deze eisen moet worden voldaan voordat verder kan worden gegaan:

- Cisco Secure Access-gebruikersaccount met de rol Full Adminuser
- Cisco Security Cloud Single Sign On-account (SCSO) om in te loggen op Secure Access.

Gebruikte componenten

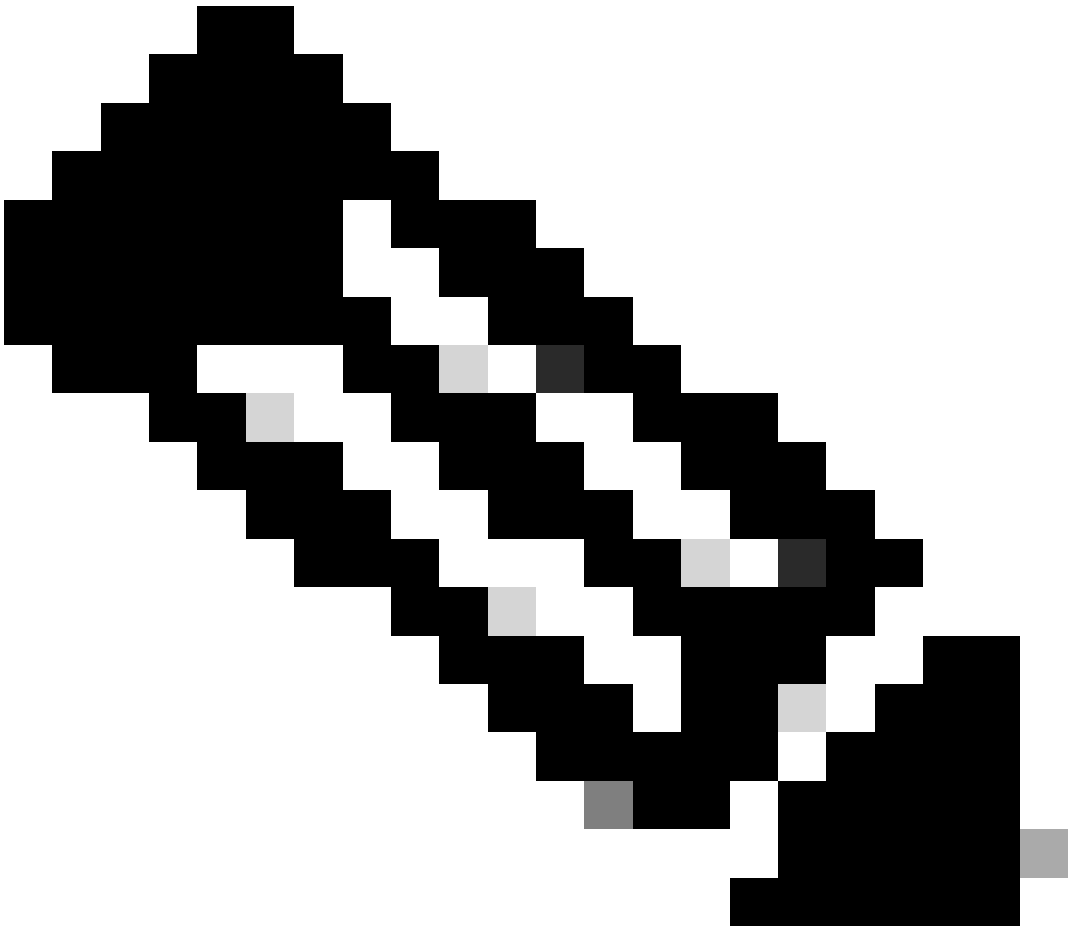
De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Secure Access Dashboard
- Python

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.

Configureren

Secure Access API biedt een standaard REST-interface en ondersteunt de OAuth 2.0 Client Credentials Flow. Meld u aan bij Secure Access en maak uw Secure Access API-toetsen aan om aan de slag te gaan. Gebruik vervolgens uw API-referenties om een API-toegangsteken te genereren.



> Opmerking: API-sleutels, wachtwoorden, geheimen en tokens geven toegang tot uw persoonlijke gegevens. U mag uw referenties nooit delen met een andere gebruiker of organisatie.

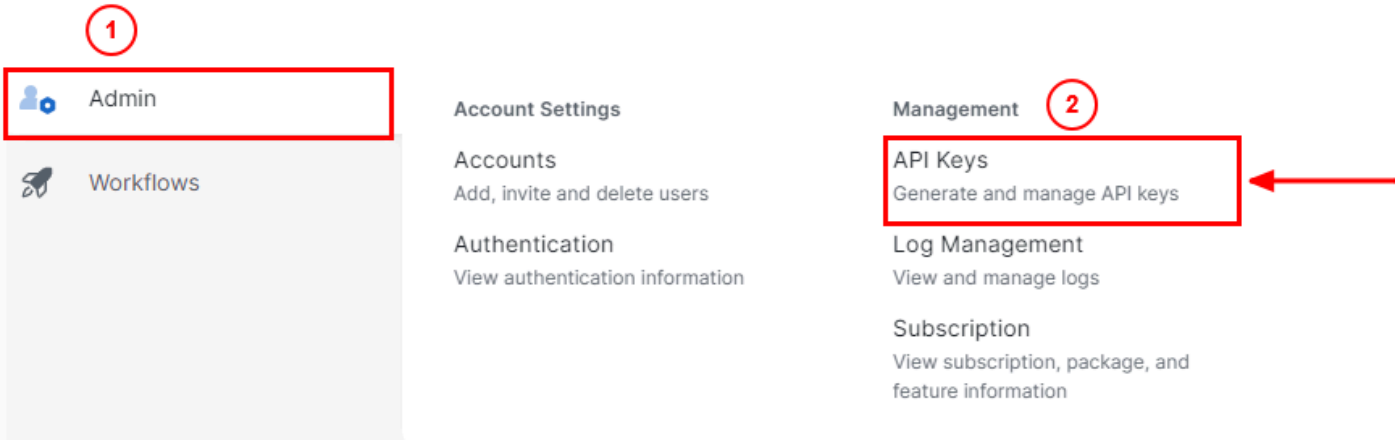
Configureer de API-toets vanuit het Secure Access Dashboard voordat u de in dit artikel

genoemde scripts uitvoert.

Een API-sleutel maken

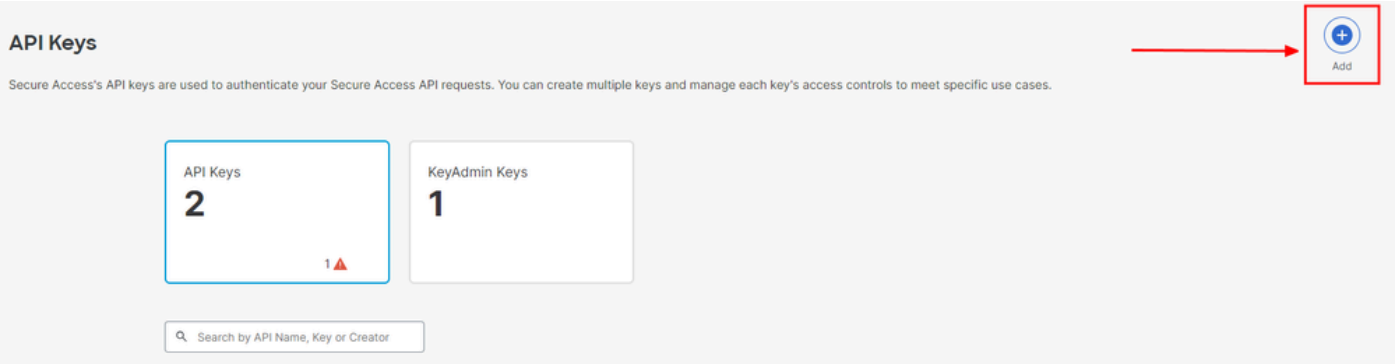
Maak een API-sleutel en geheim met deze stappen. Aanmelden voor beveiligde toegang met de URL: [Secure Access]

1. Selecteer de optie in de knoppenbalk links Admin.
    - Selecteer onder Admin de optie **API Keys**:



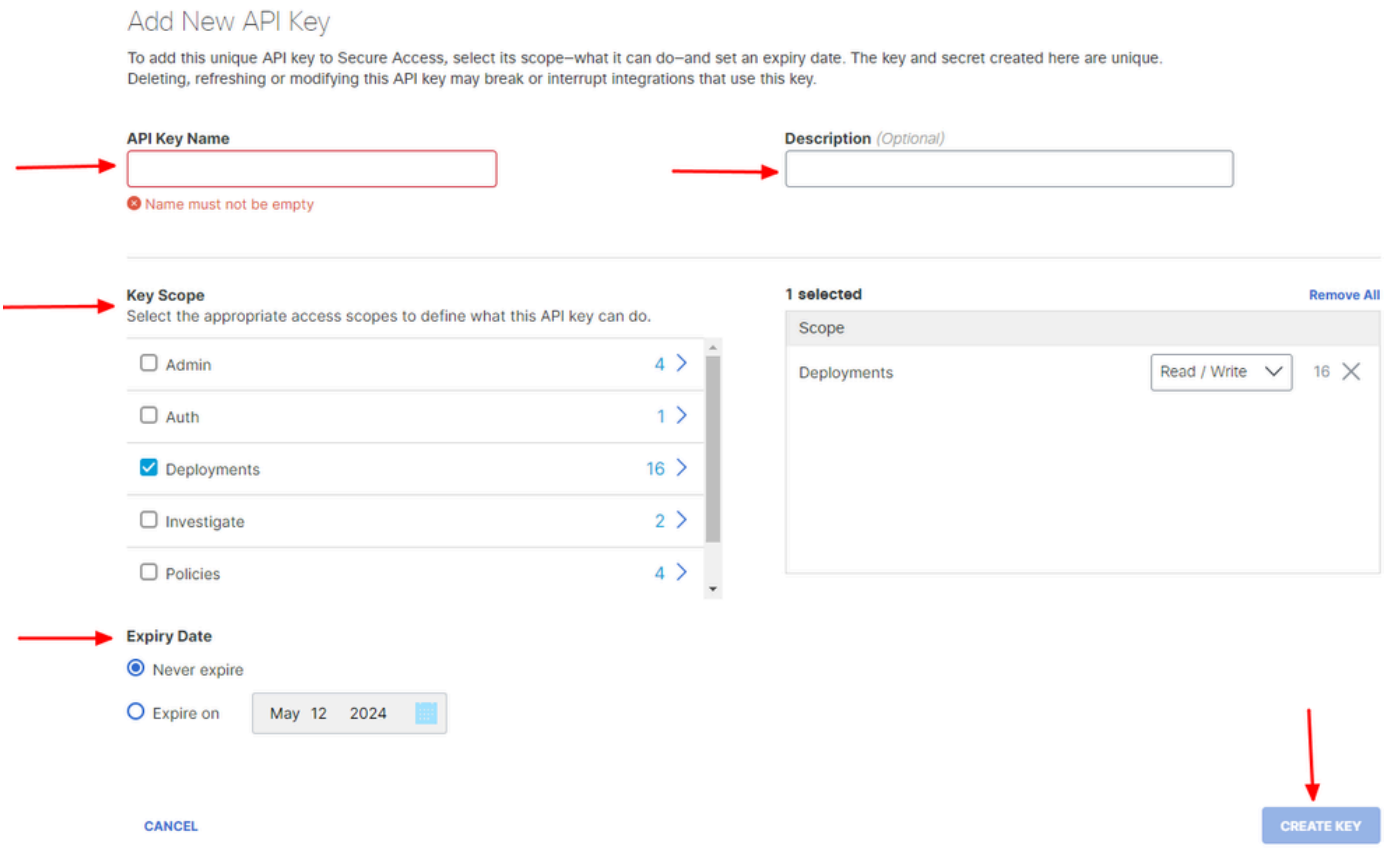
Secure Access Dashboard Admin - API-toetsen

3. Klik in de rechterbovenhoek op de + knop om een nieuwe API-sleutel toe te voegen:



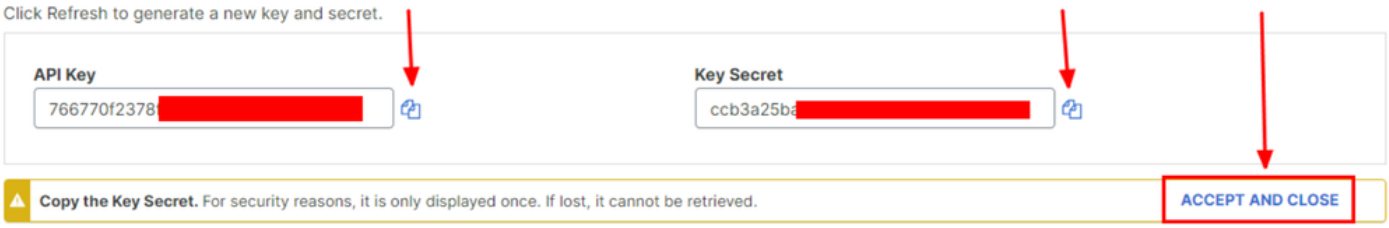
Secure Access - API-sleutel toevoegen

4. Verstrek het **API Key Name, Description**(Optioneel) en selecteer het Key scope en Expiry date volgens uw vereisten. Klik op de knop als u klaar bent **Create**:

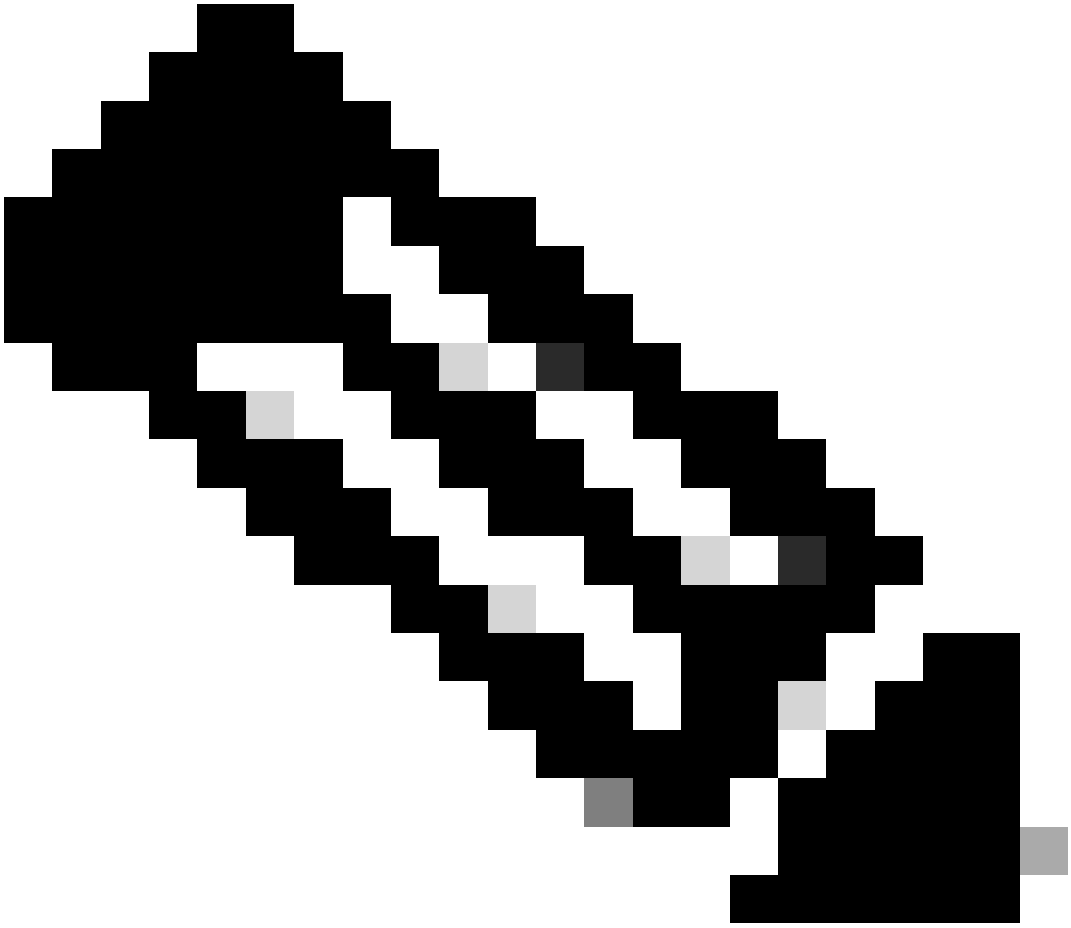


Secure Access - API-toetsgegevens

5. Kopieer de afbeelding API Keyen de afbeelding **Key Secret** en klik op ACCEPT AND CLOSE:



Secure Access - API-sleutel en geheim



Opmerking: er is maar één mogelijkheid om uw API-geheim te kopiëren. Secure Access slaat uw API-geheim niet op en u kunt het na de eerste aanmaak niet ophalen.

Python-code

Er zijn meerdere manieren om deze code te schrijven, in aanmerking genomen dat de gegenereerde token gedurende 3600 seconden (1 uur) geldig is. U kunt twee afzonderlijke scripts maken waarin het eerste script gebruikt kan worden om het Bearer Token te genereren en vervolgens een tweede script waarin dat Bearer Token gebruikt kan worden om de API aanroep (ophalen/bijwerken of verwijderen) te maken aan de bron waarin u geïnteresseerd bent, of u kunt één script schrijven om beide acties te ondernemen terwijl u ervoor zorgt dat als een token aan toonder al gegenereerd is, een voorwaarde in de code wordt gehouden dat er geen nieuw token aan toonder gegenereerd wordt telkens als het script wordt uitgevoerd.

Om het werkend in python te maken, moet u deze bibliotheken installeren:

```
pip install oauthlib pip install requests_oauthlib
```

Script 1:

Vermeld de juiste gegevens client_iden client_secretin dit script:

```
import requests from oauthlib.oauth2 import BackendApplicationClient from oauthlib.oauth2 import TokenE
```

Uitvoer:

De output van dit script moet er zo uit zien:

```
Token: {'token_type': 'bearer', 'access_token': 'eyJhbGciOiJSUzI1NiIsImtpZCI6IjcyNmI5MGUzLWxxxxxxxxxxxxx
```

Het access_tokenis erg lang met duizenden tekens en om de output leesbaar te houden, is het alleen voor dit voorbeeld ingekort.

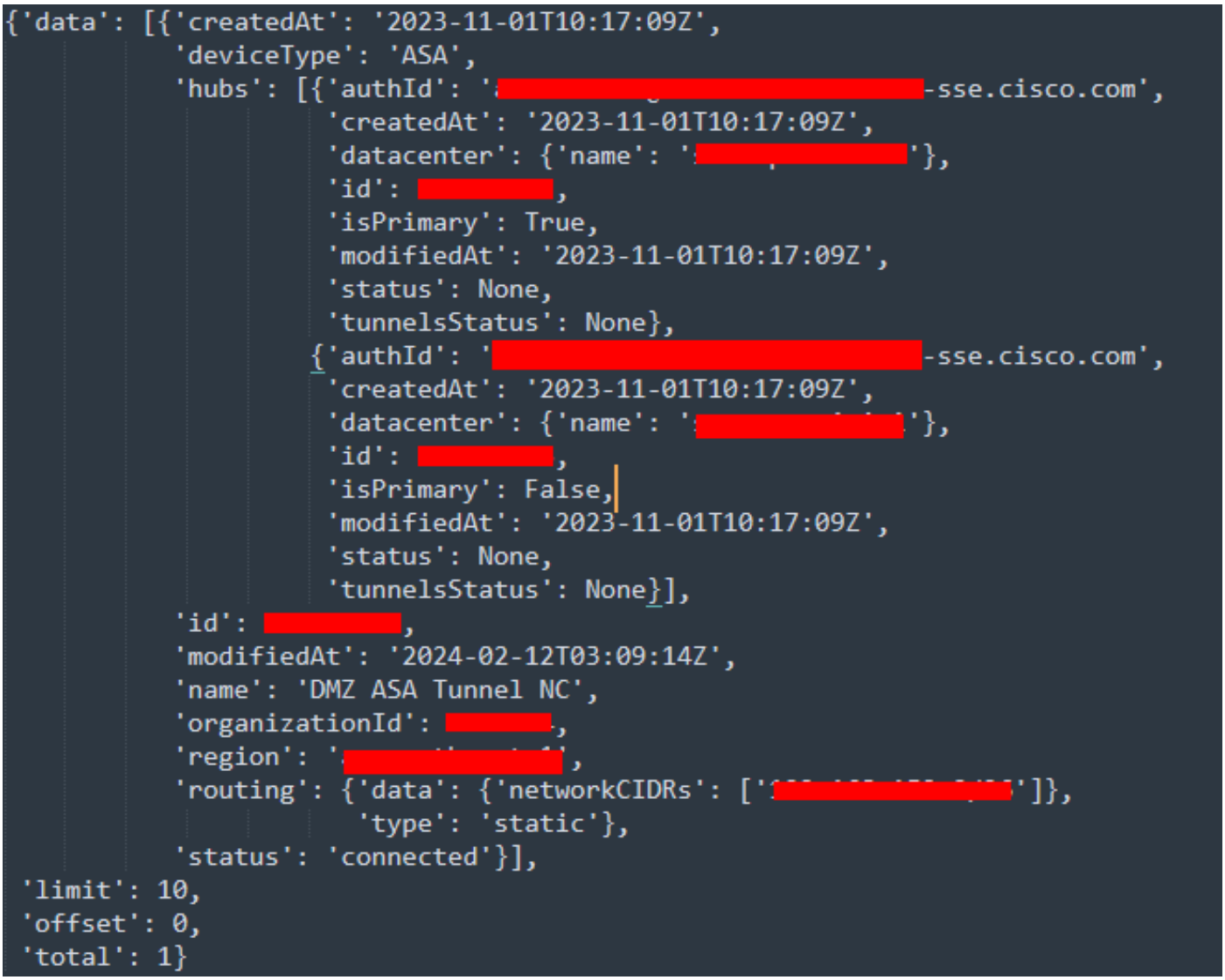
Script 2:

De access_token van Script 1 kan dan in dit script worden gebruikt om API-oproepen te maken. Als voorbeeld, gebruik Script 2 om de informatie over de Groepen van de Tunnel van het Netwerk te halen die het middel gebruiken /deployments/v2/networktunnelgroups:

```
import requests import pprint import json url = "https://api.sse.cisco.com/deployments/v2/networktunnel
```

Uitvoer:

De output van dit script moet er zo uit zien:



Python-uitvoer - Network Tunnel Groups

U kunt ook informatie opvragen over beleid, roamingcomputers, rapporten, enzovoort, met de Secure Access Developers Gebruikershandleiding.

Problemen oplossen

De Secure Access API-endpoints gebruiken HTTP-responscodes om aan te geven of een API-aanvraag is geslaagd of mislukt. Over het algemeen wijzen codes in het 2xx-bereik op succes, codes in het 4xx-bereik op een fout die het gevolg is van de verstrekte informatie, en codes in het 5xx-bereik geven serverfouten aan. De aanpak om het probleem op te lossen, zou afhangen van de responscode die wordt ontvangen:

200	**OK**	Success. Everything worked as expected.
201	**Created**	New resource created.
202	**Accepted**	Success. Action is queued.
204	**No Content**	Success. Response with no message body.
400	**Bad Request**	Likely missing a required parameter or malformed JSON. The syntax of your query may need to be revised. Check for any spaces preceding, trailing, or in the domain name of the domain you are trying to query.
401	**Unauthorized**	The authorization header is missing or the key and secret pair is invalid. Ensure your API token is valid.
403	**Forbidden**	The client is unauthorized to access the content.
404	**Not Found**	The requested resource doesn't exist. Check the syntax of your query or ensure the IP and domain are valid.
409	**Conflict**	The client requests that the server create the resource, but the resource already exists in the collection.
429	**Exceeded Limit**	Too many requests received in a given amount of time. You may have exceeded the rate limits for your organization or package.
413	**Content Too Large**	The request payload is larger than the limits defined by the server.

REST API - Antwoordcodes 1

500	**Internal Server Error**	Something wrong with the server.
503	**Service Unavailable**	Server is unable to complete request.

REST API - Antwoordcodes 2

Gerelateerde informatie

- Gebruikershandleiding voor Cisco Secure Access
- Cisco technische ondersteuning en downloads
- API-toetsen voor beveiligde toegang toevoegen
- Gebruikershandleiding Ontwikkelaars

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.