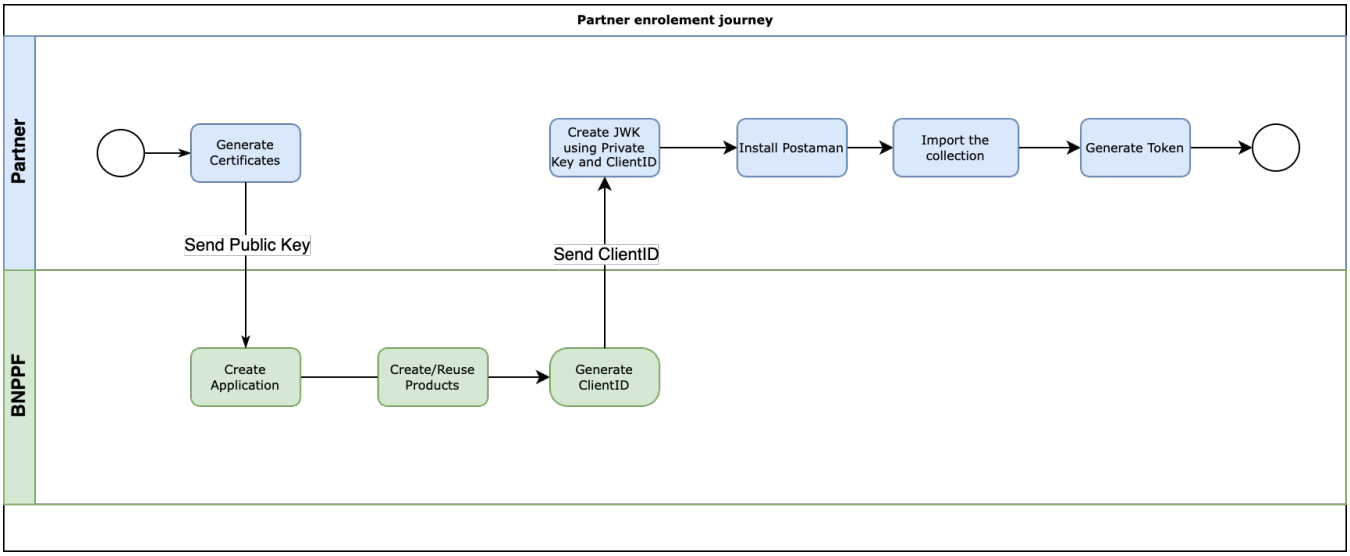
Apigee partner enrolment journey OAuth2



How to create the client Id & transmit the certificate

Environement	APIGEE URLs
Sandbox	apis-ce.staging.bnpparibas-pf.com
Staging	apis-ce.staging.bnpparibas-pf.com
Pre-Production	api-ppce.bnpparibas-pf.com
Production	api-ce.bnpparibas-pf.com

Step 1 : Use Openssl to create the private key and the certificate

Open your cmd and execute the next command

```
openssl req -x509 -newkey rsa:2048 -keyout private.key -out public.cert -days 365
```

ou should have now a certificate with this format:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

And a private key you will use to sign the certificate with the next format

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
...
-----END ENCRYPTED PRIVATE KEY-----
```

Step 2 : Transform your private key to JWK

For he Postman collection test, you will have to transform your private key to JWK:

Option 1

Use the Python script provided bellow in the same directory of your certificate.

Prerequisites

Install Python if needed : https://www.python.org/
Add crypto lib using PIP: **pip3 install cryptography==41.0.5**
Launch the script : **python3 convert\ jwk.py**

Transform private key to JWKS

```
import json import base64 from cryptography.hazmat.primitives import serialization from cryptography.hazmat.
primitives.asymmetric import rsa with open('private.decrypted.key', 'rb') as f: private_key_pem = f.read()
private_key = serialization.load_pem_private_key(private_key_pem, password=None) n = base64.urlsafe_b64encode
(int(private_key.private_numbers().public_numbers.n).to_bytes(256, byteorder='big')).rstrip(b'=').decode('utf-
8') #e = base64.urlsafe_b64encode(int(private_key.private_numbers().public_numbers.e).to_bytes(4,
byteorder='big')).rstrip(b'=').decode('utf-8') d = base64.urlsafe_b64encode(int(private_key.private_numbers().
d).to_bytes(256, byteorder='big')).rstrip(b'=').decode('utf-8') p = base64.urlsafe_b64encode(int(private_key.
private_numbers().p).to_bytes(128, byteorder='big')).rstrip(b'=').decode('utf-8') q = base64.urlsafe_b64encode
(int(private_key.private_numbers().q).to_bytes(128, byteorder='big')).rstrip(b'=').decode('utf-8') dp = base64.
urlsafe_b64encode(int(private_key.private_numbers().d % (private_key.private_numbers().p - 1)).to_bytes(128,
byteorder='big')).rstrip(b'=').decode('utf-8') dq = base64.urlsafe_b64encode(int(private_key.private_numbers().
d % (private_key.private_numbers().q - 1)).to_bytes(128, byteorder='big')).rstrip(b'=').decode('utf-8') qi =
base64.urlsafe_b64encode(pow(private_key.private_numbers().q, -1, private_key.private_numbers().p).to_bytes
(128, byteorder='big')).rstrip(b'=').decode('utf-8') jwk = { "kty": "RSA", "n": n, "e": "AQAB", "d": d, "p": p,
"q": q, 'dp': dp, 'dq': dq, 'qi': qi } with open('jwk.json', 'w') as fg: json.dump(jwk, fg, indent=2) print
('your decrypted private key was converted to jwk and writed in jwk.json file to check your assertion')
```

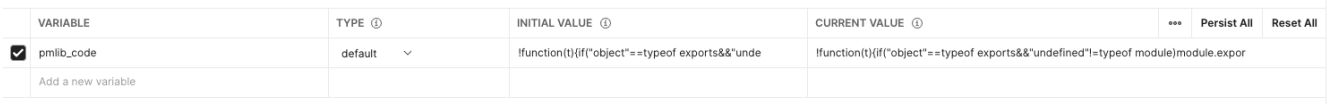
Option 2

PEM to JWK converter - Tribestream

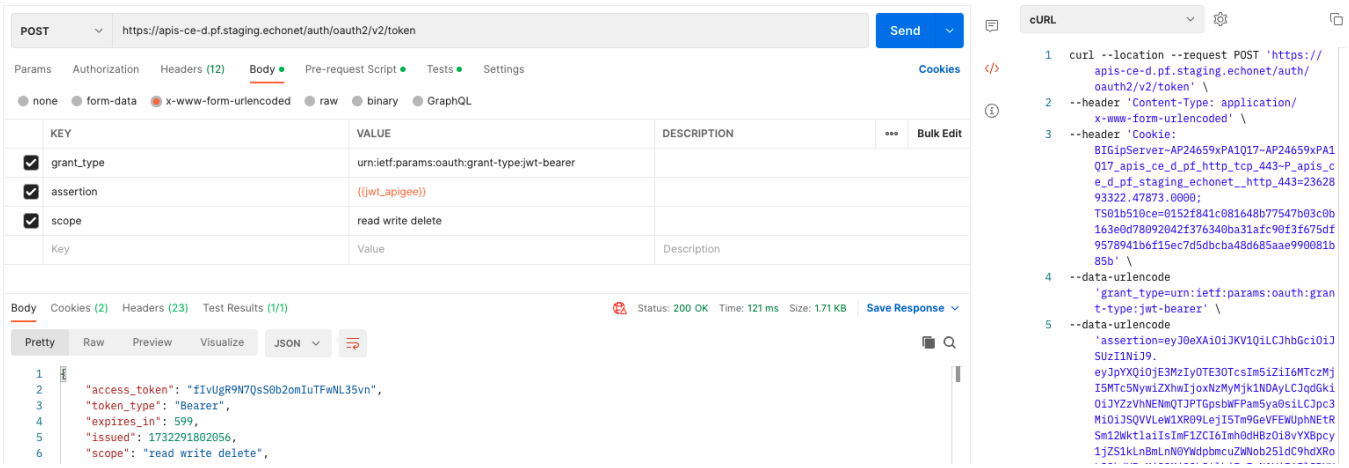
How to request an access token

Step 1 : Install Postman

Step 2: Create a postman global variable



Step 3: Create the postman Body request



Step 4 : Store the access_token as a variable to use it as authorization in the APIs call