|  |  |
| --- | --- |
| Python Programming Language Is Considered Better Than Other Languages -  DataWider  Programming Networking Essiantials  Howest Brugge Graduaat | michiel wylleman  student |

Inhoud

[Inleiding 2](#_Toc59551945)

[Verslag 3](#_Toc59551946)

[Functies 3](#_Toc59551947)

[Ticket aanvragen 3](#_Toc59551948)

[Input accestoken 5](#_Toc59551949)

[Lijst van rooms tonen 6](#_Toc59551950)

[Room kiezen 7](#_Toc59551951)

[Recentste bericht tonen van de room 8](#_Toc59551952)

[Command aan de bot geven(print\_devices\_uptime()) 9](#_Toc59551953)

[Tabel aanmaken 10](#_Toc59551954)

[Tabel versturen naar de gekozen room van de gebruiker 11](#_Toc59551955)

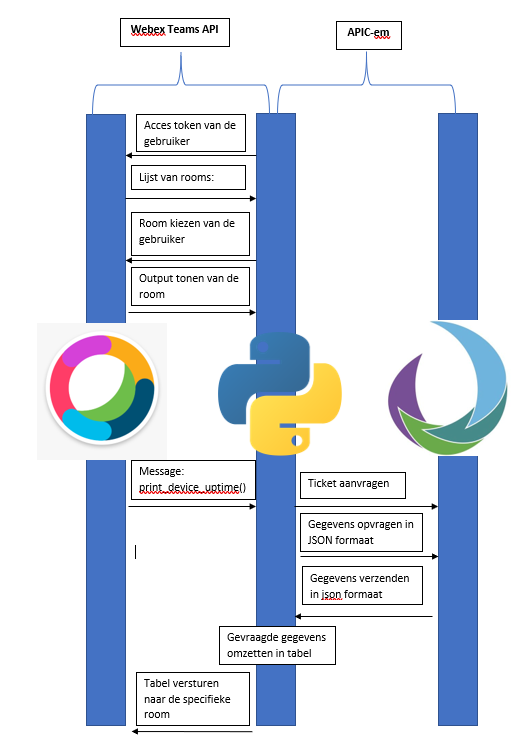
[Command commands() 13](#_Toc59551956)

[Command quit() 14](#_Toc59551957)

[Bronnen: 15](#_Toc59551958)

# Inleiding

*Ik heb een bot gemaakt die een tabel kan maken met waar de up‑time getoont van de virtuele apparaten op APIC‑EM . Dit heb ik allemaal gedaan door een script te schrijven in de scripttaal python waar je een commando kan maken op een room voor Webex Teams.*



# Verslag

## Functies

Ik heb een aantal modules opgevraagd. **Modules** bevatten definities en statements op python die je zult implementeren. Ik zal ze een voor een opsommen.



**time**: handelt tijd gerelateerde taken.   
**JSON**: een data interchange formaat die we gebruiken om gegevens van verschillende **API’s** uit te wisselen in dit formaat en toe te spreken.  
**Requests**: is een standaard voor **HTTP requests**. Zo kunnen verschillende **API’s** aanspreken via **URL**.

Met **Tabulate** kun je in Python tabellen maken.





Deze lijn code zorgt ervoor dat waarschuwingen van **HTTPS requests** uitgeschakeld wordt.  
Er zijn verschillende waarschuwingen op basis ban het niveau van de certificaat verificatie.   
De waarschuwing geven bijzondere situatie aan die op een andere manier opgelost kunnen worden. Zoals bijvoorbeeld de **InsecureRequestWarning**.

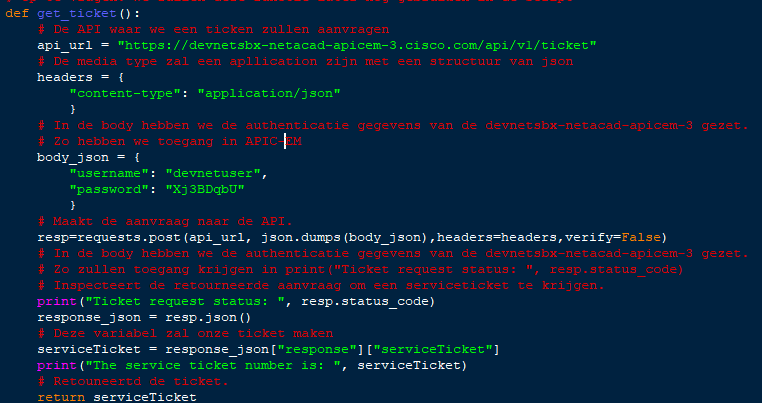
Een **InsecureRequestWarning** waarschuwing komt het meest voor en dit gebeurd enkel wanneer een **request** wordt aangemaakt met een **HTTPS** **URL** zonder certificatie verificatie. In praktijk is dit uiterst belangrijk om te doen omdat het een extra beveiliging is voor scripts maar voor nu nemen we de risico omdat we niks verkeerds doen met de gegevens en enkel gebruiken om ervan leren.



## Ticket aanvragen

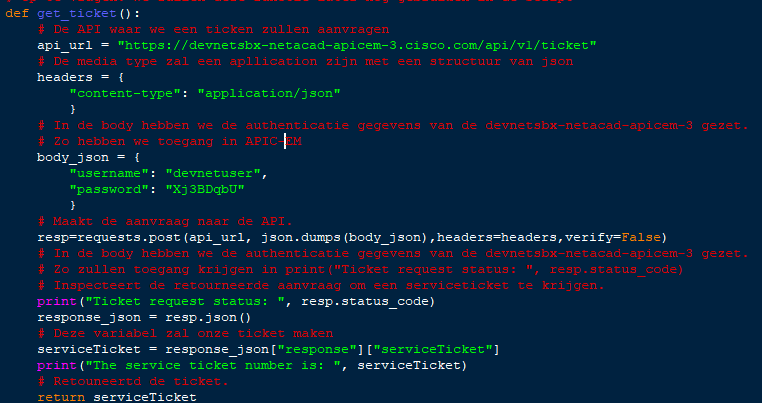
**Application Policy Infrastructure Controller Enterprise Module(APIC‑EM)** is een **SDN controller** die virtuele apparaten beheert en die **API’s** kan creëren. Hier gaan we een aanvraag doen virtuele apparaten van de gebruiker op **APIC‑EM** maar voor dat we dit doen moeten we eerst een **service ticket hebben.**

Bijna elke **request** op **APIC‑EM** moet een geldige service ticket nummer hebben die gebruikt wordt als een authenticatie token.  
Dit gedeelte wordt in functie gestoken om later hem te herroepen. Bij mij heet deze functie **get\_ticket()** omdat dit een duidelijk naam is.

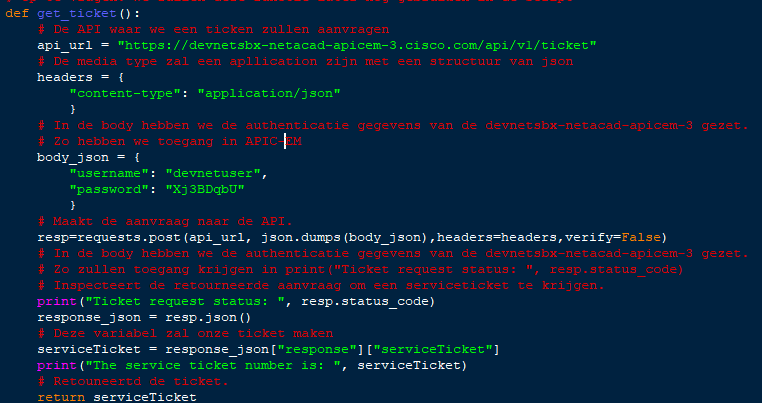


Wat deze functie nodig heeft zijn een aantal variabelen zoals **api\_url**, **headers**, **body\_json**, **resp**, **response\_json** en **serviceTicket**.

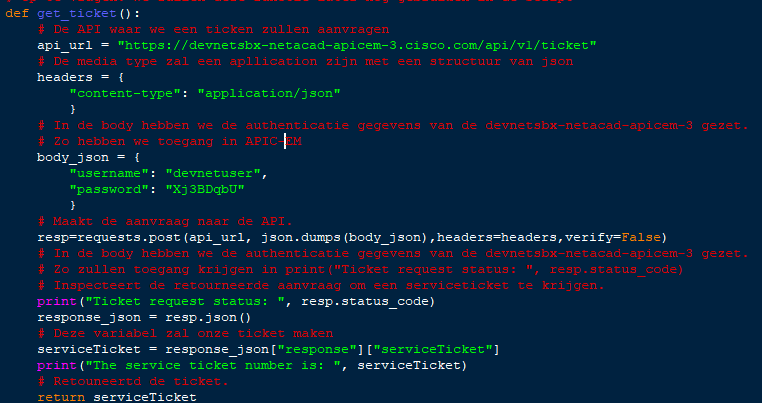
**Api\_url**: In deze variabel geven we de **URL** in van de **API** die we willen gebruiken.



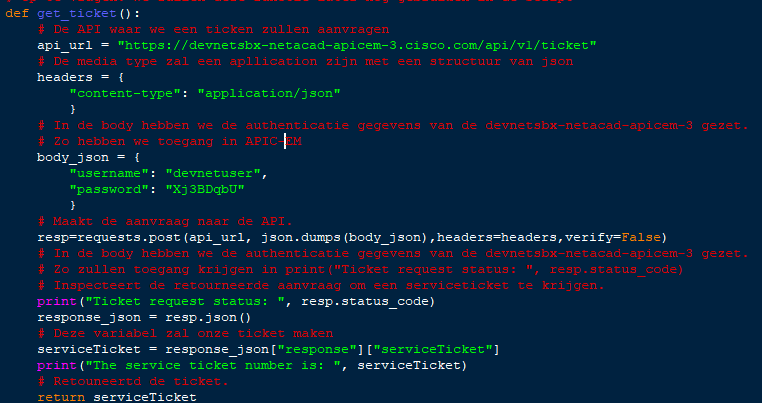
**headers**: De **content‑type header** vertelt aan de client wat dat de type is van de content.



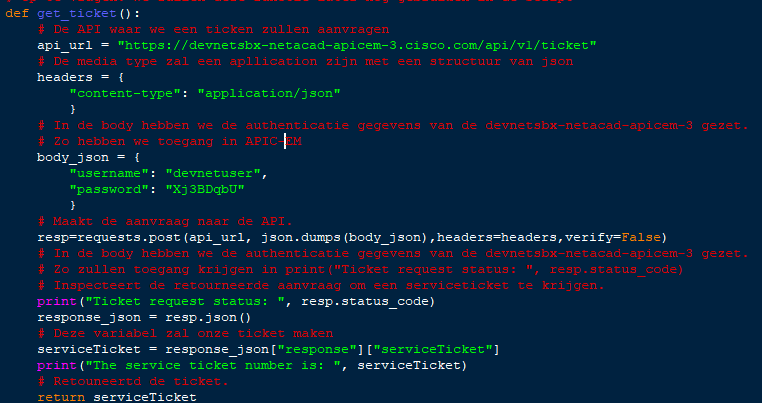
**Body\_json**: In de body steken we gebruikersnaam en wachtwoord in van de gebruiker op **APIC‑EM** om zo te toegang te krijgen.



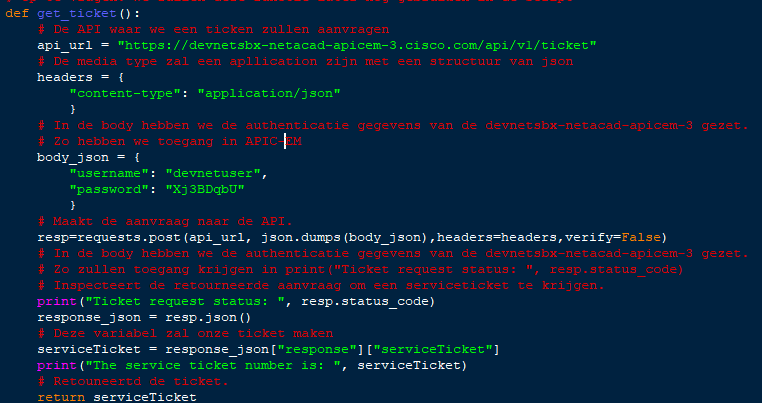
**Resp**: In deze variabel zitten de opgesomde variabels in. Hij spreekt de **API** aan en zal dan gegevens in hem opslaan.



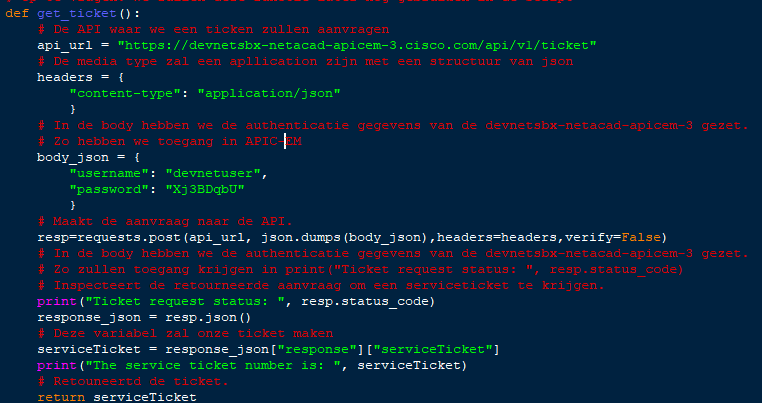
**Response\_json**: JSON is een methode die het variabel resp converteert in **JSON** formaat. De gegevens in **JSON** kunnen aanspreken via hun levels maar dit zullen we later zien.



**ServiceTicket**: Omdat de gegevens die geconverteerd zijn in **JSON** formaat gaan we de waarde opvragen aan hun level0. Response is de eerste level van de serviceticket bevindt zich onder de eerste level.[[1]](#footnote-1)

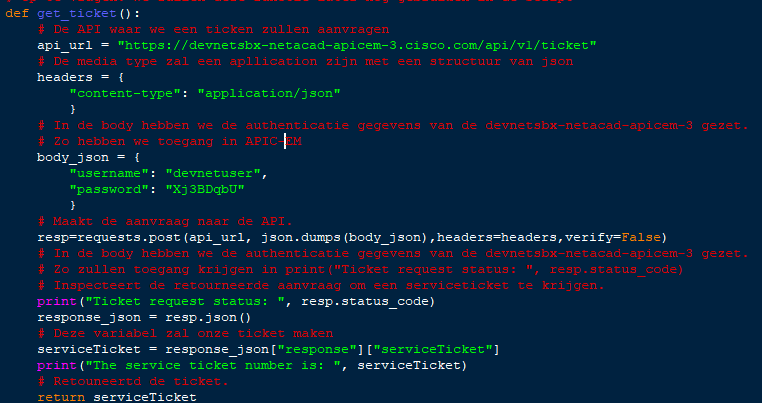


Je kan de **HTTP status code** opvragen via de print functie om zo te weten welke code we krijgen.



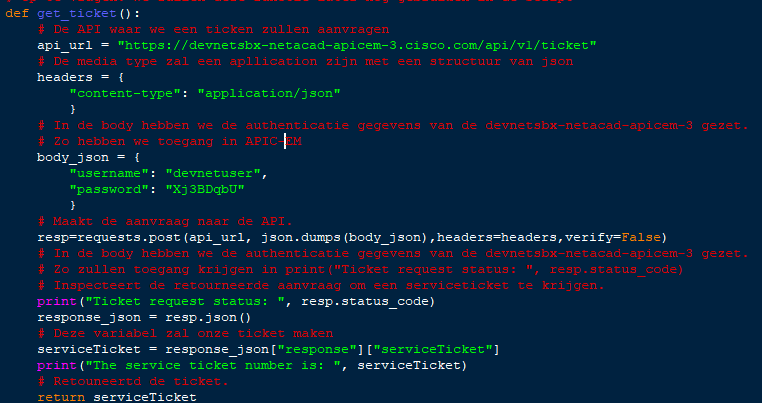


Om te weten wat dat serviceticket nummer is printen we de variabel **serviceTicket** af.



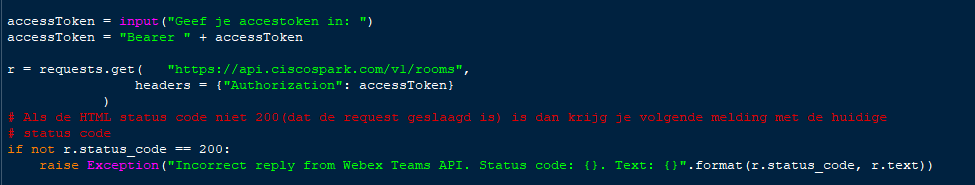


Later zal ik deze ticket nodig hebben dus retourneren we deze variabel. Dit is het eind van deze functie;



## Input accestoken

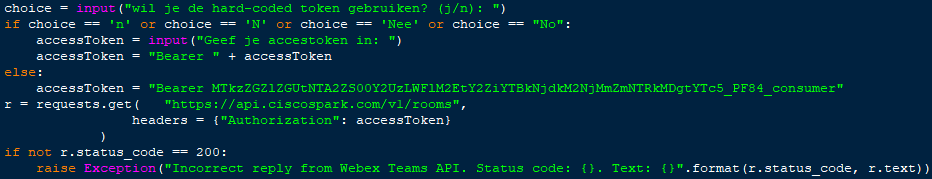
Om toegang te hebben in de **webex teams API** heb ik een **Bearer token** nodig.  
Een **bearer token** is type **access token** die **OAuth 2.0** (een industriële standaard protocol voor authenticatie) gebruikt.



Omdat **Bearer token** iets persoonlijk is voor 12 uren geldig is zal ik dit anders moeten aanpakken.

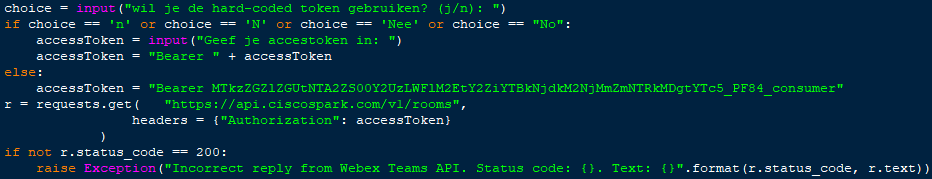


Er zal dan een **request** aangemaakt moeten worden aan de **API** van webex teams om zo de gegevens van de rooms te bereiken. Deze **request** is een **get methode** die gegevens opvraagt van de **API**. Zoals bijvoorbeeld de status code, of gegevens zoals de roomid.



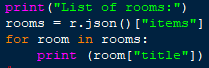
De **variabel** “**r**“ is iets anders omdat we **URL** en **headers** bij elkaar steken maar het is ongeveer dezelfde methode als de **variabel** resp in de **functie** get\_ticket().

Deze if-statement gebruikt een **excecption** als **de status code** niet 200 is. De script zal dan ook stoppen met runnen. Dit stuk code gebruik later terug op andere momenten.

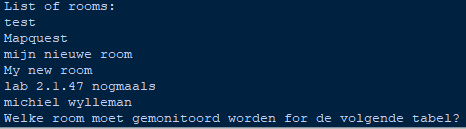


## Lijst van rooms tonen

Om een lijst van rooms te krijgen moeten we werken met een **for loop** maar voor dat we dit gebeurd moet je de **level 0 key** gebruiken van de **API(** level 0 key [“**items**”]).

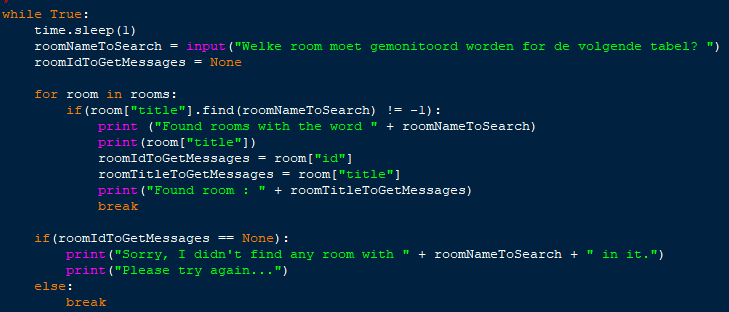


De script zal dan een lijst van rooms tonen

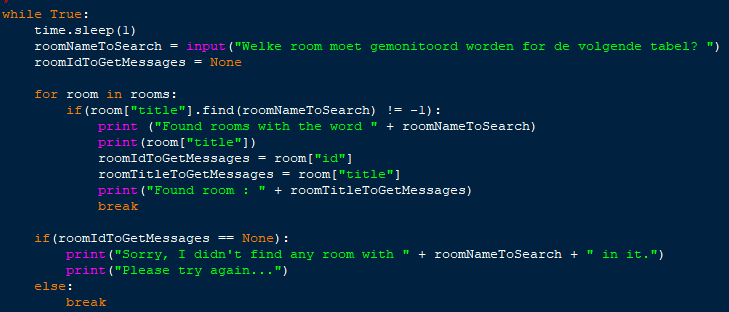


## Room kiezen

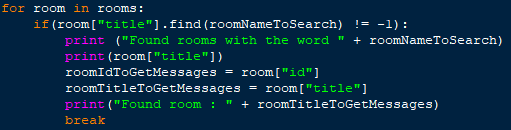
Nu dat de gebruiker een lijst heeft van zijn rooms moet hij één uitkiezen. Vanaf hier werken we constant in een **while loop** om bestaande room te kiezen.



We vertragen het proces met een seconde via **time sleep(1)** om niet de limiet van de **API calls** overschreiden.  
Ik geef ook de gebruiker de input welke room hij wilt kiezen. Het moet wel in de lijst blijven van de rooms maar dit bespreken we later.   
Er wordt ook een variabel gemaakt die voor het moment geen waarde heeft, genaamd **roomIdToGetMessages**. Dit gaat geef ik later in.

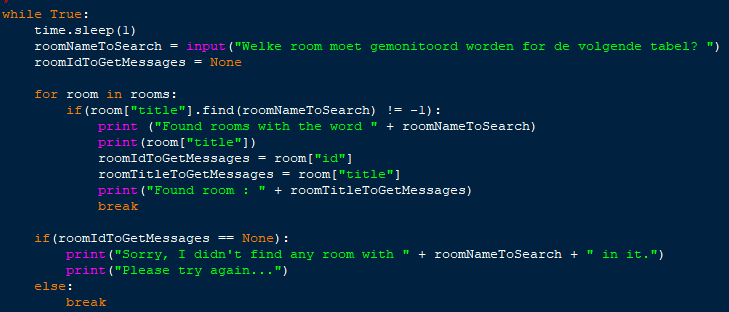


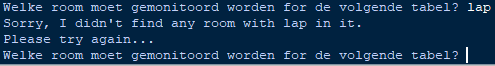
Nogmaals wordt er een **for in loop** gemaakt en daarin maken we **een if statement** voor namen die niet afgewerkt zijn op te zoeken. Dit gaan we ook afprinten zo dat gebruiker weet dat dat de room wordt opgezocht. We printen ook de ingetypte naam af en we steken de volgende gegevens in van de room in variabelen. Zoals bijvoorbeeld: de variabel “**RoomIdToGetMessages**” krijgt de roomid en “**roomTitleToGetMessages**” krijgt de naam van de room. Met deze variabel zal ook afgeprint moeten worden voor het resultaat aan de gebruiker te tonen.[[2]](#footnote-2)





De tweede **if statement** zorgt ervoor dat als er naam wordt gegeven die niet in de lijst staat dan krijgt de gebruiker de volgende meldingen, die hem vertellen dat deze room niet bestaat en dat hij een andere room moet geven. De loop zal dus opnieuw beginnen tot dat de gebruiker een bestaande room gekozen heeft die in zijn lijst staat.





## Recentste bericht tonen van de room

Deze **while loop** zal de laatste zijn want ik wil dat de er constant gecontroleerd wordt op de recentste bericht van de room.



Het eerste wat er zal gebeuren is dat de **parameters** ingesteld worden. Er wordt daarvoor een **dictionairy** van gemaakt. In die **dictionary** vragen we de roomId aan via de bestaande variabel **roomIdToGetMessages**.  
Wat we ook vragen is het aantal berichten die getoond moeten worden.



Deze **get methode** is ongeveer hetzelfde maar nu komt de accestoken in de **header** als authenticatie. Deze variabel hebben we gezien toen we aan de gebruiker vraagde voor zijn **bearer** **token**.



De volgende **if statment** zorgt ervoor als er geen bericht in de room bevindt dan wordt er een **expection** gemaakt en zal de script stoppen met runnen. Dit komt omdat ik het zo geprogrammeerd heb.

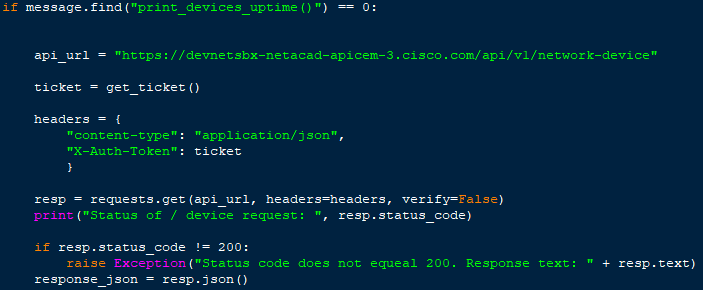


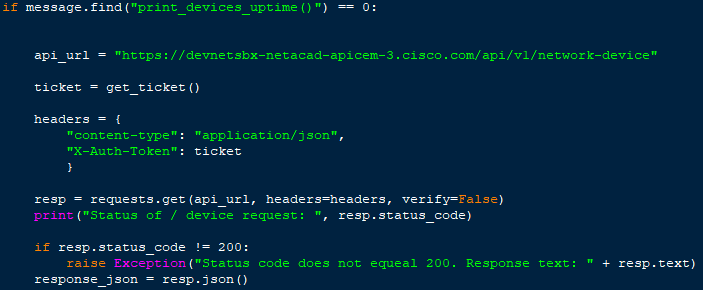




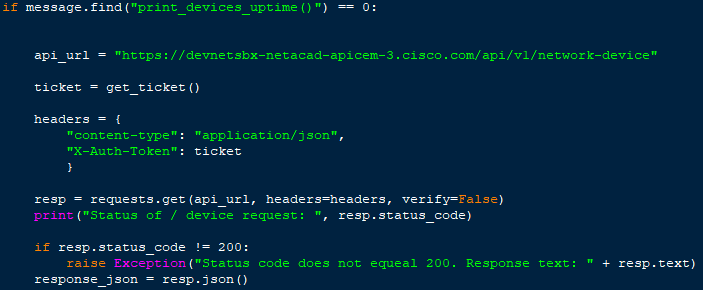
## Command aan de bot geven(print\_devices\_uptime())

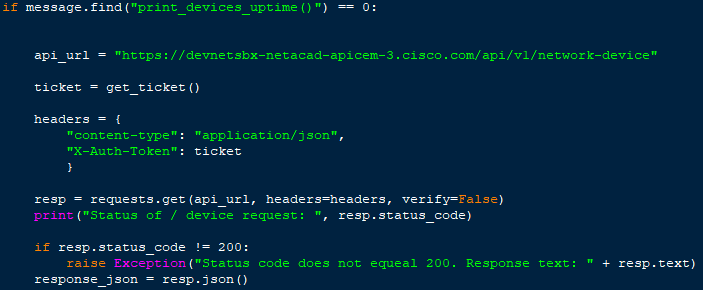
Nu kan je een **if statement** gebruiken om een command te geven in de room. Deze command zal een tabel tonen in de room die de uptime toont van de virtuele apparaten in **APIC‑EM**





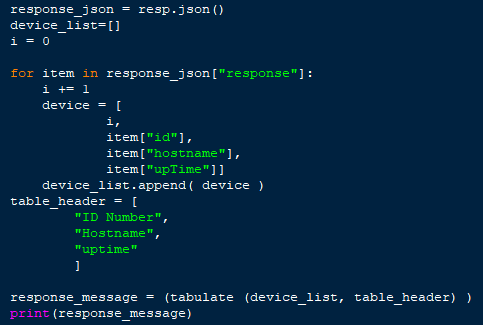
In deze **get methode** vragen we een ticket via de **functie get\_ticket()** en we stoppen hem in een variabel. Die variabel zal dan in header terecht komen als authenticatie.



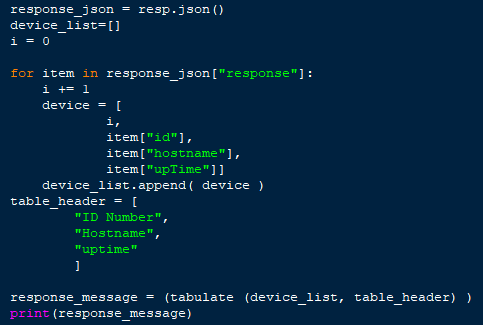


### Tabel aanmaken

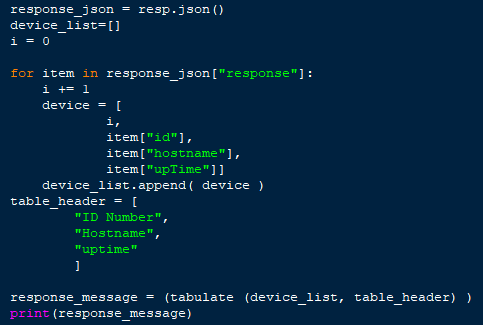
Met de gegevens die we gekregen hebben kan ik nu een tabel maken. Dit moeten level keys van de gegevens. Response is een is de eerste level. Die behoudt de andere levels die de gegevens hebben. De level 2 key hostname zal de hostname tonen van de virtuele machine.

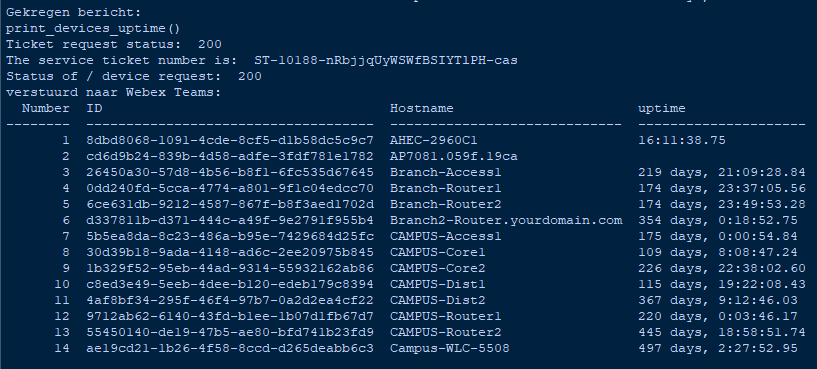


Met een **for in loop** maken we een tabel. We bespreken de elementen die gevormd zij in **JSON** formaat. Om een tabel maken moet je twee **lists** hebben. Eén die de gegevens bijhoudt en één die speelt als tabel kop.



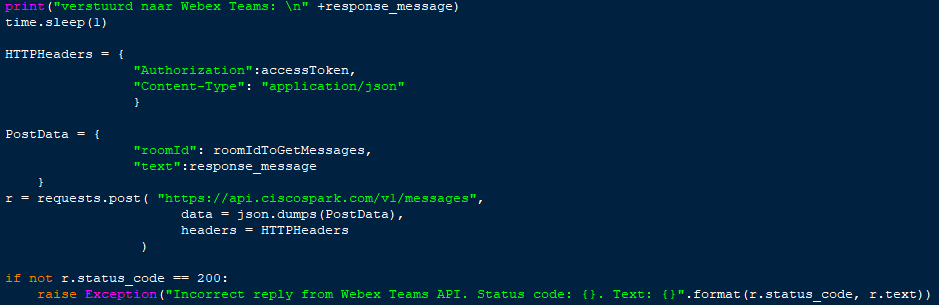
We voegen die samen in nieuwe variabel met de **module** **tabulate**. Je geeft eerst de gegevens in en dat de tabelkop.

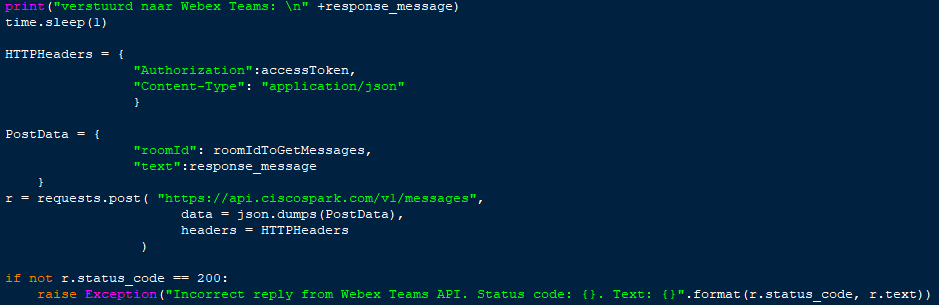




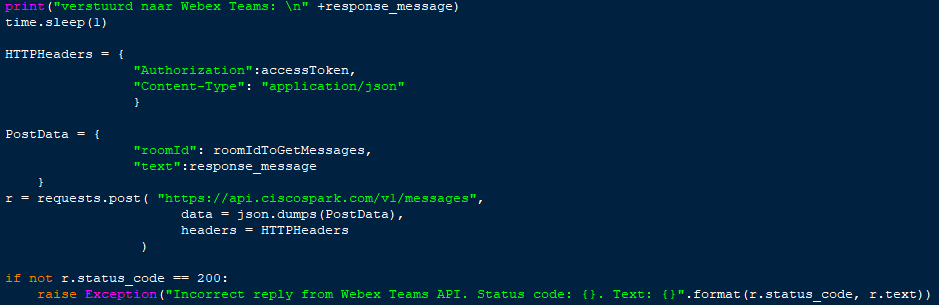
### Tabel versturen naar de gekozen room van de gebruiker

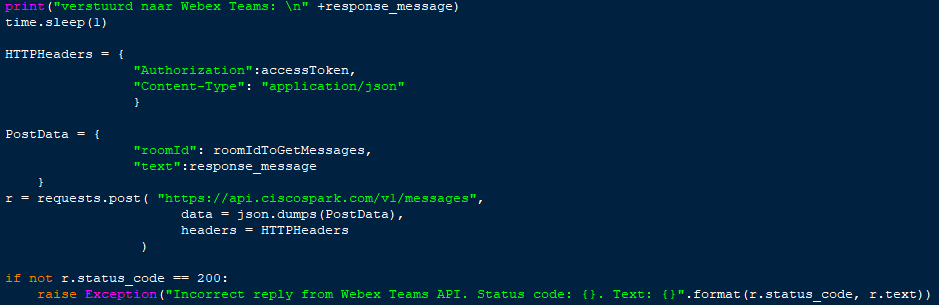
Het laatste wat ik doe bij de hoofd commando is een post methode gebruiken om de tabel als bericht te versturen.



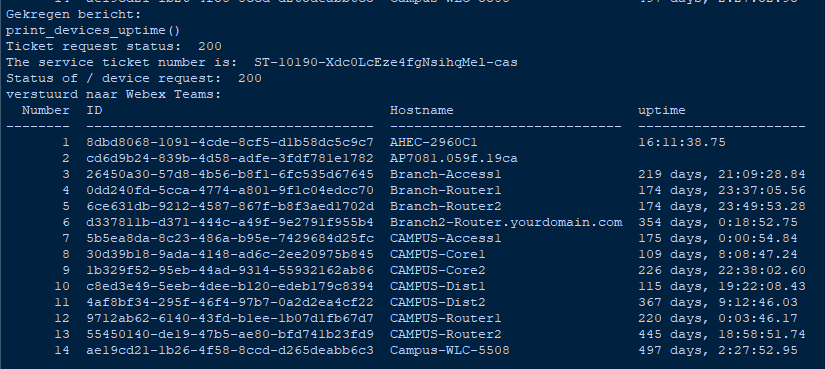


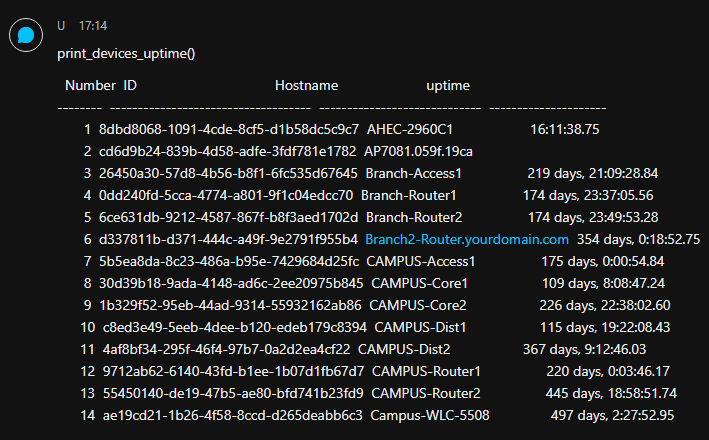
In de variabel **PostData** zal de roomId en de tekst verschijnen. De variabel **response\_message** behoort in de tekst omdat hij de tabel heeft. Postdata zal moeten inbegrepen zijn in de post methode.





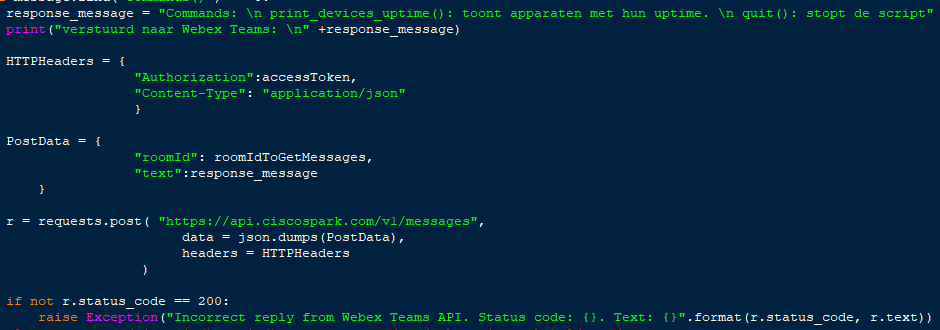
Als dat gebeurt zul je je resultaat zien niet alleen op **python** maar ook op **Webex**.

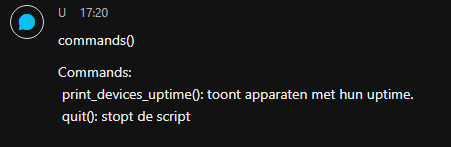




## Command commands()

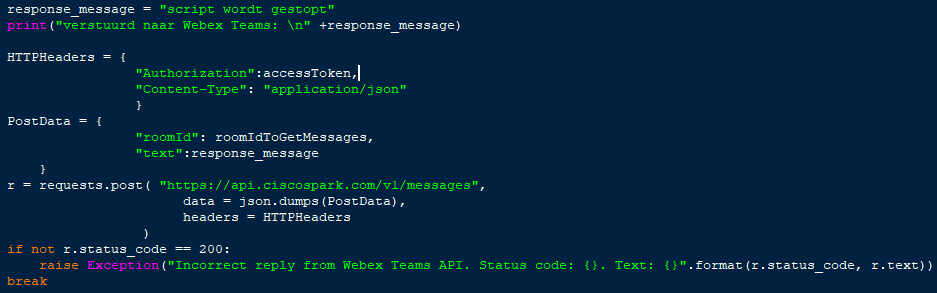
De volgende commando is kleiner omdat we maar een simpele **post methode** gebruiken. Dit is niet echt essentieel maar het is wel goed voorbeeld voor later in de praktijk. Stel nu dat de gebruiker een commando vergeten is en hij wilt hem opfrissen. Dan kan de gebruiker gewoon command zoals **help()** of **commands()** in dat geval ingeven. Om zo lijst te krijgen van **commands().**  
Het is maar een simpele string variabel die de commando’s.

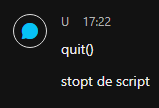


ss

## Command quit()

Omdat de bot of script nog altijd runt bij de laatste **while loop** had ik het idee om een command te maken die hem beëindigt. Deze command heet **quit()**. Met een simpele **break** statement kan je de script of deftig afsluiten. De post methode is enkel bedoeld voor de gebruiker op **Webex** om hem te laten dat de bot of script niet meer runt.





# Bronnen:

*Python time Module (with Examples). (z.d.). Geraadpleegd van https://www.programiz.com/python-programming/time*

*Json — JSON encoder and decoder — Python 3.9.1 documentation. (z.d.). Geraadpleegd van https://docs.python.org/3/library/json.html*

*Real Python. (2020, 7 november). Python’s Requests Library (Guide). Geraadpleegd van https://realpython.com/python-requests/*

*Tabulate. (2020, 22 maart). Geraadpleegd van https://pypi.org/project/tabulate/*

*Advanced Usage - urllib3 2.0.0.dev0 documentation. (z.d.). Geraadpleegd van https://urllib3.readthedocs.io/en/latest/advanced-usage.html*

*OAuth 2.0 Bearer Token Usage. (z.d.). Geraadpleegd van https://oauth.net/2/bearer-tokens/*

*Todd Fredrich, Pearson eCollege. (z.d.). HTTP Methods for RESTful Services. Geraadpleegd van https://www.restapitutorial.com/lessons/httpmethods.html*

1. Het is best om dit in af te printen of te wel beautifier te gebruiken of kleine stuk code maken die json formaat organiseerd [↑](#footnote-ref-1)
2. **TIP:** Als er rooms zijn die dezelfde reeks letters hebben wees dan iets specifieker. Het script zal automatisch de eerste resultaat nemen. Bijvoorbeeld: stel dat je twee rooms hebt die beginnen met de letter 'M'. Zet dan een ander letter erbij voor de input anders. Je kan ook de volledige naam intyppen. [↑](#footnote-ref-2)