Configuración de Secure Access para utilizar la API REST con Python

Contenido

Introducción

Este documento describe los pasos para configurar el acceso a la API y utilizarlo para obtener información de recursos de Secure Access.

Prerequisites

Cisco recomienda que tenga conocimiento sobre estos temas:

1. Python 3.x
2. API REST
3. Acceso seguro de Cisco

Requirements

Estos requisitos deben cumplirse antes de continuar:

- Cuenta de usuario de Cisco Secure Access con el rol de administrador completo.
- Cuenta Cisco Security Cloud Single Sign On (SCSO) para iniciar sesión en Secure Access.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

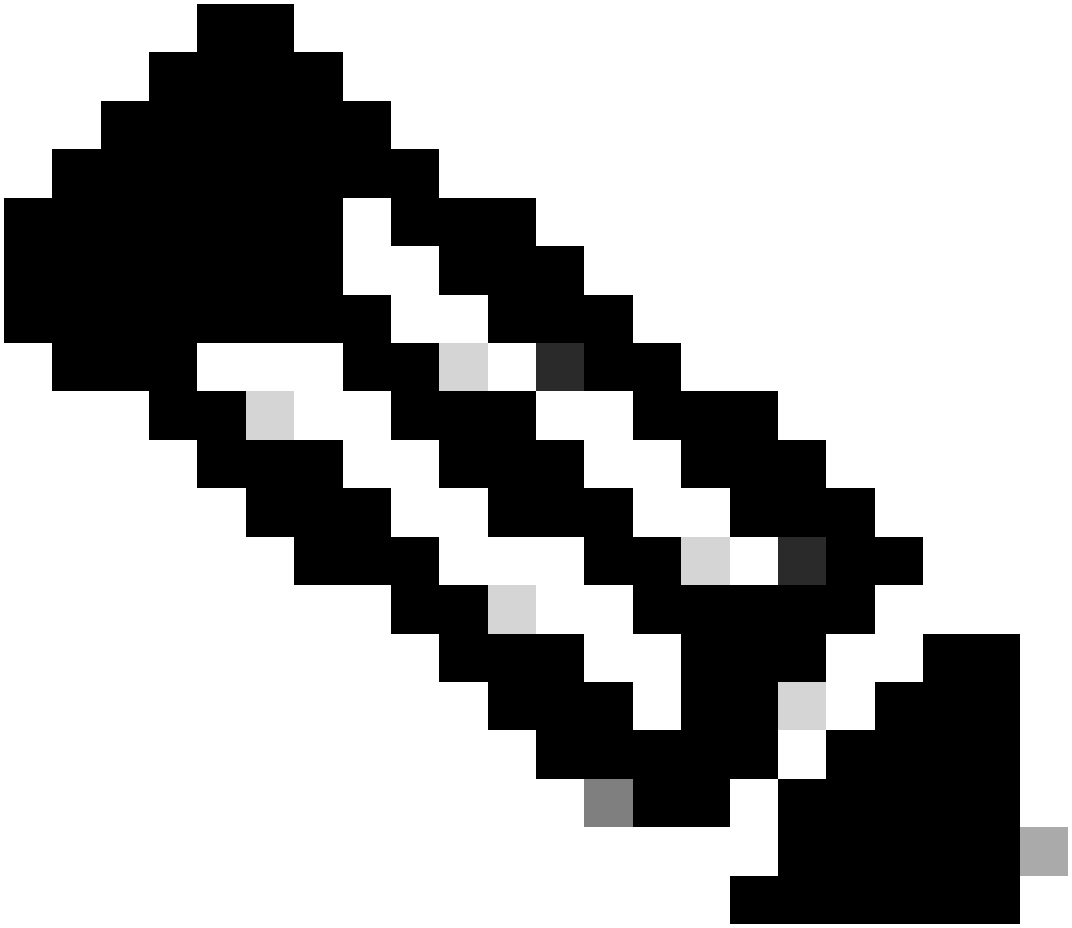
- Panel de acceso seguro

- Python

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Configurar

La API de acceso seguro proporciona una interfaz REST estándar y admite el flujo de credenciales de cliente de OAuth 2.0. Para comenzar, inicie sesión en Secure Access y cree sus claves API de Secure Access. A continuación, utilice las credenciales de la API para generar un token de acceso a la API.



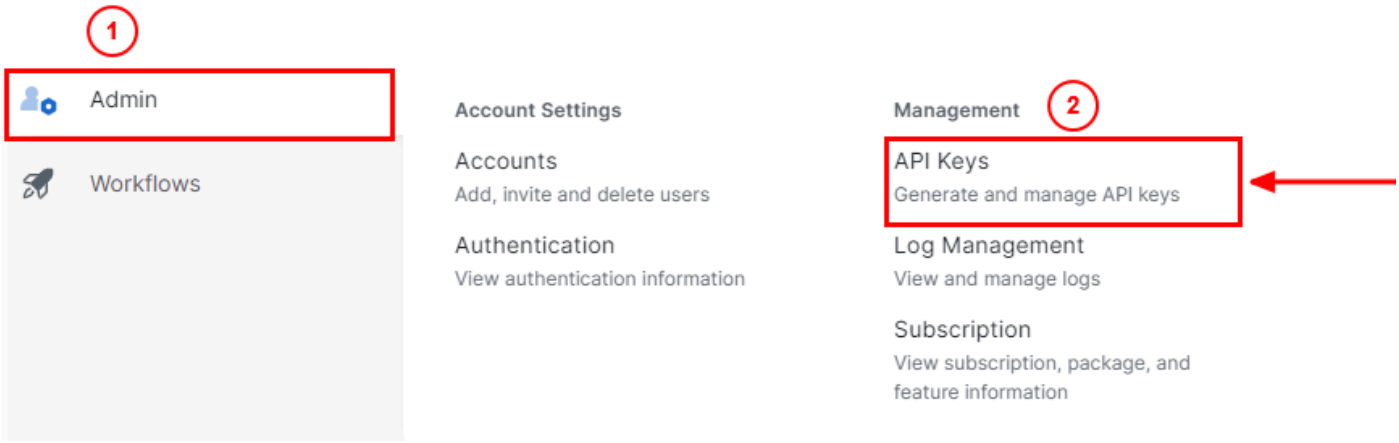
> Nota: Las claves API, las contraseñas, los secretos y los tokens permiten el acceso a sus datos privados. Nunca debe compartir sus credenciales con otro usuario u organización.

Configure la clave de la API desde el panel de acceso seguro antes de ejecutar los scripts mencionados en este artículo.

Crear una clave de API

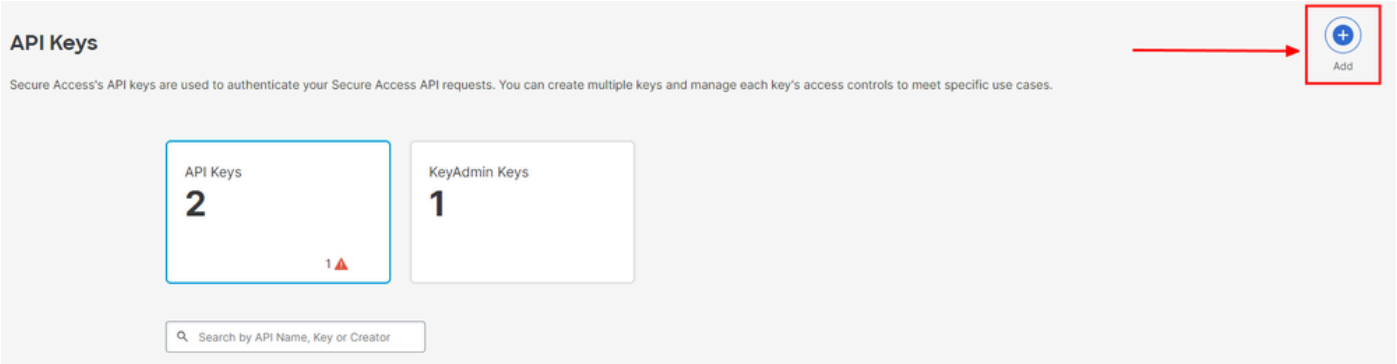
Cree una clave API y un secreto con estos pasos. Inicie sesión en Secure Access con la URL: [Secure Access]

1. En la barra lateral izquierda, seleccione la opción Admin.
    - En Admin, seleccione la opción **API Keys**:



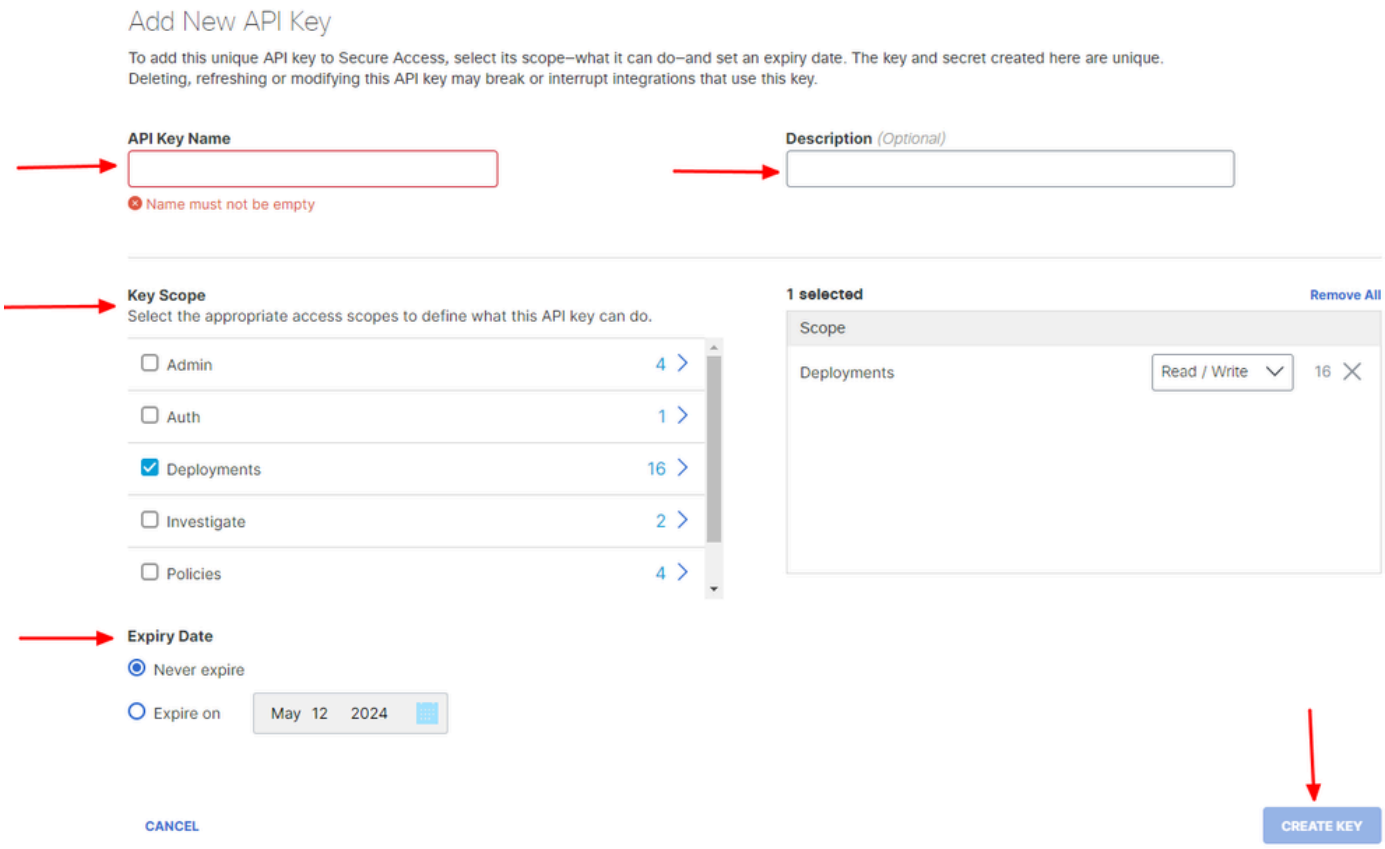
Administrador del panel de acceso seguro: claves de API

3. En la esquina superior derecha, haga clic en el + botón Add a new API Key (Agregar una nueva clave API):



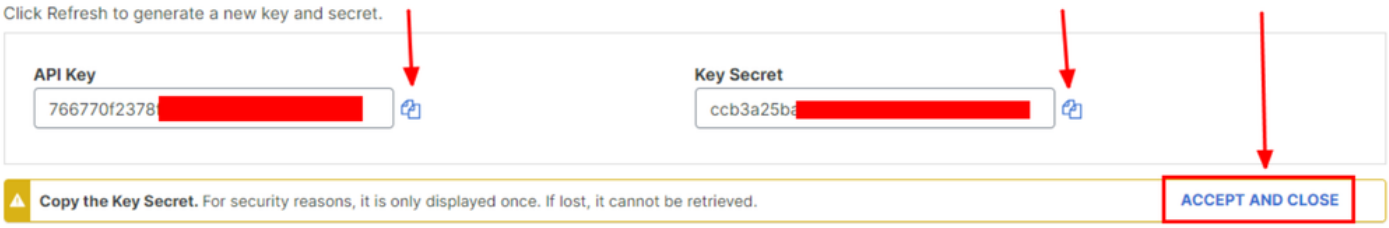
Acceso seguro - Agregar clave de API

4. Proporcione el **API Key Name, Description**(Opcional) y seleccione el Key scope y Expiry date según sus necesidades. Una vez hecho esto, haga clic en el botón **Create**:

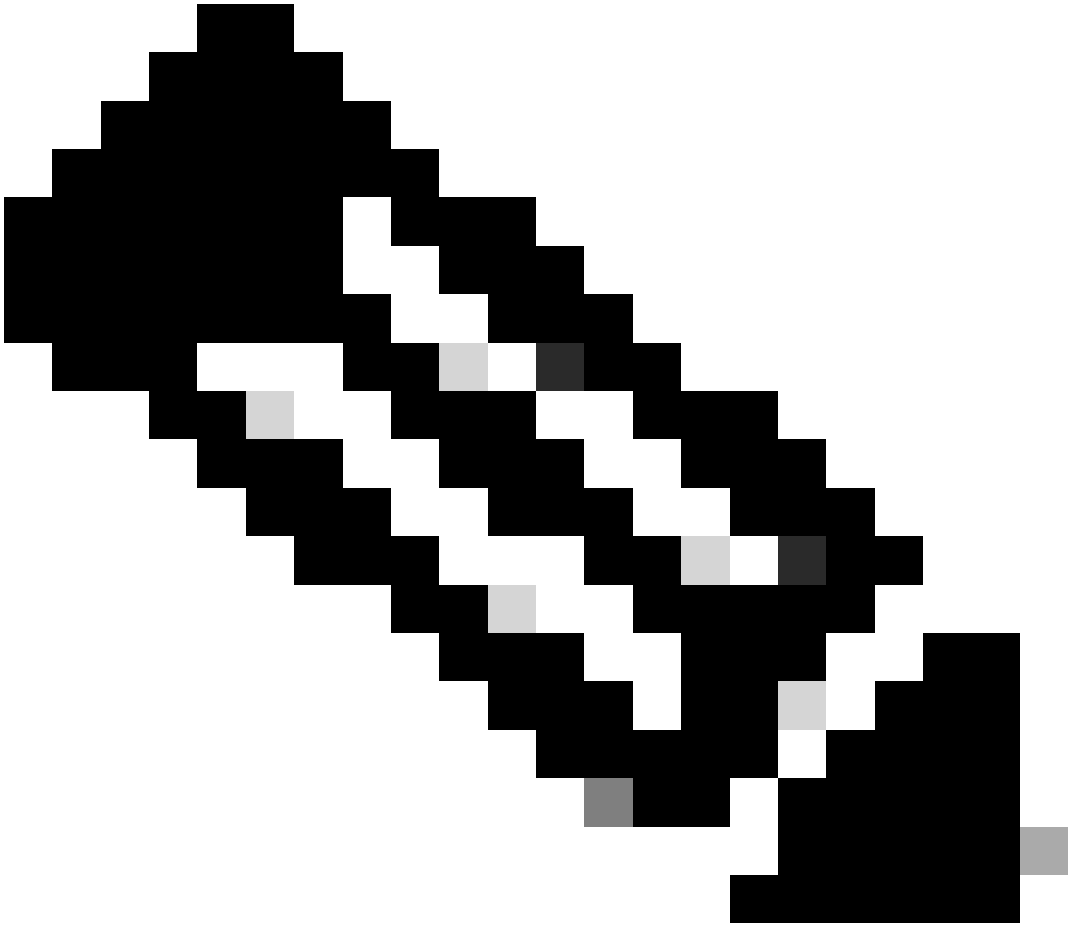


Acceso seguro: detalles clave de la API

5. Copie el API Keyy el **Key Secret** y, a continuación, haga clic en ACCEPT AND CLOSE:



Acceso seguro: clave API y secreto



Nota: solo existe una oportunidad para copiar el secreto de la API. Secure Access no guarda el secreto de la API y no puede recuperarlo después de su creación inicial.

Código Python

Hay varias formas de escribir este código, teniendo en cuenta que el token generado es válido durante 3600 segundos (1 hora). Puede crear 2 scripts separados en los que el primer script se puede utilizar para generar el token portador y luego un segundo script en el que dicho token portador se puede utilizar para realizar la llamada API (obtener/actualizar o eliminar) al recurso que le interese, o escribir un solo script para realizar ambas acciones mientras se asegura de que si ya se ha generado un token portador, se mantenga una condición en el código de que no se genere un nuevo token portador cada vez que se ejecute el script.

Para que funcione en Python, asegúrese de instalar estas bibliotecas:

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

Script 1:

Asegúrese de mencionar el client_idy client_secreten este script correcto:

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

Salida:

El resultado de este script debe ser similar a lo siguiente:

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

El access_tokenes muy largo con miles de caracteres y, por lo tanto, para mantener la salida legible, se ha acortado solo para este ejemplo.

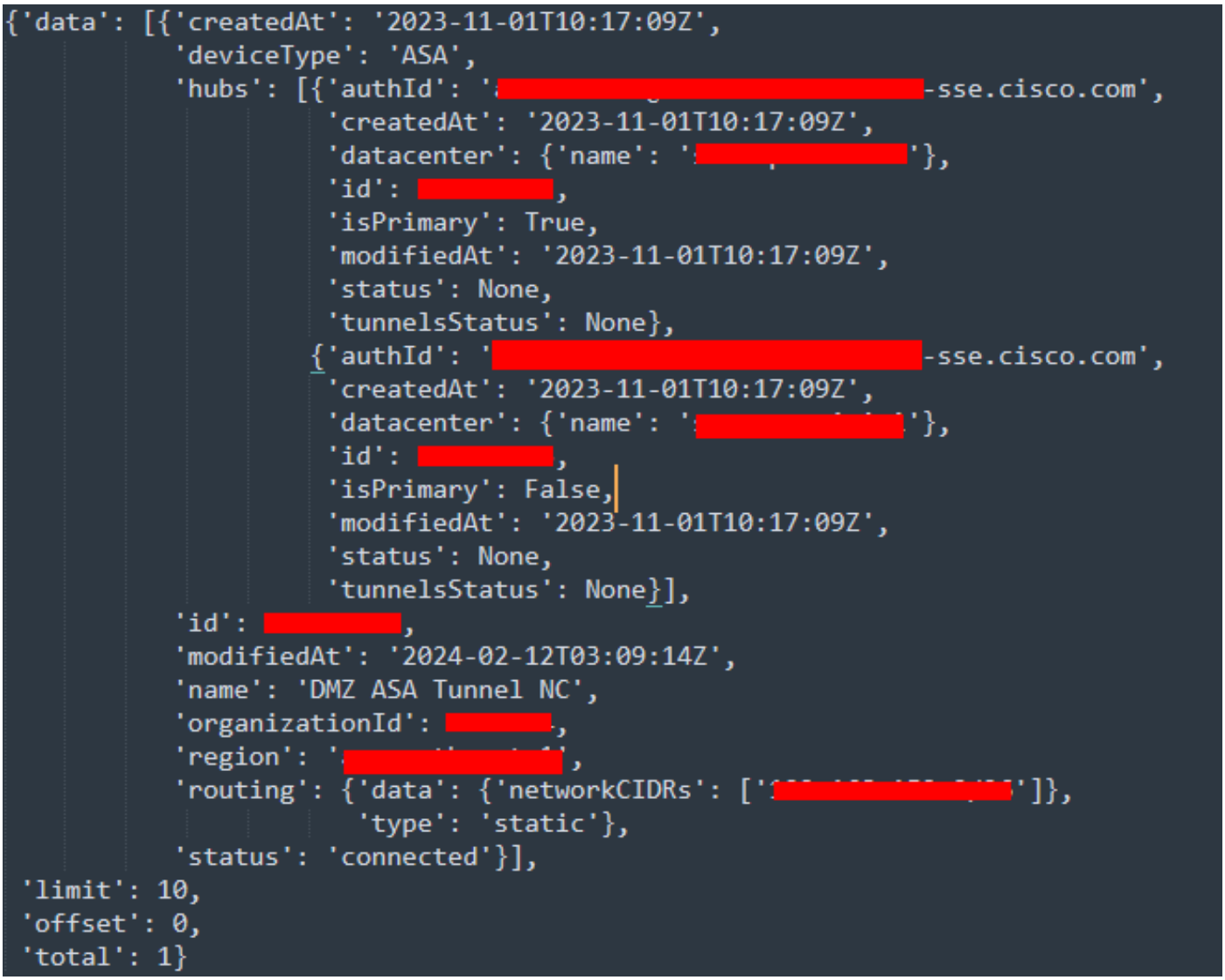
Script 2:

El access_token de Script 1 se puede utilizar en este script para realizar llamadas API. A modo de ejemplo, utilice la secuencia de comandos 2 para obtener la información sobre los grupos de túnel de red que utilizan el recurso /deployments/v2/networktunnelgroups:

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

Salida:

El resultado de este script debe ser similar a lo siguiente:



Salida de Python: grupos de túnel de red

También puede obtener información sobre directivas, equipos en roaming, informes, etc., con la Guía del usuario para desarrolladores de Secure Access.

Troubleshoot

Los terminales de la API de acceso seguro utilizan códigos de respuesta HTTP para indicar el éxito o el fracaso de una solicitud de la API. En general, los códigos de la gama 2xx indican éxito, los códigos de la gama 4xx indican un error derivado de la información proporcionada y los códigos de la gama 5xx indican errores del servidor. El enfoque para resolver el problema dependería del código de respuesta que se reciba:

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.

API REST - Códigos de respuesta 1

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

API REST - Códigos de respuesta 2

Información Relacionada

- Guía del usuario de Cisco Secure Access
- Soporte técnico y descargas de Cisco
- Agregar claves API de acceso seguro
- Guía del usuario para desarrolladores

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).