
popEye Documentation

Tal Shany

Feb 17, 2019

Installation Configuration

1	About popEye	1
1.1	installation	1
1.2	configuration	1
1.3	Mapping module	2
1.4	Loading module	2
1.5	Mapping and Loading samples	2
1.6	Adding Business logic into loading	2
2	Indices and tables	3

CHAPTER 1

About popEye

popEye is an IT integration tool. It can map diverse SQL DBs and No SQL DB, extract data from one source to another and maintain business logic in an easy SQL/python format.

popEye's main goals are creating fast integration cycles and enabling to implement new requirements in a fast manner. It also provides a platform that enables to test and update business logic using the best of SQL on one hand and python on the other.

We believe that integration can be fast and much more simple by using simple scripting language by implementing 2 major concepts :

- Fast design - the mapping module enables to create / maintain data model from scratch or by using existing data model. For example : to create full DWH implementation in Vertica based on mongo - all we have to do is create DWH model (entities, fields) using simple JSON format. popEye will manage all data type structure and internal Vertica objects based on Mongo structure
- Fast Extract and loading - Loader module enables to full load / merge / increment methods for loading data on a scheduled process based on JSON defined mapping structure

This documentation is the first version of popEye, we do look for your help and we will provide our “wish-list” .

PopEye hopes to extend and be one of the major open-source integration platforms. Come and join us

1.1 installation

install instruction

```
for i in range(10):  
    print(i)
```

1.2 configuration

config instructions

1.3 Mapping module

Mapping module is use to create target structure based on source systems strucute or by define strucure in a Json format.

Example of usage assuming we are using oracle as source and Sql-server as target :

- **create target strucure based on source query** query is : “Select field1 As Yoyo, field2 as bobo ... From oracle.table1” result

1.3.1 Jason mapping params

sample json mapping

1.3.2 Mapping samples

sample mapping

1.4 Loading module

loading module

1.4.1 Jason loading params

sample json loading

1.4.2 Loading samples

sample loading

1.5 Mapping and Loading samples

Config Json in certain directory

1.6 Adding Business logic into loading

Add sample here

CHAPTER 2

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)
- [license](#)