

# Verwalten sicherer Zugriffs-Ziellisten mithilfe von Python und REST-API

## Inhalt

---

[Einleitung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konfigurieren](#)

[Skript](#)

[Fehler](#)

[Fehlerbehebung](#)

[Zugehörige Informationen](#)

---

[Spoiler](#) (Zum Lesen markieren)

Beachten Sie, dass Cisco keinen offiziellen Support für dieses Entwicklungsdesign anbietet. Es dient lediglich als Referenzbeispiel, um die Kommunikation zwischen API und Anwendungen zu vereinfachen. Die Benutzer dürfen dieses Design nur zu Schulungszwecken und nicht als Grundlage für die Implementierung auf Produktionsebene verwenden. Die Durchführung des in diesem Artikel vorgestellten Codes erfolgt auf eigenes Risiko, und Cisco übernimmt keine Haftung für Probleme, die sich aus der Verwendung des Codes ergeben.

Beachten Sie, dass Cisco keinen offiziellen Support für dieses Entwicklungsdesign anbietet. Es dient lediglich als Referenzbeispiel, um die Kommunikation zwischen API und Anwendungen zu vereinfachen. Die Benutzer dürfen dieses Design nur zu Schulungszwecken und nicht als Grundlage für die Implementierung auf Produktionsebene verwenden. Die Durchführung des in diesem Artikel vorgestellten Codes erfolgt auf eigenes Risiko, und Cisco übernimmt keine Haftung für Probleme, die sich aus der Verwendung des Codes ergeben.

## Einleitung

In diesem Dokument wird beschrieben, wie Sie alle möglichen Operationen für Ziellisten mithilfe der Python- und REST-API durchführen.

## Voraussetzungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

1. Python
2. REST-API
3. Sicherer Zugriff von Cisco

## Anforderungen

Diese Anforderungen müssen erfüllt sein, bevor weitere Schritte unternommen werden können:

- Cisco Secure Access-Benutzerkonto mit der Rolle "Vollständiger Administrator"
- Cisco Security Cloud Single Sign On (SCSO)-Konto für die Anmeldung bei Secure Access.
- [Erstellen von API-Schlüsseln für sicheren Zugriff](#)

## Verwendete Komponenten

Die Informationen in diesem Dokument basierend auf folgenden Software- und Hardware-Versionen:

- Dashboard für sicheren Zugriff
- Python 3.x

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

## Konfigurieren

Es gibt mehrere Möglichkeiten, diesen Code unter Berücksichtigung verschiedener Aspekte wie Fehlerbehandlung, Tokengültigkeit (3600 Sekunden) usw. zu schreiben.

Bitte stellen Sie sicher, dass diese Python-Bibliotheken installiert sind, bevor Sie das Skript ausführen:

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

Skript

Ersetzen Sie die `client_id` und die `client_secret` durch die API Key und Key Secret in diesem Skript.

```
from oauthlib.oauth2 import BackendApplicationClient from oauthlib.oauth2 import TokenExpiredError from
```

Ausgabe:

Die Ausgabe dieses Skripts muss in etwa wie folgt aussehen:

```
Token Already Generated? (Y/N) :: y Available operations: 1. Get Destination Lists 2. Get Destination L
```

Nachdem dieses Programm erfolgreich ausgeführt wurde, wird zu Beginn eine Frage zum Cookie gestellt - Cookie Already Generated? (Y/N). Der Grund, diese Frage zu stellen, ist sicherzustellen, dass Sie das Cookie nicht mehrmals generieren, da ein einmal generiertes Cookie für 3600 Sekunden (1 Stunde) gültig ist. Wenn Sie oder eingeben yY, wird kein neues Cookie generiert. Wenn Sie jedoch noder N eingeben, wird ein neues Cookie generiert und in einer lokalen Textdatei im gleichen Verzeichnis/Ordner gespeichert. Das Cookie aus dieser Datei wird in Ihren nachfolgenden Anfragen verwendet.

## Fehler

Dieser Fehler tritt auf, wenn Sie für einen Vorgang, für den Sie die DestinationList-ID angeben müssen, eine falsche ID eingeben:

```
{'message': 'no Route matched with those values'}
```

Wenn Sie beim Erstellen einer DestinationList den Namen von DestinationList erwähnen, der mehr als 255 Zeichen umfasst, wird folgender Fehler angezeigt:

```
{'code': 400, 'code_text': 'Bad Request', 'error': 'invalid_request', 'message': {'name': {'code': 'str
```

Weitere Informationen zu Richtlinien, Roaming-Computern, Berichten usw. finden Sie im [Secure Access Developers User Guide](#).

## Fehlerbehebung

Die Endpunkte der API für sicheren Zugriff verwenden HTTP-Antwortcodes, um den Erfolg oder Misserfolg einer API-Anforderung anzuzeigen. Im Allgemeinen weisen Codes im 2xx-Bereich auf Erfolg hin, Codes im 4xx-Bereich auf einen Fehler hin, der sich aus den bereitgestellten Informationen ergibt, und Codes im 5xx-Bereich auf Serverfehler. Der Ansatz zur Behebung des Problems hängt vom empfangenen Antwortcode ab:

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

#### REST API - Antwortcodes 1

500	<b>Internal Server Error</b>	Something wrong with the server.
503	<b>Service Unavailable</b>	Server is unable to complete request.

#### REST API - Antwortcodes 2

#### Zugehörige Informationen

- [Cisco Secure Access Benutzerhandbuch](#)
- [Technischer Support und Downloads von Cisco](#)
- [Hinzufügen von API-Schlüsseln für sicheren Zugriff](#)

- [Entwickler-Benutzerhandbuch](#)
- [Konfigurieren von sicherem Zugriff zur Verwendung der REST-API mit Python](#)
- [Ziellisten über cURL verwalten](#)

## Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.