

MyTool

データベースに接続しメタ情報からAsakusa用DMDL、embulkスクリプトを生成。
以下手順ではテスト実行の手順はDockerを利用した手順を記載しています。

requirements

- Java 1.8
- oracle instantclient_sqlplus 9.0.3
- embulk 0.9.22
- embulk plugin embulk-input-oracle (0.10.1)
- embulk plugin embulk-output-oracle (0.8.6)
- docker 19.03.5
- OracleContainer 11.2.0.2

準備

ダウンロード

```
$ git clone git@github.com:SugioNakazawa/MyTools.git
```

ビルド&配置

DBが用意されていない場合はJUnitでエラーになりますので最初はテストをスキップしてビルドします。

```
$ cd MyTools
$ ./gradlew build
# 任意の作業場所へ
$ cd ~/work
$ tar xvf ~/github/MyTools/build/distributions/MyTools-1.0.tar
```

Databse環境

Docker Container起動。

```
$ ./MyTools-1.0/init/startOracleContainer.sh
```

実行内容は以下と同じ。

```
$ docker run --rm --name docker_oracle_11202 --shm-size=1g \  
-p 1521:1521 -p 8080:8080 -e ORACLE_PWD=password oracle/database:11.2.0.2-xe
```

以下のコメントが出力されたらDB起動完了です。

```
#####  
DATABASE IS READY TO USE!  
#####
```

- オプション `--rm` によりcontainer停止時に削除されます。
止めるときは別のターミナルから

```
$ docker container stop docker_oracle_11202
```

この例ではcontainerはフォアグラウンドにて実行しますので、以降は別のターミナルにてworkディレクトで作業をおこないます。

スキーマ準備

usr1ユーザ、ディレクトリの作成、init.dmpをcontainerへコピーしimpdpの実行をします。

```
$ ./MyTools-1.0/init/prepareContainerDb.sh
```

ユーザーとテーブルの確認

```
$ sqlplus usr1/pass1@localhost:1521/XE
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 19 14:43:21 2019
```

```
Copyright (c) 1982, 2016, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
```

```
SQL> select table_name from user_tables;
```

```
TABLE_NAME
-----
HOGE_TBL

SQL>
```

CreateDataDef

OracleDBに接続し、指定したテーブルのDMDL、embulk用スクリプトを生成（標準出力に出力）。

接続DBはローカルのOracleDB11.2.0.2を想定。

```
CreateDataDef type tableName
```

処理概要

引数

- 引数1 :type
 - dmdl:AsakusaFW用DMDL
 - tocsv:embulk用 DB -> csv
 - todb:embulk用 csv -> db
- 引数2 :tableName
 - 対象にするテーブル名。DBのに存在すること。

システム変数

- DB_HOST ホスト名（IP）。初期値=localhost
- DB_PORT ポート。初期値=1521
- DB_SID DB_SID。初期値=xe
- DB_USR ユーザ名。初期値=usr1
- DB_PASS パスワード。初期値=pass1
- DMDL_NAME_SPACE 出力するDMDLのnamespace。初期値=db

カラム型変換対象

Oracle	Asakusa DMDL	embulk
VARCHAR2	TEXT	string
CHAR	TEXT	string
CLOB	TEXT	string
NUMBER	DECIMAL	double
DATE	DATE	timestamp, format: '%Y-%m-%d'
TIMESTAMP(6)	DATETIME	timestamp, format: '%Y-%m-%d %k:%M:%S'
上記以外の型はスクリプトに出力されません。		

DMDLの生成

実行例

```
$ java -cp MyTools-1.0/MyTools-1.0.jar com.hoge.CreateDataDef dmdl HOGE_TBL
```

出力結果

```
"HOGE_TBL"  
@namespace(value = db)  
@windgate.jdbc.table(name = "HOGE_TBL")  
@directio.csv
```

```
hoge_tbl = {  
  "CHAR_COLUMN"  
  @windgate.jdbc.column(name = "CHAR_COLUMN")  
  char_column : TEXT;  
  
  "VARCHAR2_COLUMN"  
  @windgate.jdbc.column(name = "VARCHAR2_COLUMN")  
  varchar2_column : TEXT;  
  
  "CLOB_COLUMN"  
  @windgate.jdbc.column(name = "CLOB_COLUMN")  
  clob_column : TEXT;  
  
  "NUMBER_10_3_COLUMN"  
  @windgate.jdbc.column(name = "NUMBER_10_3_COLUMN")  
  number_10_3_column : DECIMAL;  
  
  "NUMBER_8_COLUMN"  
  @windgate.jdbc.column(name = "NUMBER_8_COLUMN")  
  number_8_column : DECIMAL;  
  
  "DATE_COLUMN"  
  @windgate.jdbc.column(name = "DATE_COLUMN")  
  date_column : DATE;  
  
};
```

embulkスクリプト

準備

embulkを実行するためデプロイディレクトリから共通定義ファイルとJDBCドライバを実行ディレクトリにコピーします。

```
## 共通ファイル  
$ cp MyTools-1.0/bin/_myenv.yml.liquid .  
## ドライバ  
$ cp MyTools-1.0/lib/ojdbc8.jar .  
## 格納先ディレクトリを作成  
$ mkdir -p tocsv/hoge_tbl  
$ mkdir -p todb/hoge_tbl
```

_myenv.yml.liquid ファイル：DB接続などの共通情報を記載しています。環境に合わせて適時修正してください。

スクリプト作成

モジュールを実行してDBスキーマからembulk定義を作成。

```
$ java -cp MyTools-1.0/MyTools-1.0.jar com.hoge.CreateDataDef tocsv hoge_tbl \  
> hoge_tbl_tocsv.yml.liquid  
$ java -cp MyTools-1.0/MyTools-1.0.jar com.hoge.CreateDataDef todb hoge_tbl \  
> hoge_tbl_todb.yml.liquid
```

CSVダウンロード実行

```
## プレビューで確認  
$ embulk preview hoge_tbl_tocsv.yml.liquid  
## 実行  
$ embulk run hoge_tbl_tocsv.yml.liquid
```

```
## 確認  
$ tree tocsv/  
tocsv/  
└─ hoge_tbl  
    ├── hoge_tbl000.00.csv  
    ├── hoge_tbl001.00.csv  
    ├── hoge_tbl002.00.csv  
    └─ hoge_tbl003.00.csv  
  
1 directory, 4 files
```

DBへのアップロード実行

```
## データ準備。ダウンロードしたものを使用。  
$ cp tocsv/hoge_tbl/* todb/hoge_tbl/  
## プレビューで確認  
$ embulk preview hoge_tbl_todb.yml.liquid  
## 実行  
$ embulk run hoge_tbl_todb.yml.liquid
```

DBにデータが登録されていることを確認します。LONG_COLUMNはLONG型のため無視されるのでnullになります。