mssql-runner

Release 0.1.0

Equinox Fitness

Apr 08, 2020

CONTENTS

1 Installation 3

2 Quickstart 5

3 Development 7

3.1 Getting Started 7

3.2 Testing 7

4 Contributing 9

MSSQL Runner provides a way of running MSSQL script with a set of parameters for ETL usage

CHAPTER ONE

INSTALLATION

MSSQL Runner requires Python 3.6+

```
python3 -m venv <virtual env name>
source <virtual env name>/bin/activate
pip install mssql-runner
```

CHAPTER TWO

QUICKSTART

There are three types of substitution parameters that can be used through mssql runner. The first is a set of standard etl params for ease of use:

- -f, from_date, default 1776-07-24
- -t,to_date, default 9999-12-31
- -b, batch_no, default is -1

The second option is for arbitrary variable expansion. This is passed in the following format because rundeck commands don't like json quotes:

- -p, example: -p "param1-val1, param2-val2"

The final option is to use the batchy integration

- -wf, batchy_job, this will substitue parameters from a batchy workflow, this should be a fully qualified batchy job name of the format wf.job, if no job is specified it will assume global

You also have the option to choose which config provider to use.

- -conf, config, default core

Datacoco Core - This is default config provider which is looking for a file called etl.cfg in your project root.

etl.cfg folder structure

` [sample] db_name=local user=user server=server password=XXXX port=1433 type=mssql `

Secret Manager - Using this option assumes that you already have secret manager setup in your aws account and as aws key and secret is configured in your environment for you to connect to aws.

Here's example how to use secret manager config option * -conf secret_manager –secret_project_name <your_project> –secret_team <your_team>

Here is a sample SQL Script. If run in SQL workbench you will be prompted for values for var1 and var2.

```
drop table if exists  zzztemp;

create table zzztemp (
dt timestamp,
var varchar,
from_date timestamp,
batch_no integer
);

insert into zzztemp
values (getdate(), '$[?var1]', '$[?from_date]', '$[?batch_no]');
```

(continues on next page)

(continued from previous page)

```
insert into zzztemp
values (getdate(), '$[?var2]', '$[?from_date]', '$[?batch_no]');

select * from zzztemp;
```

In mssql runner you would use the following params to substitute that value. It is assumed these params would be dynamically substituted by the calling script or informatica process:

```
python -m mssql_runner.module -s "sample/mssql_runner_test.sql" -p "var1-cat, var2-dog" -b '9999'
```

Assuming you had workflow config in batchy under wf3, you could also use this script:

```
python -m mssql_runner.module -s sample/mssql_runner_test.sql -wf wf3
```

CHAPTER

THREE

DEVELOPMENT

3.1 Getting Started

It is recommended to use the steps below to set up a virtual environment for development:

```
python3 -m venv <virtual env name>
source <virtual env name>/bin/activate
pip install -r requirements.txt
```

3.2 Testing

```
pip install -r requirements-dev.txt
```

To run the testing suite, simply run the command: `tox` or `python -m unittest discover tests`

CHAPTER FOUR

CONTRIBUTING

Contributions to mssql_runner are welcome!

Please reference guidelines to help with setting up your development environment here.