Processed Transformation ETH

Uses EtherScan reference data for ETH data enrichment

Input:
- Data Content:EtherScan Reference Data
- Data Type: Parquet
- Data Source: Preprocessed Layer

Output:
- Data Content:EtherScan Enriched ETH Data
- Data Type: Parquet
- Data Destination: Processed Layer

** Notebook is simply for reference and only takes in a sample collection

```
from pyspark.sql.functions import udf
from Azure_configs import preprocessed_data_path, processed_data_path
from API_configs import etherscan_url, eth_api_key
import datetime
import time
import requests
```

```
today=datetime.date.today().strftime('%m-%d-%y')
```

```
def get_eth_balance(eth_address):
    url=etherscan_url
    api_key=eth_api_key
    param={'ETH_balance':{'module':'account',\
                          'action':'balance',\
                          'address':eth_address,\
                          'tag':'latest',\
                          'apikey':api_key}}
    limit_exceeded=True
    while limit_exceeded==True:
        response=requests.get(url,params=param['ETH_balance'])
        message=response.json()
        if message['result']=='Max rate limit reached':
            time.sleep(0.5)
            continue
        else:
            limit_exceeded=False
            return message['result']
    return
```

```
def process_eth_data(nft_name):

    EScan_parquet_path=f'{preprocessed_data_path}/{today}/EScan/NFT={nft_name}/'


    EScan_reference_DF=spark.read.parquet(EScan_parquet_path)


    eth_udf=udf(lambda x : get_eth_balance(x))

    eth_balance_df=EScan_reference_DF.withColumn('ETH_Balance',eth_udf(EScan_reference_DF['owner_address']))
    eth_balance_df.cache()
    eth_balance_df.show(10,truncate=False)
    eth_balance_df.write.mode('overwrite').parquet(f'{processed_data_path}{today}/ETH_Balance/NFT={nft_name}/')

    return
```

```
process_eth_data('CryptoPunks')
```

```
+------------------------------------------+--------------------+
|owner_address                             |ETH_Balance         |
+------------------------------------------+--------------------+
|0x040da2c464933005b6d1ffeccec7fb4025dc9ddb|348286866000000000  |
|0x04ae2f0bda04f1405991d91c2e8420d6148369ea|210106156418171898  |
|0x06e63138f3241a420829bc125e6cb6bebf88c2c2|584320090945613308  |
|0x0a8f4037729accb854da856431a87b8a264d8c40|0                   |
|0x1251122f1d77fa46e1e576c4fd6dd56ab17812ff|0                   |
|0x1db12c2a7c803567c9fd59a202504edf0e56ca78|11905000000000000000|
|0x22eab1c78521596f9d6d73dd878778009c39317c|10308627837000000000|
|0x266892ed0d40ea5c37f3e0239537999c13468311|0                   |
|0x2e675eeae4747c248bfddbafaa3a8a2fdddaa44b|551848095532668789  |
|0x34669322bdfca9e801ca334e7b0e6d69d1f87137|270971634624580300879|
+------------------------------------------+--------------------+
only showing top 10 rows
```