A common KPI for any company is it's market share. The basic formula for the calculation is company sales / total market sales. One neat way to get the denominator automactically (ie total market sales) is to use the Australian Bureau of Statistics API. The below script as an example gets the total sales for the retail supermarket Australia wide.

https://www.abs.gov.au/ausstats/abs@.nsf/Lookup/1407.0.55.002Main+Features3User+Guide (https://www.abs.gov.au/ausstats/abs@.nsf/Lookup/1407.0.55.002Main+Features3User+Guide)

In [8]:

```python
# First we will use some open source libraries
from datetime import datetime
import requests
import pandas as pd
import pypyodbc
import numpy as np
import xlsxwriter

# Get start and end period for API query using the datetime library
now = datetime.now()
end = str(now.year) + "-" + str(now.month).zfill(2)
start = str(now.year - 3) + "-" + str('01')

"Start: " + start,"End: " + end
```

Out[8]:

```
('Start: 2018-01', 'End: 2021-05')
```

Using the date time library, we can get our start and last date to insert into the url. This could also be done manually, or via the input() function built into python, but for my purposes I've created a variable which gets todays year and month as the end, and the same month three years ago for the start.

In [11]:

```
# Using the URL from the ABS website to make the request. Note that parameters are fed into the URL.

url = 'http://stat.data.abs.gov.au/sdmx-json/data/RT/0.2.01.10.M/all?detail=Full&dimensionAtObservation=AllDimensions&startPeriod=' + start + '&endPeriod=' + end
data = requests.get(url).json()

data
```

Out[11]:

```
{'header': {'id': '530bf19f-33a1-4a76-8c32-3d762013191e',
  'test': False,
  'prepared': '2021-05-13T12:16:13.0868292Z',
  'sender': {'id': 'ABS', 'name': 'Australian Bureau of Statistics'},
  'links': [{'href': 'http://stat.data.abs.gov.au:80/sdmx-json/data/RT/0.
2.01.10.M/all?detail=Full&dimensionAtObservation=AllDimensions&startPeriod
=2018-01&endPeriod=2021-05',
    'rel': 'request'}]},
 'dataSets': [{'action': 'Information',
   'observations': {'0:0:0:0:0:0': [9001.13, 0, None],
    '0:0:0:0:0:1': [8252.32, 0, None],
    '0:0:0:0:0:2': [9348.06, 0, None],
    '0:0:0:0:0:3': [8685.34, 0, None],
    '0:0:0:0:0:4': [8950.29, 0, None],
    '0:0:0:0:0:5': [8625.18, 0, None],
    '0:0:0:0:0:6': [8918.78, 0, None],
    '0:0:0:0:0:7': [9115.1, 0, None],
    '0:0:0:0:0:8': [8968.78, 0, None],
    '0:0:0:0:0:9': [9355.1, 0, None],
    '0:0:0:0:0:10': [9390.87, 0, None],
    '0:0:0:0:0:11': [10536.39, 0, None],
    '0:0:0:0:0:12': [9494.2, 0, None],
    '0:0:0:0:0:13': [8703.54, 0, None],
    '0:0:0:0:0:14': [9608.02, 0, None],
    '0:0:0:0:0:15': [9163.69, 0, None],
    '0:0:0:0:0:16': [9303.5, 0, None],
    '0:0:0:0:0:17': [8963.13, 0, None],
    '0:0:0:0:0:18': [9270.19, 0, None],
    '0:0:0:0:0:19': [9536.82, 0, None],
    '0:0:0:0:0:20': [9217.06, 0, None],
    '0:0:0:0:0:21': [9740.05, 0, None],
    '0:0:0:0:0:22': [9741.44, 0, None],
    '0:0:0:0:0:23': [10751.57, 0, None],
    '0:0:0:0:0:24': [9784.65, 0, None],
    '0:0:0:0:0:25': [9397.76, 0, None],
    '0:0:0:0:0:26': [11994.51, 0, None],
    '0:0:0:0:0:27': [9704.5, 0, None],
    '0:0:0:0:0:28': [10281.94, 0, None],
    '0:0:0:0:0:29': [9855.03, 0, None],
    '0:0:0:0:0:30': [10525.09, 0, None],
    '0:0:0:0:0:31': [10607.61, 0, None],
    '0:0:0:0:0:32': [10237.96, 0, None],
    '0:0:0:0:0:33': [10770.24, 0, None],
    '0:0:0:0:0:34': [10525.25, 0, None],
    '0:0:0:0:0:35': [11589.21, 0, None],
    '0:0:0:0:0:36': [10638.71, 0, None],
    '0:0:0:0:0:37': [9460.46, 0, None],
    '0:0:0:0:0:38': [10393.05, 0, None]}}],
 'structure': {'links': [{'href': 'http://stat.data.abs.gov.au/sdmx-json/d
ataflow/RT/all',
    'rel': 'dataflow'}],
  'name': 'Retail Trade',
  'description': 'Retail Trade',
  'dimensions': {'observation': [{'keyPosition': 0,
     'id': 'ASGC_2010',
     'name': 'Region',
     'values': [{'id': '0', 'name': 'Australia'}]},
    {'keyPosition': 1,
     'id': 'DT',
        'name': 'Data Type',
        'values': [{'id': '2', 'name': 'Current Prices ($ Million)'}]},
       {'keyPosition': 2,
        'id': 'IND_R',
        'name': 'Retail Industry',
        'values': [{'id': '01', 'name': 'Supermarket and grocery stores'}]},
       {'keyPosition': 3,
        'id': 'TSEST',
        'name': 'Adjustment Type',
        'values': [{'id': '10', 'name': 'Original'}]},
       {'keyPosition': 4,
        'id': 'FREQUENCY',
        'name': 'Frequency',
        'values': [{'id': 'M', 'name': 'Monthly'}],
        'role': 'FREQ'},
       {'id': 'TIME_PERIOD',
        'name': 'Time',
        'values': [{'id': '2018-01', 'name': 'Jan-2018'},
         {'id': '2018-02', 'name': 'Feb-2018'},
         {'id': '2018-03', 'name': 'Mar-2018'},
         {'id': '2018-04', 'name': 'Apr-2018'},
         {'id': '2018-05', 'name': 'May-2018'},
         {'id': '2018-06', 'name': 'Jun-2018'},
         {'id': '2018-07', 'name': 'Jul-2018'},
         {'id': '2018-08', 'name': 'Aug-2018'},
         {'id': '2018-09', 'name': 'Sep-2018'},
         {'id': '2018-10', 'name': 'Oct-2018'},
         {'id': '2018-11', 'name': 'Nov-2018'},
         {'id': '2018-12', 'name': 'Dec-2018'},
         {'id': '2019-01', 'name': 'Jan-2019'},
         {'id': '2019-02', 'name': 'Feb-2019'},
         {'id': '2019-03', 'name': 'Mar-2019'},
         {'id': '2019-04', 'name': 'Apr-2019'},
         {'id': '2019-05', 'name': 'May-2019'},
         {'id': '2019-06', 'name': 'Jun-2019'},
         {'id': '2019-07', 'name': 'Jul-2019'},
         {'id': '2019-08', 'name': 'Aug-2019'},
         {'id': '2019-09', 'name': 'Sep-2019'},
         {'id': '2019-10', 'name': 'Oct-2019'},
         {'id': '2019-11', 'name': 'Nov-2019'},
         {'id': '2019-12', 'name': 'Dec-2019'},
         {'id': '2020-01', 'name': 'Jan-2020'},
         {'id': '2020-02', 'name': 'Feb-2020'},
         {'id': '2020-03', 'name': 'Mar-2020'},
         {'id': '2020-04', 'name': 'Apr-2020'},
         {'id': '2020-05', 'name': 'May-2020'},
         {'id': '2020-06', 'name': 'Jun-2020'},
         {'id': '2020-07', 'name': 'Jul-2020'},
         {'id': '2020-08', 'name': 'Aug-2020'},
         {'id': '2020-09', 'name': 'Sep-2020'},
         {'id': '2020-10', 'name': 'Oct-2020'},
         {'id': '2020-11', 'name': 'Nov-2020'},
         {'id': '2020-12', 'name': 'Dec-2020'},
         {'id': '2021-01', 'name': 'Jan-2021'},
         {'id': '2021-02', 'name': 'Feb-2021'},
         {'id': '2021-03', 'name': 'Mar-2021'}],
        'role': 'TIME_PERIOD'}]},
     'attributes': {'dataSet': [],
      'series': [],
      'observation': [{'id': 'TIME_FORMAT',
        'name': 'Time Format',
      'values': [{'id': 'P1M', 'name': 'Monthly'}]},
     {'id': 'OBS_STATUS', 'name': 'Observation Status', 'values': []}]},
   'annotations': [{'title': 'Statistical usage warning',
     'uri': '',
     'text': 'ABS.Stat beta is continuing to be developed.  Data will be up
dated as soon as possible following its 11:30 am release on the ABS websit
e.'}]}}
```

As you can see, we get a JSON file with our data set. Its a bit messy, but we can iterate through the JSON file with some very simple code.

In [13]:

```
# This picks up the data
ds = data['dataSets'][0]['observations']
df = pd.DataFrame.from_dict(ds, orient = 'index').reset_index()

# This picks up the month periods
ds1 = data['structure']['dimensions']['observation'][5]['values']
df1 = pd.DataFrame(ds1).reset_index()

df.head(5), df1.head(5)
```

Out[13]:

```
(         index        0  1     2
 0  0:0:0:0:0:0  9001.13  0  None
 1  0:0:0:0:0:1  8252.32  0  None
 2  0:0:0:0:0:2  9348.06  0  None
 3  0:0:0:0:0:3  8685.34  0  None
 4  0:0:0:0:0:4  8950.29  0  None,
    index       id      name
 0      0  2018-01  Jan-2018
 1      1  2018-02  Feb-2018
 2      2  2018-03  Mar-2018
 3      3  2018-04  Apr-2018
 4      4  2018-05  May-2018)
```

Now that we have our data frames, we can now combine them using the pandas funcion concat on the index.

In [14]:

```
combined = pd.concat([df, df1.reindex(df1.index)], axis=1).drop(['index',1,2],axis=1)

combined.head(10)
```

Out[14]:

	0	id	name
0	9001.13	2018-01	Jan-2018
1	8252.32	2018-02	Feb-2018
2	9348.06	2018-03	Mar-2018
3	8685.34	2018-04	Apr-2018
4	8950.29	2018-05	May-2018
5	8625.18	2018-06	Jun-2018
6	8918.78	2018-07	Jul-2018
7	9115.10	2018-08	Aug-2018
8	8968.78	2018-09	Sep-2018
9	9355.10	2018-10	Oct-2018