

● Navigate here (<https://support.cylance.com/s/article/Cyl...>)



WELCOME TO CYLANCE KNOWLEDGE BASE

Knowledge Base (<https://support.cylance.com/s/KBHome>) / Detail

English

← (<https://support.cylance.com/s/KBHome>)

Back to all articles (<https://support.cylance.com/s/KBHome>)

Subscribe to this article updates

Cylance API - Install Python and PyJWT on Windows



dyamasaki

(<https://support.cylance.com/s/profile/005E0000007nOXGIA2>)

January 8, 2019 02:30 PM UTC

Note: Cylance Support does not provide assistance with installing non-Cylance programs (like Python) or security (like PyJWT). The following examples are provided AS-IS and there is no guarantee the examples will work in your environment.

Install Python and PyJWT on Windows

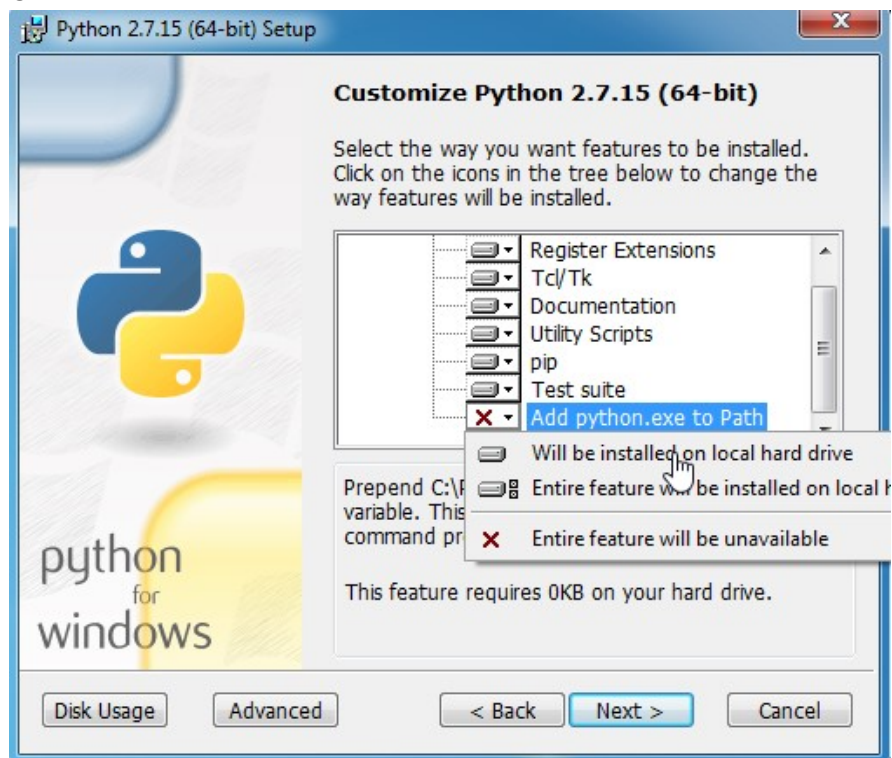
These steps are provided as an example of how to install Python, PyJWT, and Requests on a Windows system, in preparation for using the Cylance API. Installing Python advanced features is not covered in this example. Installing on other operating systems is not covered in this example.

Install Python and PyJWT

The Cylance API requires Python version 2.7 (the latest version is recommended). Python 2.6 and lower are not supported. Python 3.x is not supported.

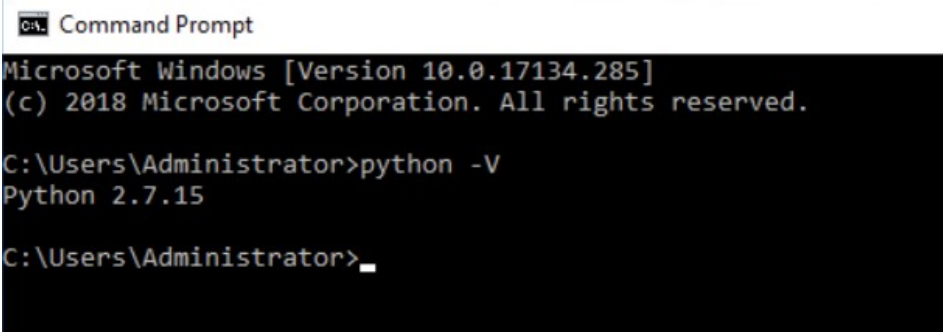
Note: This example uses Python version 2.7.15.

1. Go to the Python website and download Python 2.7. Link: <https://www.python.org/downloads/> (<https://www.python.org/downloads/>)
2. Under **Looking for a specific release?**, find the latest 2.7.x version, then click the link. The link will take you to the download page for the selected version.
3. Under **Files**, click the link for the Windows MSI installer, either x86 for 32-bit systems or x86-64 for 64-bit systems.
4. Install Python. This example uses the following settings, most are the default setting:
 - Start the Python installer.
 - Select **Install for all users** (default).
 - Directory `C:\Python27\` (default).
 - Customize Python 2.7.x, change **Add python.exe to Path** to **Will be installed on local hard drive**. This allows Command Prompt access to the Python executable. By default, this is set to **Entire feature will be unavailable**.
 - Click the dropdown next to Add python.exe to Path
 - Click Will be installed on local hard drive



- Complete the installation.
5. To verify the Python installation:
 - Open the command prompt.
 - Type `python -V`, then press **Enter**. Displays the Python version installed. Be sure to use a capital V in the command. The Python version displays in

the command prompt.



```
Command Prompt
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>python -V
Python 2.7.15

C:\Users\Administrator>_
```

6. Type `pip install pyjwt`, then press **Enter**. This installs PyJWT.
Note: With Python 2.7.9 and higher, the `pip` command is included in the installation.
7. Type `pip install requests`, then press **Enter**. This installs the requests module.

Python Example Code

Use the Python example below, adding the required token claims needed - tenant unique identifier, (`tid_val`) application unique identifier (`app_id`), and application secret (`app_secret`). For example, with `tid_val = "1234567890"`, replace **1234567890** with your tenant unique identifier. You can find the tenant unique identifier, application unique identifier, and the application secret on the Integrations page in the Console (**Settings > Integrations**).

```
import jwt # PyJWT version 1.5.3 as of the time of authoring.
import uuid
import requests # requests version 2.18.4 as of the time of authoring.
import json
from datetime import datetime, timedelta
# 30 minutes from now
timeout = 1800
now = datetime.utcnow()
timeout_datetime = now + timedelta(seconds=timeout)
epoch_time = int((now - datetime(1970, 1, 1)).total_seconds())
epoch_timeout = int((timeout_datetime - datetime(1970, 1, 1)).total_seconds())
jti_val = str(uuid.uuid4())
tid_val = "1234567890" # The tenant's unique identifier.
app_id = "ABCDEFGHJKLM" # The application's unique identifier.
app_secret = "NOPQRSTUVWXYZ" # The application's secret to sign the auth token
with.
AUTH_URL = "https://protectapi.cylance.com/auth/v2/token"
claims = {
    "exp": epoch_timeout,
    "iat": epoch_time,
    "iss": "http://cylance.com",
    "sub": app_id,
    "tid": tid_val,
    "jti": jti_val
```

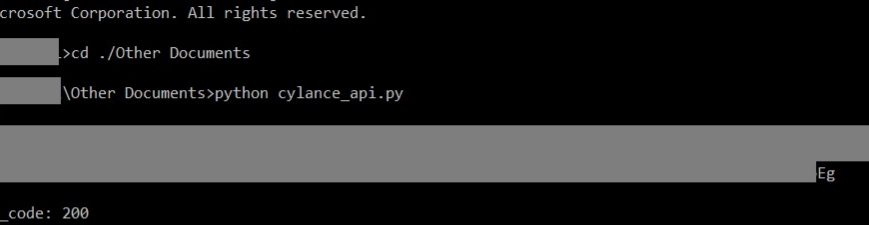
```
# The following is optional and is being noted here as an example on how one can
restrict
# the list of scopes being requested
# "scp": "policy:create, policy:list, policy:read, policy:update"
}
encoded = jwt.encode(claims, app_secret, algorithm='HS256')
print "auth_token:\n" + encoded + "\n"
payload = {"auth_token": encoded}
headers = {"Content-Type": "application/json; charset=utf-8"}
resp = requests.post(AUTH_URL, headers=headers, data=json.dumps(payload))
print "http_status_code: " + str(resp.status_code)
print "access_token:\n" + json.loads(resp.text)['access_token'] + "\n"
```

Create and Run Token PY File Using Command Prompt (Windows 10)

Because the authentication token expires, it is recommended to create a PY file with the Python example below.

Note: Using the Command Prompt to copy and paste the Access Token works with Windows 10. In Windows 7, copying and pasting includes line breaks in the token, resulting in an error when running the API call. The line breaks can be manually removed and the call re-run.

1. Open a text editor.
2. Copy and paste the Python example into the text editor.
3. Save the file with a .py extension. For example, `cylance_api.py`.
4. Open the **Command Prompt**.
5. Navigate to the folder containing the PY file (created in step 3 - `cylance_api.py`).
6. Type `python cylance_api.py`, then press **Enter**. This runs the command using the Authentication Token and returns the Access Token.
7. Highlight the Access Token (`access_token`), then right-click in the highlighted text. This copies the token and removes the highlight. Paste this token into Postman.



```
Microsoft Windows [Version 10.0.16299.785]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\...>cd ../Other Documents

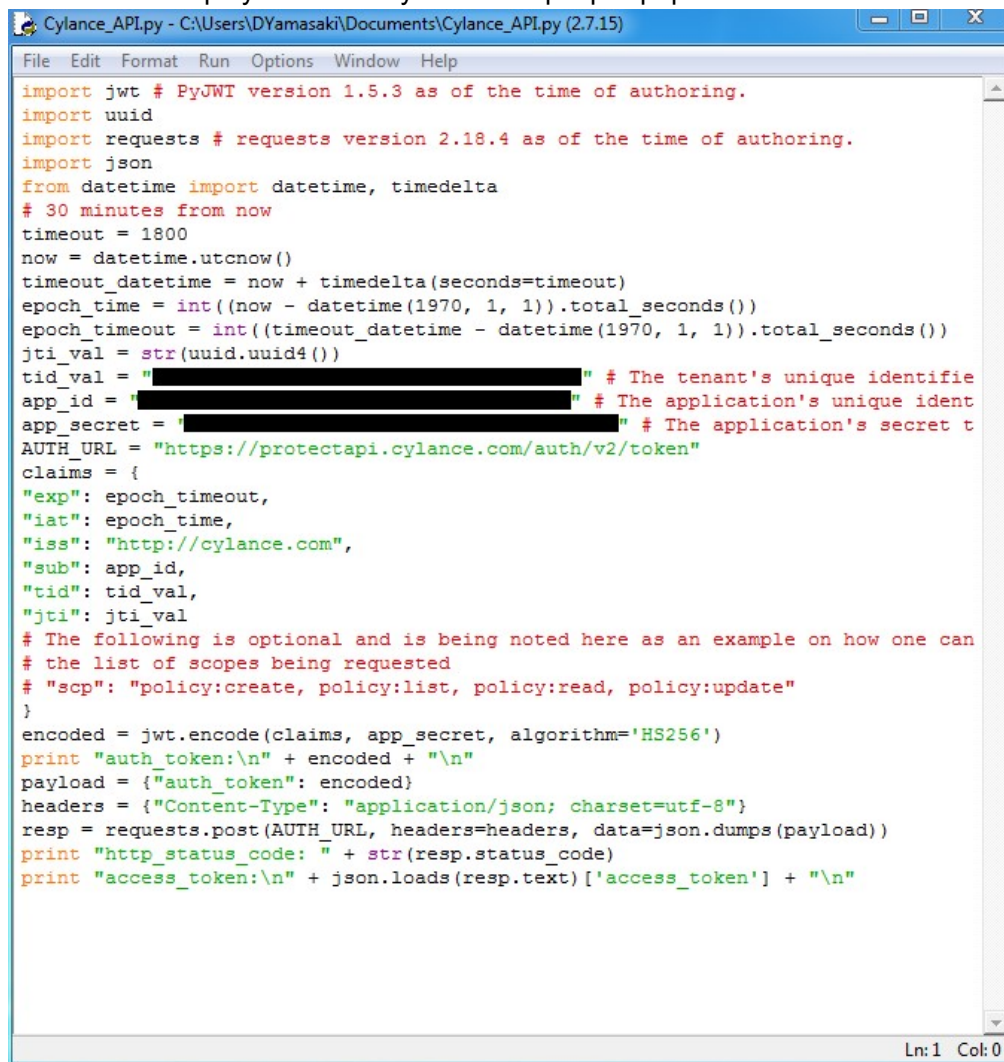
C:\Users\...\Other Documents>python cylance_api.py
auth_token:
http_status_code: 200
access_token:
```

Create and Run Token PY File Using IDLE (Python GUI)

Because the authentication token expires, it is recommended to create a PY file with the Python example below.

1. Open a text editor.
2. Copy and paste the Python example into the text editor.
3. Save the file with a .py extension. For example, `cylance_api.py`.
4. Open **IDLE (Python GUI)**. In Windows 10, click **Start**, type `idle`, then select **IDLE (Python GUI)**.
5. Select **File > Open**.

6. Select the PY file (created in step 3 - `cylance_api.py`), then click **Open**. A new window displays with the Python example pre-populated.



```
import jwt # PyJWT version 1.5.3 as of the time of authoring.
import uuid
import requests # requests version 2.18.4 as of the time of authoring.
import json
from datetime import datetime, timedelta
# 30 minutes from now
timeout = 1800
now = datetime.utcnow()
timeout_datetime = now + timedelta(seconds=timeout)
epoch_time = int((now - datetime(1970, 1, 1)).total_seconds())
epoch_timeout = int((timeout_datetime - datetime(1970, 1, 1)).total_seconds())
jti_val = str(uuid.uuid4())
tid_val = "XXXXXXXXXXXXXXXXXXXX" # The tenant's unique identifier
app_id = "XXXXXXXXXXXXXXXXXXXX" # The application's unique identifier
app_secret = "XXXXXXXXXXXXXXXXXXXX" # The application's secret token
AUTH_URL = "https://protectapi.cylance.com/auth/v2/token"
claims = {
    "exp": epoch_timeout,
    "iat": epoch_time,
    "iss": "http://cylance.com",
    "sub": app_id,
    "tid": tid_val,
    "jti": jti_val
}
# The following is optional and is being noted here as an example on how one can
# the list of scopes being requested
# "scp": "policy:create, policy:list, policy:read, policy:update"
}
encoded = jwt.encode(claims, app_secret, algorithm='HS256')
print "auth_token:\n" + encoded + "\n"
payload = {"auth_token": encoded}
headers = {"Content-Type": "application/json; charset=utf-8"}
resp = requests.post(AUTH_URL, headers=headers, data=json.dumps(payload))
print "http_status_code: " + str(resp.status_code)
print "access_token:\n" + json.loads(resp.text)['access_token'] + "\n"
```

7. Select Run > Run Module. The module runs and returns an access token. Copy the token and paste it into Postman. In the example below, the access token starts with `eyJ` and ends with `Mjc`.

A screenshot of a Windows command prompt window titled "Python 2.7.15 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The terminal shows the execution of a Python script named "cyalance_api.py" located at "C:\Users\██████\Documents\". The script outputs an authentication token as a long alphanumeric string. The first part of the token is visible: "eyJ0eXA6OiQkODQyLWY1LTUxMjEwLWVhbnRlLnNpdG86MTIzNDU2Njc4OTBkaC8G". The rest of the token is obscured by a black redaction box. Below the token, the status code "http_status_code: 200" and another label "access_token:" are shown. A second, longer alphanumeric string follows, also partially redacted. The visible portion of this second string is "eyJ0eXA6OiQkODQyLWY1LTUxMjEwLWVhbnRlLnNpdG86MTIzNDU2Njc4OTBkaC8G". The terminal ends with a prompt ">>> |". At the bottom right, the status bar indicates "Ln: 12 Col: 4".

1 likes • 83 views

Related Articles

Windows Defender exclusions for Cylance

3/7/2018

(<https://support.cylance.com/s/article/kA044000000HfwXCAS/Windows-Defender-exclusions-for-Cylance>)

Install CylanceOPTICS - Windows

8/28/2018

(<https://support.cylance.com/s/article/kA0440000000p04CAA/Install-Optics77>)

CylancePROTECT - Windows: Server Core Installation

9/10/2018

(<https://support.cylance.com/s/article/kA044000000HfZiCAK/CylancePROTECT-Windows-Server-Core-Installation>)

CylanceOPTICS API Update Release Notes - November 28 2018

12/14/2018

(<https://support.cylance.com/s/article/kA044000000Q4uGCAS/CylanceOPTICS-API-Update-Release-Notes-November-28-2018>)

Cylance Security Bulletin – Cy2018-002 – Moderate

5/2/2018

(<https://support.cylance.com/s/article/kA044000000Hg5KCAS/Cylance-Security-Bulletin-Cy2018-002-Moderate>)

Cylance Security Bulletin: Cy2018-001

5/2/2018

(<https://support.cylance.com/s/article/kA044000000Hg5FCAS/Cylance-Security-Bulletin-Cy2018-001-Important>)

When do scheduled releases and maintenance windows happen?

6/4/2018

(<https://support.cylance.com/s/article/kA044000000pEuCAI/When-do-scheduled-releases-and-maintenance-windows-happen>)

How to alter MSI installers and include proxy settings and verbose logging

2/27/2018

(<https://support.cylance.com/s/article/kA044000000CoNbCAK/How-to-alter-MSI-installers-and-include-proxy-settings-and-verbose-logging>)

Cylance OPTICS - Windows: Command Line Options

11/5/2018

(<https://support.cylance.com/s/article/kA044000000XiQ2CAK/CylanceOPTICS-Command-Line-Options>)

Self Protection Level: Best Practices for Installing and Securing CylancePROTECT for Windows

8/16/2018




(<https://support.cylance.com/s/article/kA044000000Q1GLCA0/Self-Protection-Level-Best-Practices-for-Installing-and-Securing-CylancePROTECT-for-Windows>)

Announcements

(<https://support.cylance.com/s/KBHome?categoryparam=Announcements>)

+

CylanceSUPPORT (https://support.cylance.com/s/KBHome?categoryparam=CylanceSUPPORT)	+
CylancePROTECT (https://support.cylance.com/s/KBHome?categoryparam=CylancePROTECT)	+
CylancePROTECT API (https://support.cylance.com/s/KBHome?categoryparam=CylancePROTECT API)	+
CylancePUBLIC (https://support.cylance.com/s/KBHome?categoryparam=CylancePUBLIC)	+
CylanceOPTICS (https://support.cylance.com/s/KBHome?categoryparam=CylanceOPTICS)	+
CylanceSUPPORT Site (https://support.cylance.com/s/KBHome?categoryparam=CylanceSUPPORT Site)	+

Need Support?	
Can't find the answer you're looking for? Contact us, we are here to help	
	Knowledge Base >
	Ask A Question >
	Contact Support >



Submit a case

(<https://support.cylance.com/s/NewCase>)



(<https://www.youtube.com/user/CylanceInc>)



(<https://www.linkedin.com/company/cylanceinc>)



(<https://www.facebook.com/CylanceInc>)



(<https://twitter.com/cylanceinc>)

400 Spectrum Center Drive, Suite 900, Irvine, CA 92618 USA ([https://maps.google.com/?q=400 Spectrum Center Drive, Suite 900, Irvine, CA 92618 USA](https://maps.google.com/?q=400%20Spectrum%20Center%20Drive%2C%20Suite%20900%2C%20Irvine%2C%20CA%2092618%20USA))

Call Us: +1 866 699 9689 (tel:+1 866 699 9689) Get Support: +44 203 4436394 (tel:+1 866 699 9689)

© 2018 Cylance Inc. All Rights Reserved Privacy Policy (<https://www.cylance.com/privacy-policy>) | Terms Of Service (<https://www.cylance.com/terms>)