PyDaxExtract

Release 0.2.1

Doug Shawhan

May 28, 2021

CONTENTS:

Extract DAX expressions from Power BI templates. Extract table relationships and *m* and *DAX* expressions from a *Power BI* template file.

Power BI files in the *pbix* and *pbit* formats are basically zip archives containing other compressed data.

The *DataModel* file in a *pbix* file contains all the *DAX* expressions created when processing data. All data is saved in the *Xpress9* format, which is a proprietary compresson method optimized to dump memory to disk and vice-versa, with encryption and all kinds of other wonderful features which will break your heart if you try to get a peek inside.

Fortunately, if one saves a *Power BI* workbook as a template, the table relationships, *m* expressions and *DAX* expressions are now saved in the *DataModelSchema* object, which is unencrypted and requires only a bit of fiddling to remove.

This module, and command-line script are intended to help with that fiddling, and to aid users in serializing work done in an otherwise fairly opaque binary format. Here's hoping it's useful to you.

At this point, there appears to be no way to automate exporting *pbix* files as *pbit*, so you'll have to do that the usual way.

class dax_extract.**DaxExtract**

Command line tool to extract table relationships, `m` expressions and `DAX` expressions from Power BI templates.

Usage:

```
usage: daxextract.py [-h] [--dump-json] [--dump-expressions] [--write-dax-csv]
              [--write-powerquery-csv] [--write-relationships-csv]
              pbit_path

Extract PowerQuery (m) / DAX expressions from Power BI template file.

positional arguments:
  pbit_path             Path to .pbit file.

optional arguments:
  -h, --help            show this help message and exit
  --dump-json           Write full dump of DataModelSchema to stdout in json
                        format.
  --dump-expressions    Write DAX and PowerQuery (m) expressions to stdout in
                        json format.
  --write-dax-csv       write csv file containing DAX measures and metadata in
                        csv format.
  --write-powerquery-csv
                        write csv file containing PowerQuery expressions and
                        metadata in csv format.
  --write-relationships-csv
                        write csv file containing table relationships in csv
                        format.
```

dump_expressions()

Write nicely-formatted `DAX` and `PowerQuery (m)` expressions and metadata to STDOUT

dump_json()

Dump well-formatted JSON file to STDOUT

extract_data_model_schema()

Extract `DataModelSchema` from `.pbit` archive.

Returns `dict` object of `DataModelSchema` data.

Return type data (dict)

extract_dax()
Extract DAX formulas from DataModelSchema.

Returns dax formulas and metadata.

Return type dax (dict)

extract_powerquery_expressions()
Extract PowerQuery (m) formulas from DataModelSchema.

Returns PowerQuery formulas and metadata.

Return type pqx (dict)

extract_relationships()
Extract relationships from DataModelSchema.

Returns PowerQuery formulas and metadata.

Return type pqx (dict)

write_csv(*data_type*, *data*)
Write data to file in .csv format.

write_dax_csv()
Write DAX measures and metadata to file in .csv format.

write_powerquery_csv()
Write PowerQuery (m) expressions and metadata to file in .csv format.

write_relationships_csv()
Write table relationships to file in .csv format.

dax_extract.**read_data_model_schema**(*pbit_path*)
Extract complete DataModelSchema from .pbit archive provided in pathlib.Path object.

Parameters **pbit_path** (*pathlib.Path*) – Path to .pbit archive.

Returns DataModelSchema data as dict.

Return type data (dict)

Example:

```
>> from pathlib import Path
>> from dax_extract import read_data_model_schema
>> pbit_path = Path("/path/to/my_awesome.pbit")
>> data = read_data_model_schema(pbit_path)
```

Run dax_extract module with command line arguments

scripts.daxextract.**main()**
Run DaxExtract to file or STDOUT.

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

s

INDEX