Configurazione di Secure Access per l'utilizzo dell'API REST con Python

Sommario

Introduzione

In questo documento viene descritto come configurare l'accesso API e utilizzarlo per recuperare le informazioni sulle risorse da Secure Access.

Prerequisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

1. Python 3.x
2. API REST
3. Cisco Secure Access

Requisiti

Prima di procedere, è necessario soddisfare i seguenti requisiti:

- Account utente Cisco Secure Access con il ruolo di amministratore completo.
- Account Cisco Security Cloud Single Sign On (SCSO) per accedere a Secure Access.

Componenti usati

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

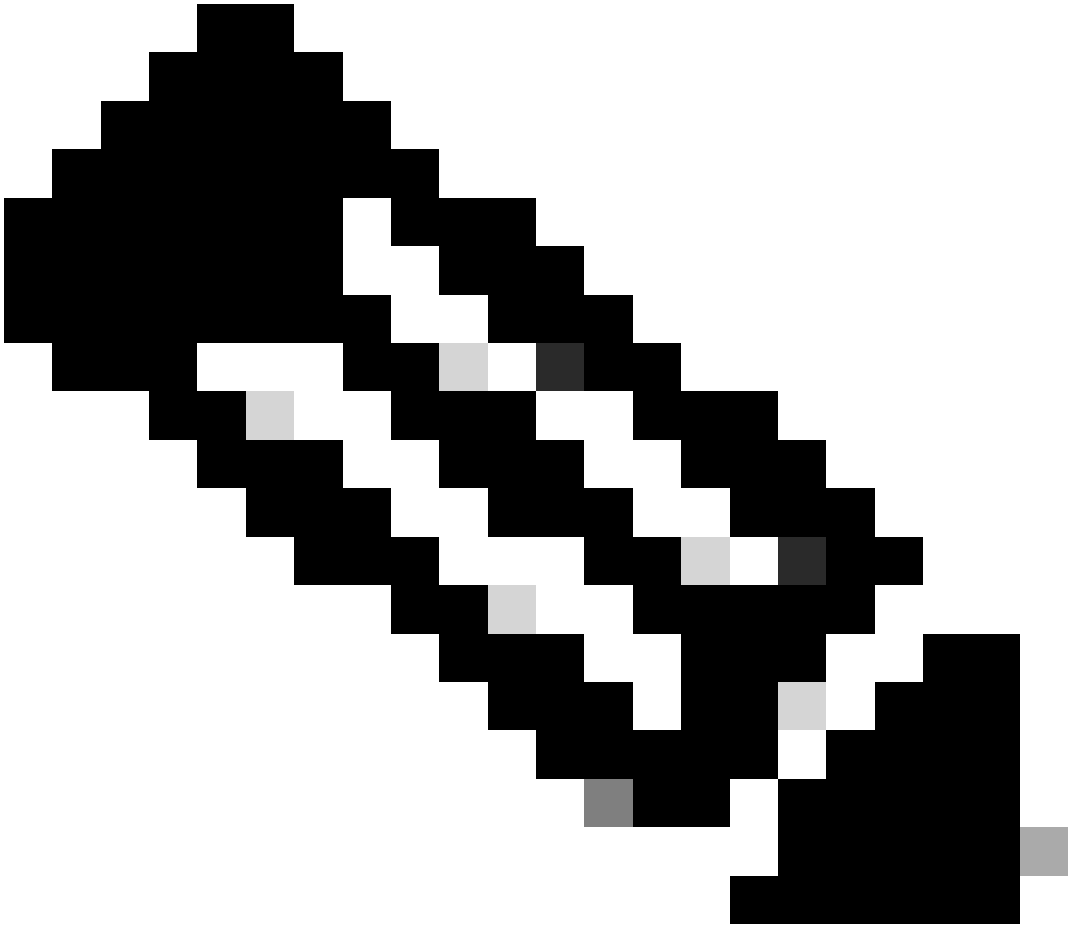
- Dashboard di accesso protetto

- Python

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.

Configurazione

L'API Secure Access fornisce un'interfaccia REST standard e supporta il flusso di credenziali del client OAuth 2.0. Per iniziare, accedere a Secure Access e creare le chiavi dell'API Secure Access. Quindi, usare le credenziali API per generare un token di accesso API.



> Nota: le chiavi API, le password, i segreti e i token consentono di accedere ai dati privati. Le credenziali non devono mai essere condivise con altri utenti o organizzazioni.

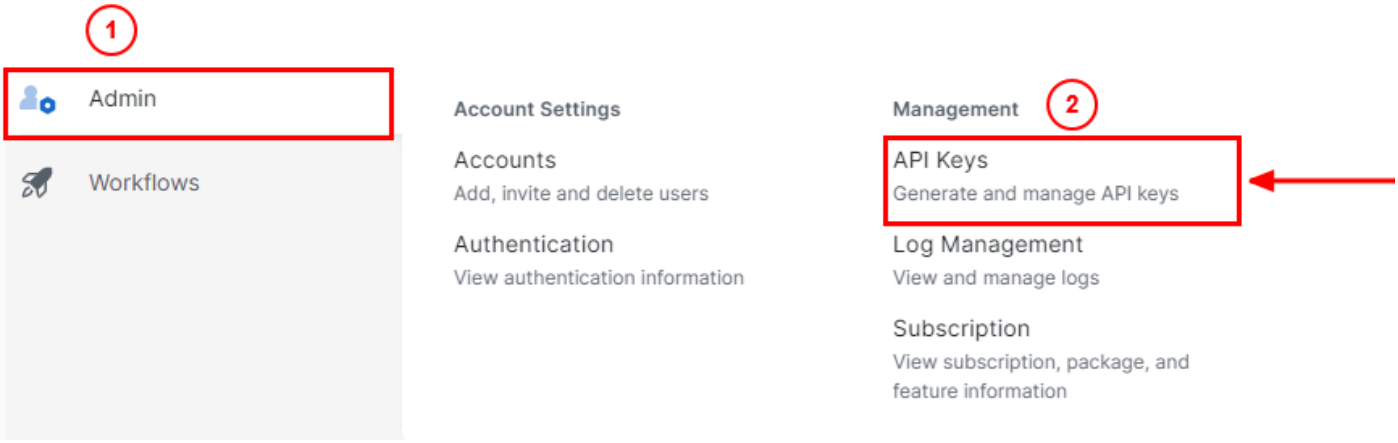
Configurare la chiave API dal dashboard di accesso sicuro prima di eseguire gli script menzionati

in questo articolo.

Crea una chiave API

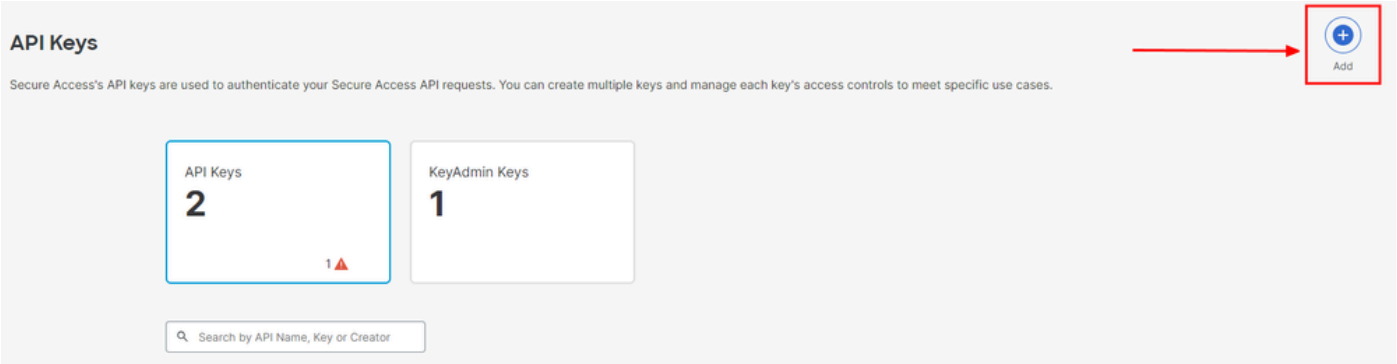
Creare una chiave API e un segreto attenendosi alla seguente procedura. Accedere a Secure Access con l'URL: Secure Access

1. Dalla barra laterale sinistra, selezionare l'opzione Admin.
    - In Admin selezionare l'opzione **API Keys**:



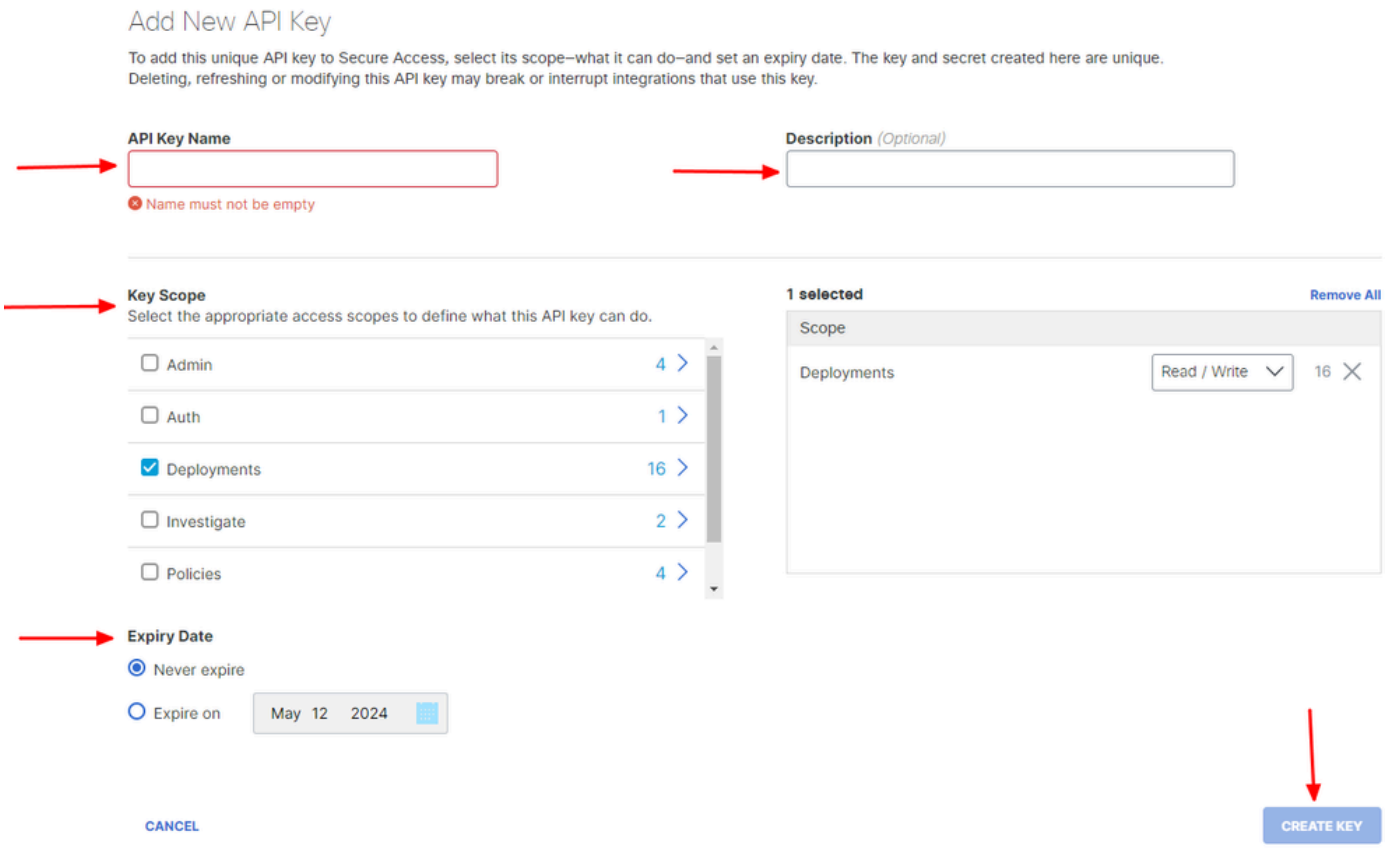
Amministratore dashboard di accesso sicuro - Chiavi API

3. Nell'angolo in alto a destra, fai clic sul + pulsante per aggiungere una nuova chiave API:



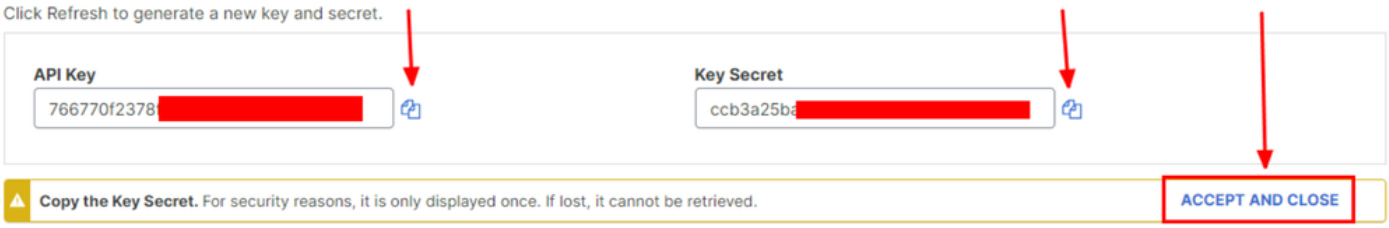
Accesso sicuro - Aggiungi chiave API

4. Fornire il **API Key Name, Description**(Facoltativo) e selezionare il Key scope e Expiry date in base alle proprie esigenze. Al termine, fare clic sul pulsante **Create**:

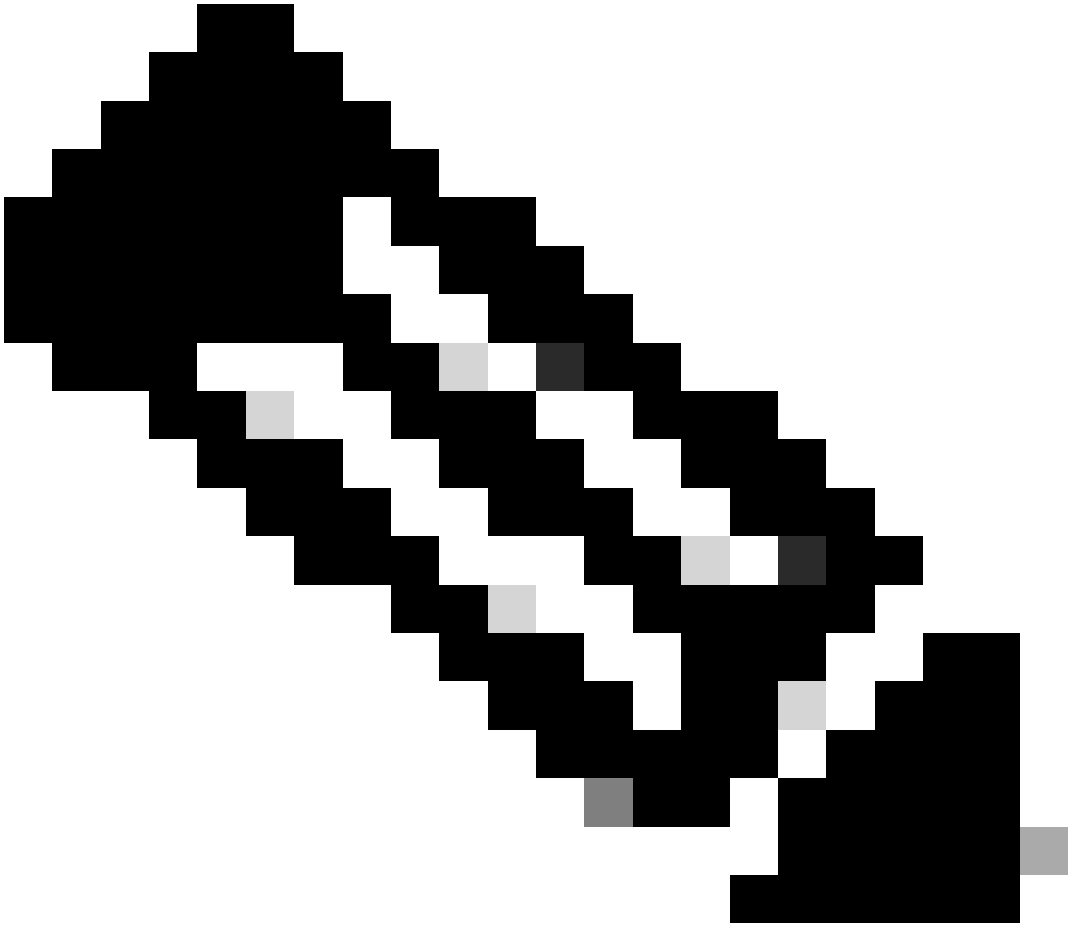


Accesso sicuro - Dettagli chiave API

5. Copiare il API Keye il **Key Secret** e poi fare clic su ACCEPT AND CLOSE:



Accesso sicuro - Chiave e segreto API



Nota: esiste solo un'opportunità per copiare il segreto API. Secure Access non salva il segreto API e non è possibile recuperarlo dopo la creazione iniziale.

Codice Python

Esistono diversi modi per scrivere questo codice, considerando che il token generato è valido per 3600 secondi (1 ora). È possibile creare due script distinti in cui il primo script può essere utilizzato per generare il token Bearer e quindi un secondo script in cui tale token Bearer può essere utilizzato per eseguire la chiamata API (recupero/aggiornamento o eliminazione) alla risorsa a cui si è interessati oppure scrivere un singolo script per eseguire entrambe le azioni assicurandosi che, se un token Bearer è già stato generato, nel codice venga mantenuta una condizione che un nuovo token Bearer non venga generato ogni volta che lo script viene eseguito.

Per fare in modo che funzioni in python, assicurarsi di installare queste librerie:

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

Script 1:

Assicurarsi di menzionare il corretto client_ide client_secretin questo script:

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

Uscita:

L'output di questo script deve essere simile al seguente:

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

Il access_tokenè molto lungo con migliaia di caratteri e, quindi, per mantenere l'output leggibile, è stato abbreviato solo per questo esempio.

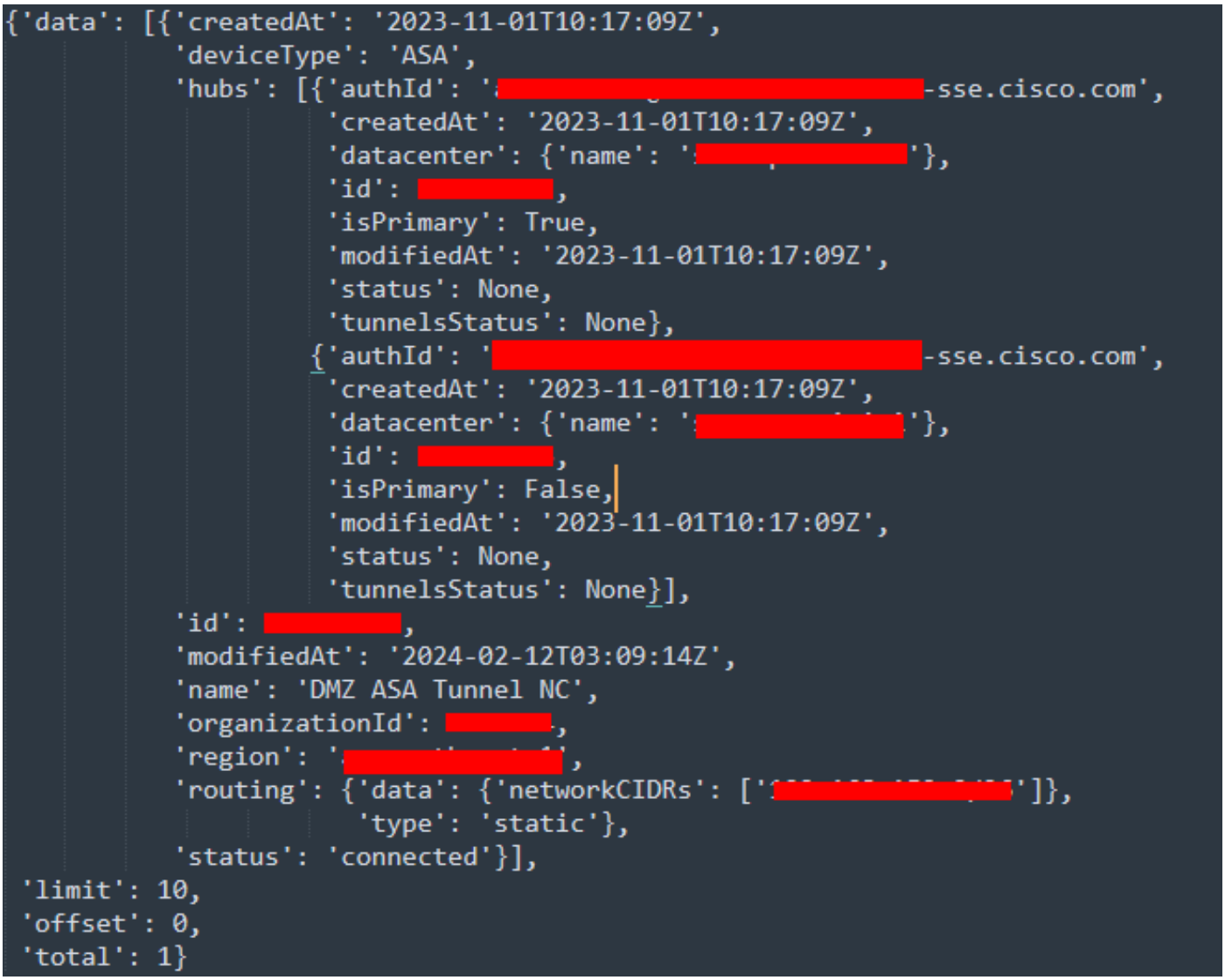
Script 2:

Il comando access_token from Script 1 può quindi essere utilizzato in questo script per effettuare chiamate API. Ad esempio, utilizzare Script 2 per recuperare le informazioni sui gruppi di tunnel di rete tramite la risorsa /deployments/v2/networktunnelgroups:

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

Uscita:

L'output di questo script deve essere simile al seguente:



Output Python - Network Tunnel Group

È inoltre possibile recuperare informazioni su criteri, computer mobili, report e così via, tramite la Guida per l'utente degli sviluppatori Secure Access.

Risoluzione dei problemi

Gli endpoint API di accesso sicuro utilizzano i codici di risposta HTTP per indicare l'esito positivo o negativo di una richiesta API. In generale, i codici nell'intervallo 2xx indicano un esito positivo, i codici nell'intervallo 4xx indicano un errore risultante dalle informazioni fornite e i codici nell'intervallo 5xx indicano errori del server. L'approccio per risolvere il problema dipende dal codice di risposta ricevuto:

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 - Codici di risposta 1

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

API REST - Codici di risposta 2

Informazioni correlate

- Guida per l'utente di Cisco Secure Access
- Supporto tecnico Cisco e download
- Aggiungi chiavi API di accesso sicuro
- Guida per l'utente per gli sviluppatori

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).