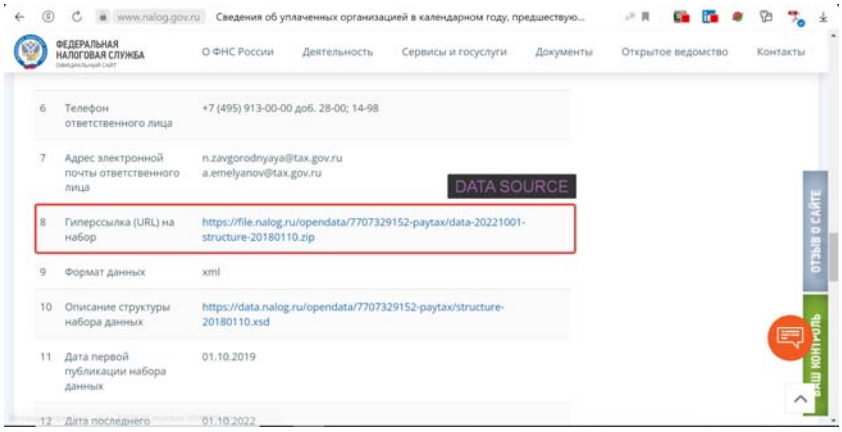
FEDERAL TAX SERVICE

Data source on web site https://www.nalog.gov.ru/opendata/7707329152-paytax/



the original code https://github.com/Championloo/FNS-parser

My contribution to the code is implementing the different approach to work with processing (pandas), and SQL database connection (pyodbc).

XML PART

In [ ]:
```
#import libraries
import os
import xmltodict
import json
import pandas as pd
from tqdm import tqdm
```

In [ ]:
```
#path to the catalog with xml files
path = r'C:\Users\Asus'
```

In [ ]:
```
%%time

df = pd.DataFrame({'inn':[], 'org_name':[], 'tax_name':[], 'tax_amount':[], 'date':[
d={}

for file in tqdm(os.listdir(path)):
    with open(f'{path}/{file}', 'r', encoding = 'utf-8') as f:
        a = f.read()
        xml = xmltodict.parse(a)
        xml_dict = json.loads(json.dumps(xml))['Файл']['Документ']
        try:
            for i in xml_dict:
                inn = i['СведНП']['@ИННЮЛ'].replace("\'", "'")
                org_name = i['СведНП']['@НаимОрг']
                date = i['@ДатаСост']
                df['inn'] = inn
                df['org_name'] = org_name
                df['date'] = date
                try:
                    for j in i['СвУплСумНал']:
                        tax_name = j['@НаимНалог']
                        tax_amount = j['@СумУплНал']
                        df['tax_name'] = tax_name
```

```
                        df['tax_amount'] = tax_amount
                        df = df.append(pd.DataFrame([d]))

                except:
                    tax_name = i['СвУплСумНал']['@НаимНалог']
                    tax_amount = i['СвУплСумНал']['@СумУплНал']
                    df['tax_name'] = tax_name
                    df['tax_amount'] = tax_amount
                    df = df.append(pd.DataFrame([d]))

        except:
            try:
                inn = xml_dict['СведНП']['@ИННЮЛ'].replace("\'", "'")
                org_name = xml_dict['СведНП']['@НаимОрг']
                date = xml_dict['@ДатаСост']
                df['inn'] = inn
                df['org_name'] = org_name
                df['date'] = date

                try:
                    for j in xml_dict['СвУплСумНал']:
                        tax_name = j['@НаимНалог']
                        tax_amount = j['@СумУплНал']
                        df['tax_name'] = tax_name
                        df['tax_amount'] = tax_amount
                        df = df.append(pd.DataFrame([d]))
                except:
                    tax_name = xml_dict['СвУплСумНал']['@НаимНалог']
                    tax_amount = xml_dict['СвУплСумНал']['@СумУплНал']
                    df['tax_name'] = tax_name
                    df['tax_amount'] = tax_amount
                    df = df.append(pd.DataFrame([d]))

            except:
                print(file)
                print(inn)
                print(xml_dict[i])

df['date'] = pd.to_datetime(df['date'], format='%d.%m.%Y')
df['tax_amount'] = df['tax_amount'].astype(float)
```

SQL PART

In [ ]:
```
server = 'server'
```

In [ ]:
```
database = 'database'
```

In [ ]:
```
schema = 'schema'
```

In [ ]:
```
table_name = 'table_name' #target table name
```

In [ ]:
```
#make a connect to sql database
def establish_connection():
    server = server
    database=database
    connection_string = 'DRIVER={SQL Server};SERVER='+server+';DATABASE='+database+'
```

```
    connection = pyodbc.connect(connection_string)

    return connection
```

In [ ]:
```
connection = establish_connection()
```

In [ ]:
```
DB_TABLE = True #overwright table - on
```

In [ ]:
```
target_table_name = f'{database}.{schema}.{table_name}'
```

In [ ]:
```
with connection:
    cursor = connection.cursor()
    cursor.execute(f'DELETE FROM {schema}.{table_name}')
    try:
        lt = list(df.itertuples(index=False, name=None)) #list tuples

        cursor.executemany("""
        INSERT INTO [{}].[{}].[{}]
        (inn, org_name, tax_name, tax_amount, date_add)
        VALUES (?,?,?,?,?)
        """).format(database, schema, table_name), list_tuples)

        connection.commit
        print('SUCCESS')
    except:
        print("UNSUCCESS, error {}, description {}".format(sys.exc_info()[0], sys.ex
        connection.rollback()
connection.close()
```