RobotResults2DB

v. 1.2.0

Tran Duy Ngoan

04.07.2022

Contents

Chapter 1

Introduction

RobotResults2DB

!!! add introduction !!

Chapter 2

Description

The RobotResults2DB:
TO BE CONTINUED

Chapter 3

CDataBase.py

3.1 Class: CDataBase

```
RobotResults2DB.CDataBase
```

CDataBase class play a role as mysqlclient and provide methods to interact with TestResultWebApp's database. Method: connect ----------

Connect to the database with provided authentication and db info.

Args: host: URL which is hosted the TestResultWebApp's database.

user : user name for database authentication.

passwd : user password for database authentication.

database : database name.

charset (optional): the connection character set.

use_unicode (optional): If True, CHAR and VARCHAR and TEXT columns are returned as Unicode strings, using the configured character set.

Returns: None.

3.1.1 Method: disconnect

Disconnect from TestResultWebApp's database. Method: cleanAllTables ----------

Delete all table data. Please be careful before calling this method. Method: sCreateNewTestResult ----------

Creates a new test result in tbl_result. This is the main table which is linked to all other data by means of test_result_id.

Args: _tbl_prj_project : project information.

_tbl_prj_variant : variant information.

_tbl_prj_branch : branch information.

_tbl_test_result_id : UUID of test result.

_tbl_result_interpretation : result interpretation.

_tbl_result_time_start : test result start time.

_tbl_result_time_end : test result end time.

_tbl_result_version_sw_target : software version information.

_tbl_result_version_sw_test : test version information.

_tbl_result_version_target : hardware version information.

_tbl_result_jenkinsurl : jenkinsurl in case test result is executed by jenkins.

_tbl_result_reporting_qualitygate : qualitygate information for reporting.

Returns: _tbl_test_result_id: test_result_id.

3.1.2 Method: nCreateNewFile

Create new file entry in tbl_file table.

Args: _tbl_file_name : file name information.

_tbl_file_tester_account : tester account information.

_tbl_file_tester_machine : test machine information.

_tbl_file_time_start : test file start time.

_tbl_file_time_end : test file end time.

_tbl_test_result_id : UUID of test result for linking to tbl_result table.

_tbl_file_origin : origin (test framework) of test file. Deafult is "ROBFW"

Returns: iInsertedID: ID of new entry.

3.1.3 Method: vCreateNewHeader

Create a new header entry in tbl_file_header table which is linked with the file.

Args: _tbl_file_id : file ID information.

_tbl_header_testtoolconfiguration_testtoolname : test tool name.

_tbl_header_testtoolconfiguration_testtoolversionstring : test tool version.

_tbl_header_testtoolconfiguration_projectname : project name.

_tbl_header_testtoolconfiguration_logfileencoding : encoding of logfile.

_tbl_header_testtoolconfiguration_pythonversion : Python version info.

_tbl_header_testtoolconfiguration_testfile : test file name.

_tbl_header_testtoolconfiguration_logfilepath : path to log file.

_tbl_header_testtoolconfiguration_logfilemode : mode of log file.

_tbl_header_testtoolconfiguration_ctrlfilepath : path to control file.

_tbl_header_testtoolconfiguration_configfile : path to configuration file.

_tbl_header_testtoolconfiguration_confname : configuration name.

_tbl_header_testfileheader_author : file author.

_tbl_header_testfileheader_project : project information.

_tbl_header_testfileheader_testfiledate : file creation date.

_tbl_header_testfileheader_version_major : file major version.

_tbl_header_testfileheader_version_minor : file minor version.

_tbl_header_testfileheader_version_patch : file patch version.

_tbl_header_testfileheader_keyword : file keyword.

_tbl_header_testfileheader_shortdescription : file short description.

_tbl_header_testexecution_useraccount : tester account who run the execution.

_tbl_header_testexecution_computername : machine name which is executed on.

_tbl_header_testrequirements_documentmanagement : requirement management information.
_tbl_header_testrequirements_testenvironment : requirement environment information.
_tbl_header_testbenchconfig_name : testbench configuration name.
_tbl_header_testbenchconfig_data : testbench configuration data.
_tbl_header_preprocessor_filter : preprocessor filter information.
_tbl_header_preprocessor_parameters : preprocessor parameters definition.

Returns: None.

3.1.4 Method: nCreateNewSingleTestCase

Create single testcase entry in tbl_case table immediately.

Args: _tbl_case_name : test case name.
_tbl_case_issue : test case issue ID.
_tbl_case_tcid : test case ID (used for testmanagement tool).
_tbl_case_fid : test case requirement (function) ID.
_tbl_case_testnumber : order of test case in file.
_tbl_case_repeatcount : test case repeatition count.
_tbl_case_component : component which test case is belong to.
_tbl_case_time_start : test case start time.
_tbl_case_result_main : test case main result.
_tbl_case_result_state : test case completion state.
_tbl_case_result_return : test case result code (as integer).
_tbl_case_counter_resets : counter of target reset within test case execution.
_tbl_case_lastlog : traceback information when test case is failed.
_tbl_test_result_id : UUID of test result for linking to file in tbl_result table.
_tbl_file_id : test file ID for linking to file in tbl_file table.

Returns: iInsertedID: ID of new entry.

3.1.5 Method: nCreateNewTestCase

Create bulk of test case entries: new test case are buffered and inserted as bulk.
Once __NUM_BUFFERD_ELEMENTS_FOR_EXECUTEMANY is reached, the creation query is executed.

Args: _tbl_case_name : test case name.
_tbl_case_issue : test case issue ID.
_tbl_case_tcid : test case ID (used for testmanagement tool).
_tbl_case_fid : test case requirement (function) ID.
_tbl_case_testnumber : order of test case in file.
_tbl_case_repeatcount : test case repeatition count.
_tbl_case_component : component which test case is belong to.
_tbl_case_time_start : test case start time.
_tbl_case_result_main : test case main result.
_tbl_case_result_state : test case completion state.
_tbl_case_result_return : test case result code (as integer).

_tbl_case_counter_resets : counter of target reset within test case execution.
_tbl_case_lastlog : traceback information when test case is failed.
_tbl_test_result_id : UUID of test result for linking to file in tbl_result table.
_tbl_file_id : test file ID for linking to file in tbl_file table.

Returns: None.

3.1.6 Method: vCreateTags

Create tag entries.

Args: _tbl_test_result_id : UUID of test result.
_tbl_usr_result_tags : user tags information.

Returns: None.

3.1.7 Method: vSetCategory

Create category entry.

Args: _tbl_test_result_id : UUID of test result.
tbl_result_category_main : category information.

Returns: None.

3.1.8 Method: vUpdateStartEndTime

Create start-end time entry.

Args: _tbl_test_result_id : UUID of test result.
_tbl_result_time_start : result start time.
_tbl_result_time_end : result end time.

Returns: None.

3.1.9 Method: arGetCategories

Get existing categories.

Returns: arCategories : list of exsiting categories.

3.1.10 Method: vCreateAbortReason

Create abort reason entry.

Args: _tbl_test_result_id : UUID of test result.
_tbl_abort_reason : abort reason.
_tbl_abort_message : detail message of abort.

Returns: None.

3.1.11 Method: vCreateReanimation

Create reanimation entry.

Args: _tbl_test_result_id : UUID of test result.
_tbl_num_of_reanimation : counter of target reanimation during execution.

Returns: None.

3.1.12 Method: vCreateCCRdata

Create CCR data per test case.

Args: _tbl_test_case_id : test case ID.
lCCRdata : list od CCR data.

Returns: None.

3.1.13 Method: vFinishTestResult

Finish upload:

- First do bulk insert of rest of test cases if buffer is not empty.
- Then set state to "new report".

3.1.14 Method: vUpdateEvtbls

Call update_evtbls stored procedure. Method: vUpdateEvtbl ----------
Call update_evtbl stored procedure to update provided test_result_id. Method: vEnableForeignKeyCheck ----------
Switch foreign_key_checks flag. Method: sGetLatestFileID ----------
Get latest file ID from tbl_file table.

Args: sResultID : UUID of test result to get the latest file ID.

Returns: sFileID : file ID.

3.1.15 Method: vUpdateFileEndTime

Update test file end time.

Args: sFileID : file ID to be updated.
sEndtime : end time information.

Returns: None.

3.1.16 Method: vUpdateResultEndTime

Update test result end time.

Args: sResultID : test result UUID to be updated.
sEndtime : end time information.

Returns: None.

3.1.17 Method: bExistingResultID

Verify the given test result UUID is existing in tbl_result table or not.

Args: sResultID : test result UUID to be verified.

Returns: bExisting : True if test result UUID is already existing.

Chapter 4

__init__.py

RobotResults2DB package provides an interface to import robot result(s) from output.xml result file to TestResultWebApp database.

TestResultWebApp is available in Github as below repo: https://github.com/test-fullautomation/TestResultWebApp

Chapter 5

robot2db.py

5.1 Function: is_valid_uuid

Verify the given UUID is valid or not.

Args: uuid_to_test : UUID to be verified.
version (optional): UUID version.

Returns: True if the given UUID is valid.

5.2 Function: get_from_tags

Extract testcase information from tags.

Example: TCID-xxxx, FID-xxxx, ...

Args: lTags : list of tag information.
reInfo : regex to get the expectated info (ID) from tag info.

Returns: lInfo : list of expected information (ID)

5.3 Function: get_branch_from_swversion

Get branch name from software version information.

Convention of branch information in suffix of software version:

- All software version with .0F is the main/freature branch. The leading number is the current year. E.g. 17.0F03
- All software version with .1S, .2S, ... is a stabi branch. The leading number is the year of branching out for stabilization. The number before "S" is the order of branching out in the year.

Args: sw_version : software version.

Returns: branch_name : branch name.

5.4 Function: format_time

Format the given time string to TestResultWebApp's format for importing to db.

Args: stime : string of time.

Returns: TestResultWebApp's time format.

5.5 Function: process_suite_metadata

Try to find metadata information from all suite levels.

Note: Metadata at top suite level has a highest priority.

Args: suite : Robot suite object.
default_metadata: initial Metadata information for updating.

Returns: dMetadata : dictionary of Metadata information.

5.6 Function: process_metadata

Extract metadata from suite result bases on DEFAULT_METADATA

Args: metadata : Robot metadata object.
default_metadata: initial Metadata information for updating.

Returns: dMetadata : dictionary of Metadata information.

5.7 Function: process_suite

Process to the lowest suite level (test file):

- Create new file and its header information
- Then, process all child test cases

Args: suite : Robot suite object.
_tbl_test_result_id : UUID of test result for importing.
root_metadata : metadata information.
dConfig: configuration data which is parsed from given json configuration file.

Returns: None.

5.8 Function: process_test

Process test case data and create new test case record.

Args: db : database object.
test : Robot test object.
file_id : file ID for mapping.
test_result_id : test result ID for mapping.
metadata_info : metadata information.
test_number : order of test case in file.

Returns: None.

5.9 Function: process_config_file

Parse information from configuration file:

- 'component': :

```
{
   "component" : {
      "componentA" : "componentA/path/to/testcase",
      "componentB" : "componentB/path/to/testcase",
      "componentC" : [
         "componentC1/path/to/testcase",
         "componentC2/path/to/testcase"
      ]
   }
}
Then all testcase which its path contain componentA/path/to/testcase
will be  belong to componentA, ...
```

- variant, 'version_sw': configuration file has low priority than command line

Args: config_file : path to configuration file.

Returns: dConfig : configuration object.

5.10 Function: validate_config

Validate the json configuration base on given schema.

Default schema just supports "component", "variant" and "version_sw" :

```
CONFIG_SCHEMA = {
   "component" : [str, dict],
   "variant"   : str,
   "version_sw": str,
}
```

Args: dConfig : json configuration object to be verified.

sSchema (optional): schema for the validation. CONFIG_SCHEMA is used as default.

bExitOnFail (optional): If True, exit tool in case the validation is fail.

Returns: bValid : True if the given json configuration data is valid.

5.11 Function: normailze_path

Normalize path file.

Args: sPath : string of path file to be normalized.

Returns: sNPath : string of normalized path file.

5.12 Function: truncate_string

Truncate input string before importing to database.

Args: sString : input string for truncation.

iMaxLength : max length of string to be allowed.

sEndChars (optional): end characters which are added to end of truncated string.

Returns: content : string after truncation.

5.13 Function: RobotResults2DB

Import robot results from output.xml to TestResultWebApp's database

Flow to import Robot results to database:

1. Process provided arguments from command line
2. Connect to database
3. Parse Robot results
4. Import results into database
5. Disconnect from database

Args: args : Argument parser object:

- outputfile : path to the output file or directory with output files to be imported.
- server : server which hosts the database (IP or URL).
- user : user for database login.
- password : password for database login.
- database : database name.
- recursive : if True, then the path is searched recursively for log files to be imported.
- dryrun : if True, then just check the RQM authentication and show what would be done.
- UUID : UUID used to identify the import and version ID on TestResultWebApp.
- variant : variant name to be set for this import.
- versions : metadata: Versions (Software;Hardware;Test) to be set for this import.
- config : configuration json file for component mapping information.

Returns: None.

5.14 Class: Logger

```
RobotResults2DB.robot2db
```

Logger class for logging message Method: config ----------

Configure Logger class.

Args: output_console : write message to console output.

output_logfile : path to log file output.

indent : offset indent.

dryrun : if set, a prefix as 'dryrun' is added for all messages.

Returns: None.

5.14.1 Method: log

Write log message to console/file output.

Args: msg : message to write to output.
color : color style for the message.
indent : offset indent.

Returns: None.

5.14.2 Method: log_warning

Write warning message to console/file output.

Args: msg : message to write to output.

Returns: None.

5.14.3 Method: log_error

Write error message to console/file output.

Args: msg : message to write to output.
fatal_error : if set, tool will terminate after logging error message.

Returns: None.

Chapter 6

Appendix

About this package:

Table 6.1: Package setup

Setup parameter	Value
Name	RobotResults2DB
Version	1.2.0
Date	04.07.2022
Description	Imports robot result(s) to TestResultWebApp database
Package URL	robotframework-testresultwebapptool
Author	Tran Duy Ngoan
Email	Ngoan.TranDuy@vn.bosch.com
Language	Programming Language :: Python :: 3
License	License :: OSI Approved :: Apache Software License
OS	Operating System :: OS Independent
Python required	>=3.0
Development status	Development Status :: 4 - Beta
Intended audience	Intended Audience :: Developers
Topic	Topic :: Software Development

Chapter 7

History

0.1.0	07/2022
Initial version	