**API Authorization and Authentication**

In Python, we can use **pyjwt** library to generate and verify api jwt tokens. Here are the steps to implement it. <https://pyjwt.readthedocs.io/en/latest/>

**Generate token:**

1. Install pyjwt library and import jwt.

pip install pyjwt

import jwt

1. Create payload in json format.

payload = {‘userid’: 1}

1. Create secret key.

secret\_key = “secret”

1. Generate token.

token = jwt.encode(payload, secret\_key, algorithm='HS256').decode('utf-8')

1. Send this token to user.

**Verify token and access api:**

When user wants to access our api, they should first generate tokens and access api with their token and secret key.

import jwt

secret\_key = "secret”

token = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxMjN9.oF\_jJKavmWrM6d\_io5M5PBiK9AKMf\_OcK4xpc17kvwI'

payload = jwt.decode(token, secret\_key, algorithms=['HS256'])

This payload contains userid. We have to check user api limit and give access to api.

**Other Libraries:**

In python, we also have libraries to implement JWT and OAuth.

* authlib – It includes OAuth and JWT authentication. <https://docs.authlib.org/en/latest/>
* python-jose – It includes JWT authentication. <https://python-jose.readthedocs.io/en/latest/jwt/api.html>
* jwcrypto – It includes JWT authentication. <https://jwcrypto.readthedocs.io/en/latest/>

Best one to implement is authlib library for OAuth authentication.

**RATE LIMIT :**

To limit the api hits, we can use **ratelimit** library. <https://pypi.org/project/ratelimit/>

from ratelimit import limits

import requests

DAY= 86400

@limits(calls=100, period=DAY)

def call\_api(url):

response = requests.get(url)

if response.status\_code != 200:

raise Exception('API response: {}'.format(response.status\_code))

return response

**Other libraries:**

* Flask-Limiter - <https://flask-limiter.readthedocs.io/en/stable/>