**Check list for Windows Environment Setup for Cognitive Claim Python Development with Pycharm**

**Step 1 – Install Anaconda and Pycharm**

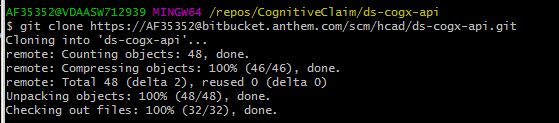
**Both program could be pushed by requesting SNOW ticket**

**Step 2 – Download Source code from BitBucket:**

Start Anaconda Prompt from Start manual:

Download the BitBucket code:

git clone <https://afxxxxx@bitbucket.anthem.com/scm/hcad/ds-cogx-api.git>



**Step 3 – Create conda environment (virtualEnv) for the project**

- Disable two dependencies within the ./env/environment.yml (The file is within the source code)

----------------------

name: cognitive-claims

channels:

- conda-forge

- anaconda

- defaults

dependencies:

- numpy==1.15.1

- pandas==0.23.4

- pickleshare==0.7.4

- pyhamcrest==1.9.0

- scikit-image==0.14.0

- scikit-learn==0.20.0

- scipy==1.1.0

# - xgboost==0.81

- flasgger==0.9.2

- flask==1.0.2

# - gunicorn==19.9.0

- python==3.7.0

- htmlmin==0.1.12

- requests==2.21.0

- requests-kerberos==0.12.0

- flask-restplus

- pytest

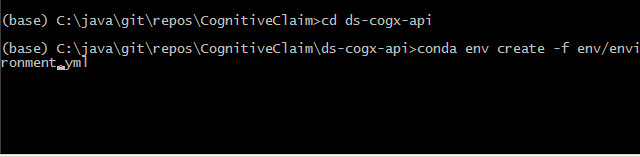
prefix: /usr/local/envs/cognitive-claims

---------------------

Note: These two dependencies does not exist in the anaconda (base) and conda install will have issues. We will add them manually later after created the virtual environment

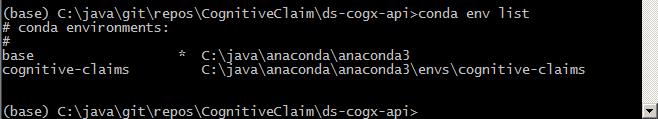
- Now, it is time to create the virtual environment within Anaconda prompt

conda env create -f env/environment.yml



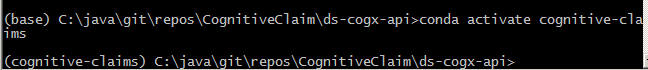
Conda env list

- This command will help us to find out the existing environment



conda activate cognitive-claims

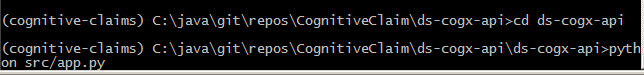
- make a note of the location for the cognitive-claims – you will need it later

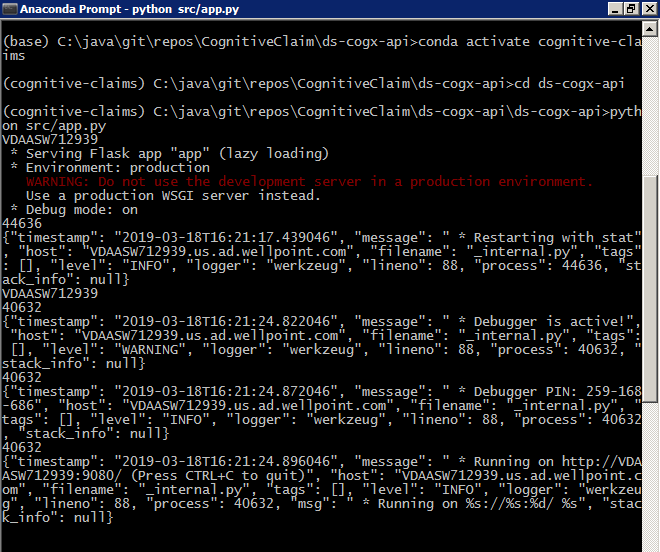


- Now, you are in the virtual environment of cognitive-claims

**Step 4 - Test run the program:**

Python src/app.py

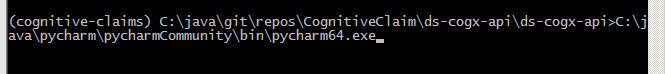




**Step 5 – It is time to make pycharm works**

- First exist the running mode from the anaconda prompt by contrl+C – We are going to need this environment (This is the interesting part)

- Second, manual start pycharm64.exe within the anaconda virtual environment you created:



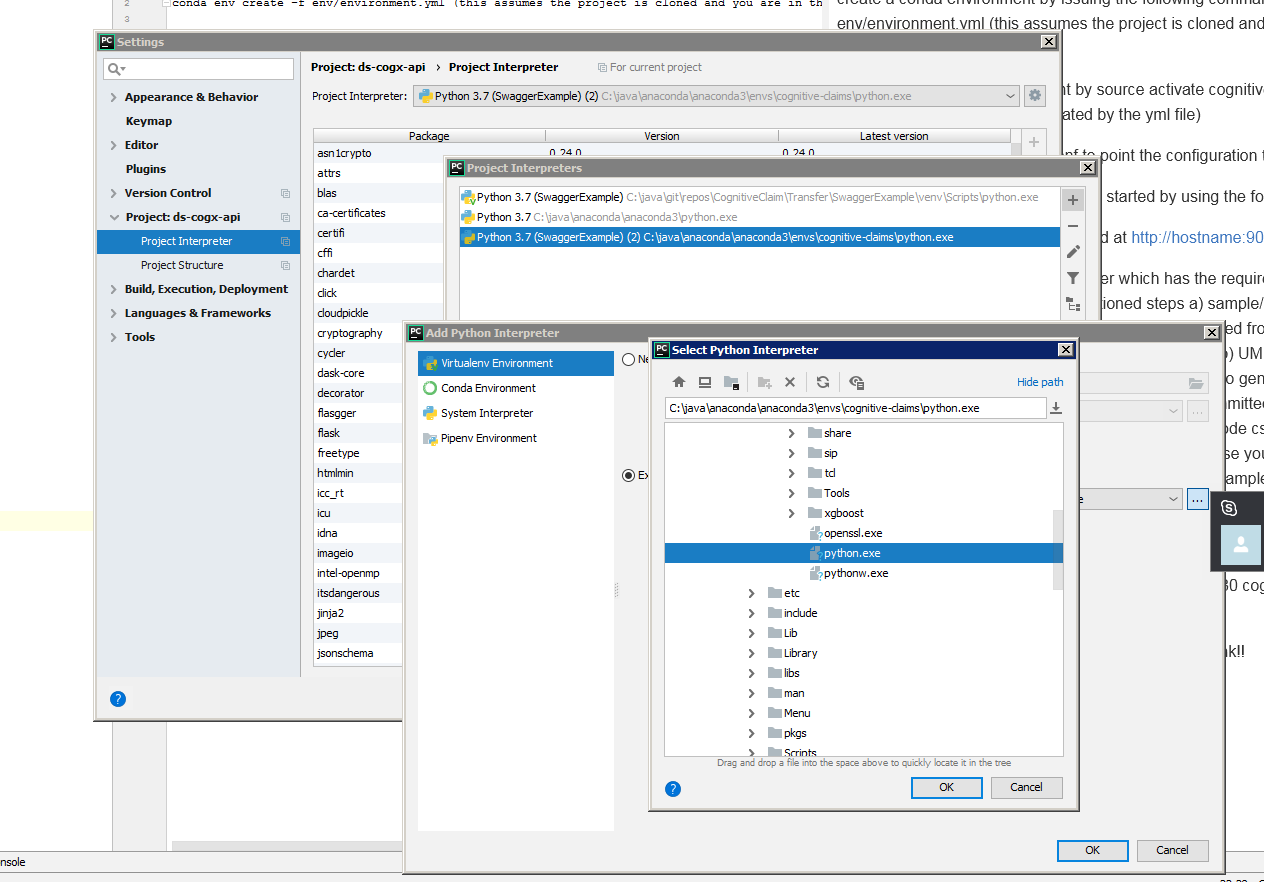
Note: this is the tricky part

- You need to start the pycharm64.exe, not the pycharm.exe

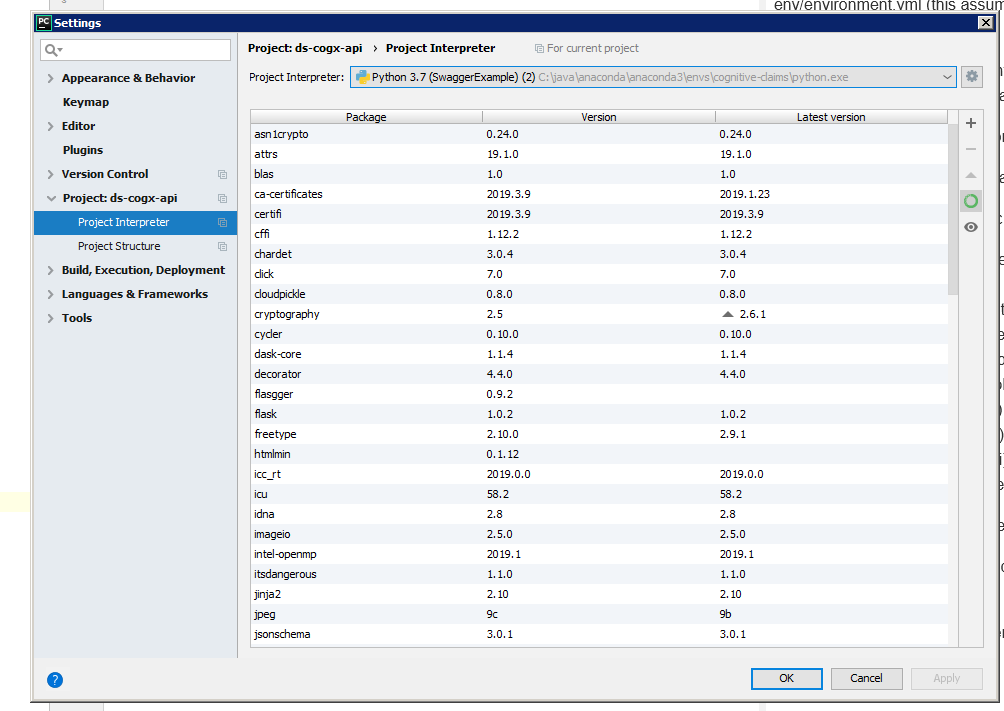
- You need to do this within the virtual environment you created. This is because the default pycharm environment will have conflict with the conda virtual environment, and if you start pycharm outside anaconda, some packages will stop working. And the frustration part is: some of the packages works. It is really difficult to troubleshooting. To avoid that issue, we run pycharm64.exe within the virtual environment.

**Step 6 – Create the virtual environment within Pycharm to sync with conda**

- File -> settings -> Project Interpreter -> show all -> new + Virtualenv Environment -> {Select the location of your conda virtual environment}



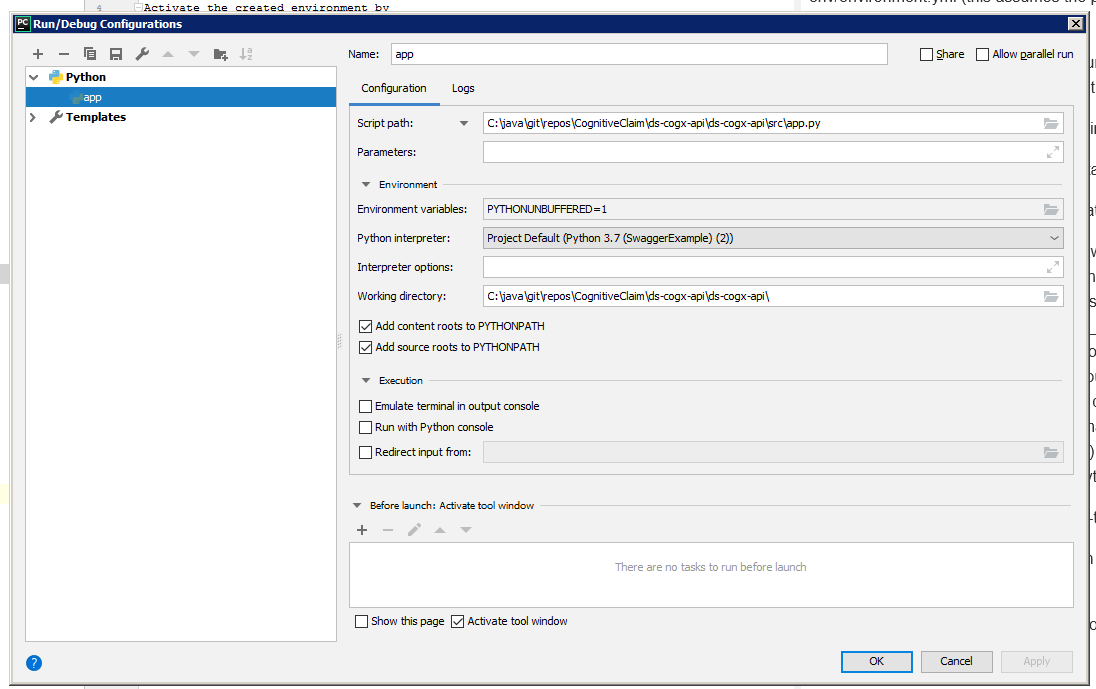
- Verify all the fancy dependencies you included in your conda virtual environment should exist here now:



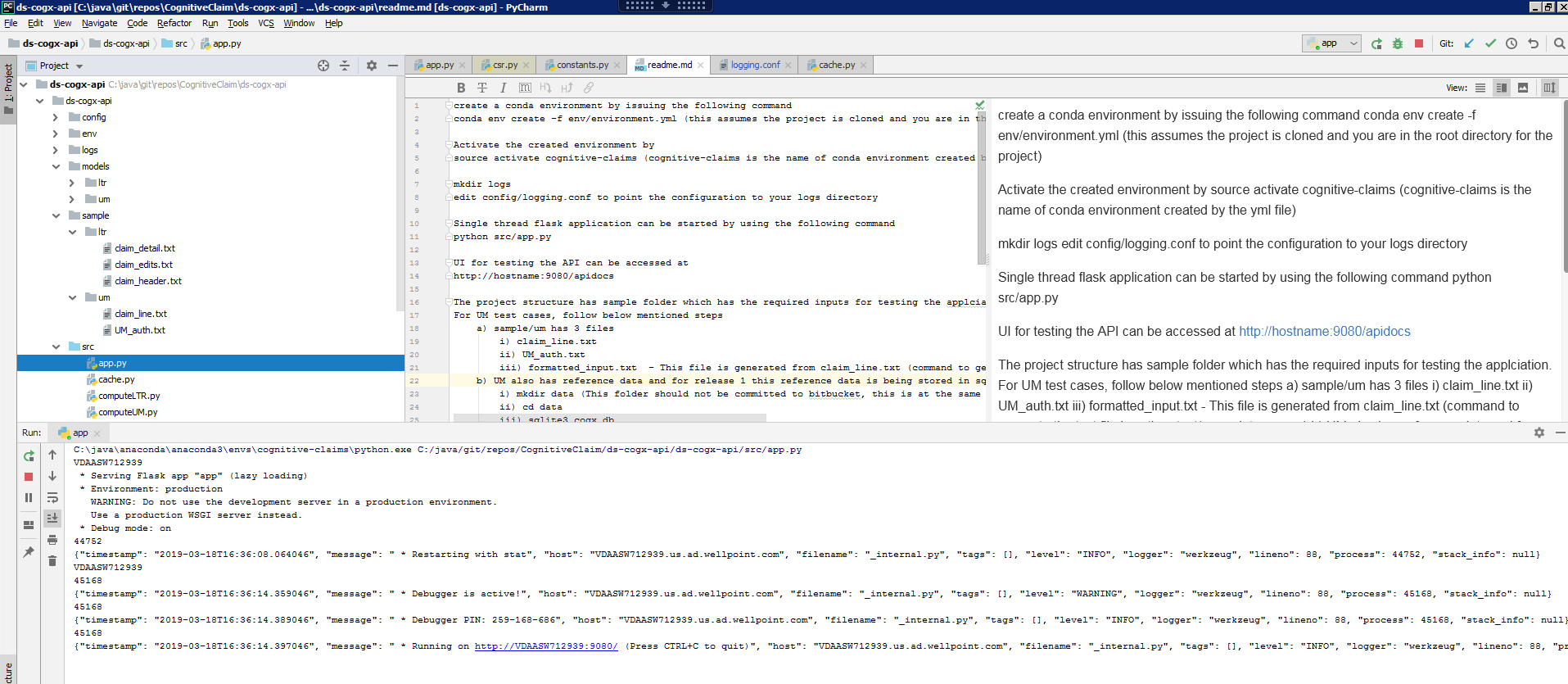
**Step 7 – Time to run the program**

- First configure the run with the right parameter and working directory:

Run -> edit configuration:



- If you get the following screen, congratulations, you did it



**Step 8 – Verify the service and connection:**

- Use chrome and paste the URL: <http://localhost:9080/apidocs>

You should have the following:

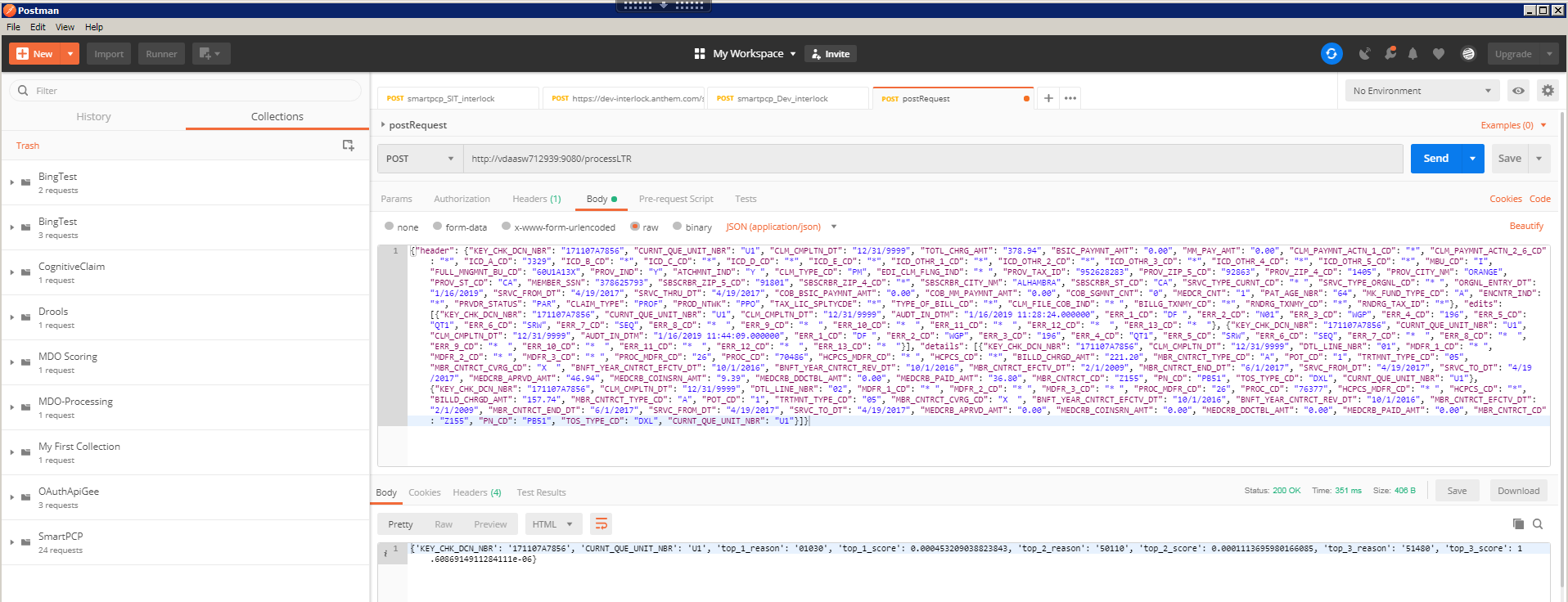


- If you have postman setup (We highly recommend you to install postman or SoapUI for testing), you can try to create the request collection by the following post message:

-------------------------

{"header": {"KEY\_CHK\_DCN\_NBR": "171107A7856", "CURNT\_QUE\_UNIT\_NBR": "U1", "CLM\_CMPLTN\_DT": "12/31/9999", "TOTL\_CHRG\_AMT": "378.94", "BSIC\_PAYMNT\_AMT": "0.00", "MM\_PAY\_AMT": "0.00", "CLM\_PAYMNT\_ACTN\_1\_CD": "\*", "CLM\_PAYMNT\_ACTN\_2\_6\_CD": "\*", "ICD\_A\_CD": "J329", "ICD\_B\_CD": "\*", "ICD\_C\_CD": "\*", "ICD\_D\_CD": "\*", "ICD\_E\_CD": "\*", "ICD\_OTHR\_1\_CD": "\*", "ICD\_OTHR\_2\_CD": "\*", "ICD\_OTHR\_3\_CD": "\*", "ICD\_OTHR\_4\_CD": "\*", "ICD\_OTHR\_5\_CD": "\*", "MBU\_CD": "I", "FULL\_MNGMNT\_BU\_CD": "60U1A13X", "PROV\_IND": "Y", "ATCHMNT\_IND": "Y ", "CLM\_TYPE\_CD": "PM", "EDI\_CLM\_FLNG\_IND": "\* ", "PROV\_TAX\_ID": "952628283", "PROV\_ZIP\_5\_CD": "92863", "PROV\_ZIP\_4\_CD": "1405", "PROV\_CITY\_NM": "ORANGE", "PROV\_ST\_CD": "CA", "MEMBER\_SSN": "378625793", "SBSCRBR\_ZIP\_5\_CD": "91801", "SBSCRBR\_ZIP\_4\_CD": "\*", "SBSCRBR\_CITY\_NM": "ALHAMBRA", "SBSCRBR\_ST\_CD": "CA", "SRVC\_TYPE\_CURNT\_CD": "\* ", "SRVC\_TYPE\_ORGNL\_CD": "\* ", "ORGNL\_ENTRY\_DT": "1/16/2019", "SRVC\_FROM\_DT": "4/19/2017", "SRVC\_THRU\_DT": "4/19/2017", "COB\_BSIC\_PAYMNT\_AMT": "0.00", "COB\_MM\_PAYMNT\_AMT": "0.00", "COB\_SGMNT\_CNT": "0", "MEDCR\_CNT": "1", "PAT\_AGE\_NBR": "64", "MK\_FUND\_TYPE\_CD": "A", "ENCNTR\_IND": "\*", "PRVDR\_STATUS": "PAR", "CLAIM\_TYPE": "PROF", "PROD\_NTWK": "PPO", "TAX\_LIC\_SPLTYCDE": "\*", "TYPE\_OF\_BILL\_CD": "\*", "CLM\_FILE\_COB\_IND": "\* ", "BILLG\_TXNMY\_CD": "\*", "RNDRG\_TXNMY\_CD": "\*", "RNDRG\_TAX\_ID": "\*"}, "edits": [{"KEY\_CHK\_DCN\_NBR": "171107A7856", "CURNT\_QUE\_UNIT\_NBR": "U1", "CLM\_CMPLTN\_DT": "12/31/9999", "AUDT\_IN\_DTM": "1/16/2019 11:28:24.000000", "ERR\_1\_CD": "DF ", "ERR\_2\_CD": "N01", "ERR\_3\_CD": "WGP", "ERR\_4\_CD": "196", "ERR\_5\_CD": "QT1", "ERR\_6\_CD": "SRW", "ERR\_7\_CD": "SEQ", "ERR\_8\_CD": "\*  ", "ERR\_9\_CD": "\*  ", "ERR\_10\_CD": "\*  ", "ERR\_11\_CD": "\*  ", "ERR\_12\_CD": "\*  ", "ERR\_13\_CD": "\*  "}, {"KEY\_CHK\_DCN\_NBR": "171107A7856", "CURNT\_QUE\_UNIT\_NBR": "U1", "CLM\_CMPLTN\_DT": "12/31/9999", "AUDT\_IN\_DTM": "1/16/2019 11:44:09.000000", "ERR\_1\_CD": "DF ", "ERR\_2\_CD": "WGP", "ERR\_3\_CD": "196", "ERR\_4\_CD": "QT1", "ERR\_5\_CD": "SRW", "ERR\_6\_CD": "SEQ", "ERR\_7\_CD": "\*  ", "ERR\_8\_CD": "\*  ", "ERR\_9\_CD": "\*  ", "ERR\_10\_CD": "\*  ", "ERR\_11\_CD": "\*  ", "ERR\_12\_CD": "\*  ", "ERR\_13\_CD": "\*  "}], "details": [{"KEY\_CHK\_DCN\_NBR": "171107A7856", "CLM\_CMPLTN\_DT": "12/31/9999", "DTL\_LINE\_NBR": "01", "MDFR\_1\_CD": "\* ", "MDFR\_2\_CD": "\* ", "MDFR\_3\_CD": "\* ", "PROC\_MDFR\_CD": "26", "PROC\_CD": "70486", "HCPCS\_MDFR\_CD": "\* ", "HCPCS\_CD": "\*", "BILLD\_CHRGD\_AMT": "221.20", "MBR\_CNTRCT\_TYPE\_CD": "A", "POT\_CD": "1", "TRTMNT\_TYPE\_CD": "05", "MBR\_CNTRCT\_CVRG\_CD": "X  ", "BNFT\_YEAR\_CNTRCT\_EFCTV\_DT": "10/1/2016", "BNFT\_YEAR\_CNTRCT\_REV\_DT": "10/1/2016", "MBR\_CNTRCT\_EFCTV\_DT": "2/1/2009", "MBR\_CNTRCT\_END\_DT": "6/1/2017", "SRVC\_FROM\_DT": "4/19/2017", "SRVC\_TO\_DT": "4/19/2017", "MEDCRB\_APRVD\_AMT": "46.94", "MEDCRB\_COINSRN\_AMT": "9.39", "MEDCRB\_DDCTBL\_AMT": "0.00", "MEDCRB\_PAID\_AMT": "36.80", "MBR\_CNTRCT\_CD": "Z155", "PN\_CD": "PB51", "TOS\_TYPE\_CD": "DXL", "CURNT\_QUE\_UNIT\_NBR": "U1"}, {"KEY\_CHK\_DCN\_NBR": "171107A7856", "CLM\_CMPLTN\_DT": "12/31/9999", "DTL\_LINE\_NBR": "02", "MDFR\_1\_CD": "\* ", "MDFR\_2\_CD": "\* ", "MDFR\_3\_CD": "\* ", "PROC\_MDFR\_CD": "26", "PROC\_CD": "76377", "HCPCS\_MDFR\_CD": "\* ", "HCPCS\_CD": "\*", "BILLD\_CHRGD\_AMT": "157.74", "MBR\_CNTRCT\_TYPE\_CD": "A", "POT\_CD": "1", "TRTMNT\_TYPE\_CD": "05", "MBR\_CNTRCT\_CVRG\_CD": "X  ", "BNFT\_YEAR\_CNTRCT\_EFCTV\_DT": "10/1/2016", "BNFT\_YEAR\_CNTRCT\_REV\_DT": "10/1/2016", "MBR\_CNTRCT\_EFCTV\_DT": "2/1/2009", "MBR\_CNTRCT\_END\_DT": "6/1/2017", "SRVC\_FROM\_DT": "4/19/2017", "SRVC\_TO\_DT": "4/19/2017", "MEDCRB\_APRVD\_AMT": "0.00", "MEDCRB\_COINSRN\_AMT": "0.00", "MEDCRB\_DDCTBL\_AMT": "0.00", "MEDCRB\_PAID\_AMT": "0.00", "MBR\_CNTRCT\_CD": "Z155", "PN\_CD": "PB51", "TOS\_TYPE\_CD": "DXL", "CURNT\_QUE\_UNIT\_NBR": "U1"}]}

----------------------------



- Finally, verify the log within PyCharm: ./logs/log.out

