Below is a step by step guide, which can help build Snaplogic pipeline for JWT token generation and usage :

**JWT Token Generation :**

Snaplogic comes with an inbuilt snap called as **JWT Generate** which helps to generate a JSON Web Token. Inputs needed for these are:

**Header**: Which specifies the algorithm to be used

 Sample Json structure:

 header **=** {

typ: "JWT",

   alg: "RS256",

};

**Claims**: Which specifies the connectivity details

 Sample Json structure:

 claims **=** {

   aud: "https://login.XXX.com",

   scope: "urn:portal:app:XXXX",

   iss: "urn:portal:acct:XXXX"

};

**Signature**: Which is created using a Keystore that has an encoded private, private key combination.

 Private(.pem), Public key(.pem), Keystore are generated using below open SSL commands:

Step 1: First generate private key and cert

openssl req -x509 -newkey rsa:2048 -keyout private.pem -out cert.pem –nodes

Step 2: Convert private key and cert in P12 format
openssl pkcs12 -export -inkey private.pem -in cert.pem -out keyStore.p12

Step 3: Extract public key from private key
openssl rsa -in private.pem -pubout > public.pub

If we have these information handy, then let’s get started to map these into Snaplogic snaps as below:

**Step1: Take a mapper snap and define all the parameters for building the Claims part as below:**



**Step 2: Build the Claims entity as below:**

{ aud: [$aud] , scope: $scope, iss: $iss}



**Step 3: Give details in JWT Generate Snap as below:**

 

**Step 4: Configure the JWT account:**

Under the account Tab click Add account :



Create the account by giving below details:

****

**Once account is created, click validate button and validate the account.**

**Step 5: Bravo! Once these configurations are done, you would have the JWT token which can then be extracted using a mapper as shown below:**



The pipeline till now would look something like this:



Once the JWT token is generated, the actual OAuth access token can be generated by making rest call to the APIs exposed as below:





An OAuth access token would then be generated.

**Moving to the 2nd challenge, we now need to generate a dynamic Json structure. This can be achieved using JSON Generator Snap as shown below:**



**Click the Edit JSON and provide the code, in Format Code Section, to build the Json dynamically based on the input data elements:**

**A code snippet would look something like this:**



**The output of Json generator would look like :**



**Point to note:** The file to be uploaded using rest post, is a **.json file**, but here the **output of JSON generator is an array**. We can either handle this by **having a mapper after the JSON Generator Snap** **and** **converting the array object to JSON**, **or by using Formatter Snap, and enabling the option “Json Lines”** as below:

**Once our connection, file is ready, we can move to our challenge 3, to transfer this file to a rest API using REST POST Snap as below:**

1. We had to do the file transfer not as a body content, but as a **Json file upload**. To achieve this, we wrote the output from JSON generator to SLDB first using JSON Formatter, File Writer Snap first, then took the file name using a mapper and passed it to Rest POST snap and read back the file from SLDB for transfer. The pipeline would look like below:



The output of mapper after File Writer snap would be file name, which can thereafter be used as a parameter to read from SLDB in the REST post snap.



**The Rest post snap for file transfer would look like below:**

**Note:** We can use Single file upload/ Multiple File upload option in rest post based on the number of files we want to upload to Rest URL. To upload multiple files, specify similar details under “Form Upload”. Here I have only 1 file to upload so I am using the Single file upload option as below:





**Note:** Once the file transfer is complete, for security reason it is advisable to remove the file stored in SLDB. For this a **File Delete** Snap can be used as below:



The overall pipeline looks like:



Thanks for reading on, have a nice time exploring!!