RobotResults2RQM

v. 1.1.1

Tran Duy Ngoan

05.07.2022

Contents

Chapter 1

Introduction

RobotResults2RQM

!!! add introduction - if wanted !!

Chapter 2

Description

The RobotResults2RQM:

TO BE CONTINUED

Chapter 3

CRQM.py

3.1 Function: get_xml_tree

Parse xml object from file.

Args: file_name : path to file or file-like object. bdtd_validation : if True, validate against a DTD referenced by the document.

Returns: oTree : xml etree object

3.2 Class: CRQMClient

```
RobotResults2RQM.CRQM
```

CRQMClient class uses RQM REST APIs to get, create and update resources (testplan, testcase, test result, ...) on RQM - Rational Quality Manager

Resoure type mapping:

- buildrecord: Build Record
- configuration: Test Environment
- testplan: Test Plan
- testsuite: Test Suite
- suiteexecutionrecord: Test Suite Execution Record (TSER)
- testsuitelog: Test Suite Log
- testcase: Test Case
- executionworkitem: Test Execution Record (TCER)
- executionresult: Execution Result

3.2.1 Method: login

Log in RQM by provided user & password.

Note: When the authentication is successful, the JSESSIONID from cookies will be stored as header for later POST method.

Returns: True if successful, False otherwise.

3.2.2 Method: verifyProjectName

Verify the project name by searching it in project-areas XML response.

Note: The found project ID will be stored and:

- required for team-areas request (project name cannot be used)
- used for all later request urls instead of project name

Returns:

- True if the authentication is successful.
- False if the authentication is failed.

3.2.3 Method: disconnect

Disconnect from RQM Method: config --------------

Configure RQMClient with testplan ID, build, configuration, createmissing, ...

- Verify the existence of provided testplan ID.
- Verify the existences of provided build and configuration names before creating new ones.

Args: plan_id : testplan ID of RQM project for importing result(s).

build_name (optional) : the Build Record for linking result(s). Set it to None if not be used, the empty name ” may lead to error.

config_name (optional) : the Test Environment for linking result(s). Set it to None if not be used, the empty name ” may lead to error.

createmissing (optional) : in case this argument is set to True, the testcase without tcid information will be created on RQM.

updatetestcase (optional) : in case this argument is set to True, the information of testcase on RQM will be updated bases on robot testfile.

suite_id (optional) : testsuite ID of RQM project for importing result(s).

Returns: None.

3.2.4 Method: userURL

Return interaction URL of provided userID

Args: userID : the user ID

Returns: userURL : the interaction URL of provided userID

3.2.5 Method: integrationURL

Return interaction URL of provided reource and ID.

Note: ID can be internalID (contains only digits) or externalID.

Args: resourceType : the RQM resource type (e.g: ”testplan”, ”testcase”, ...)

id (optional) : ID of given resource. If given: the specified url to resource ID is returned. If ‘None‘: the url to resource type (to get all entity) is returned.

forceinternalID (optional) : force to return the url of resource as internal ID.

Returns: integrationURL : interaction URL of provided reource and ID.

3.2.6 Method: webIDfromResponse

Get internal ID (number) from response of POST method.

Note: Only executionresult has response text. Other resources has only response header.

Args: response : the response from POST method.
tagID : tag name which contains ID information.

Returns: resultId : internal ID (as number).

3.2.7 Method: webIDfromGeneratedID

Return web ID (ns2:webId) from generate ID by get resource data from RQM.

Note:

- This method is only used for generated testcase, executionworkitem and executionresult.
- buildrecord and configuration does not have ns2:webId in response data.

Args: resourrceType : the RQM resource type.
generateID : the Slug ID which is returned in Content-Location from POST response.

Returns: webID : web ID (number).

3.2.8 Method: getResourceByID

Return data of provided resource and ID by GET method

Args: resourrceType : the RQM resource type.
id : ID of resource.

Returns: res : response data of GET request.

3.2.9 Method: getAllByResource

Return all entries of provided resource by GET method.

Note: This method will try to fetch all entries in all pages of resource.

Args: resourrceType : the RQM resource type.

Returns: dReturn : a dictionary which contains response status, message and data.

3.2.10 Method: getAllBuildRecords

Get all available build records of project on RQM and store them into dBuildVersion property. Method: getAllConfigurations ----------------------------

Get all available configurations of project on RQM and store them into dConfiguation property. Method: getAllTeamAreas -----------------------

Get all available team-areas of project on RQM and store them into dTeamAreas property.

Example: { 'teamA' : '{host}/qm/process/project-areas/{project-id}/team-areas/{teamA-id}, 'teamB' : '{host}/qm/process/project-areas/{project-id}/team-areas/{teamB-id} }

3.2.11 Method: addTeamAreaNode

Append team-area node which contains URL to given team-area into xml template

Note: team-area information is case-casesensitive

Args: root : xml root object.

sTeam : team name to be added.

Returns: root : xml root object with addition team-area node.

3.2.12 Method: createTestcaseTemplate

Return testcase template from provided information.

Args: testcaseName : testcase name.

sDescription (optional) : testcase description.

sComponent (optional) : component which testcase is belong to.

sFID (optional) : function ID(requirement ID) for linking.

sTeam (optional) : team name for linking.

sRobotFile (optional) : link to robot file on source control.

sTestType (optional) : test type information.

sASIL (optional) : ASIL information.

sOwnerID (optional) : user ID of testcase owner.

sTCtemplate (optional) : existing testcase template as xml string. If not provided, template file under *RQM_templates* is used as default.

Returns: xml template as string.

3.2.13 Method: createTCERTemplate

Return testcase execution record template from provided information

Args: testcaseID : testcase ID.

testcaseName : testcase name.

testplanID : testplan ID for linking.

confID (optional) : configuration - Test Environment for linking.

sTeam (optional) : team name for linking.

sOwnerID (optional) : user ID of testcase owner.

Returns: xml template as string.

3.2.14 Method: createExecutionResultTemplate

Return testcase execution result template from provided information

Args: testcaseID : testcase ID.

testcaseName : testcase name.

testplanID : testplan ID for linking.

TCERID : testcase execution record (TCER) ID for linking.

resultState : testcase result status.

startTime : testcase start time.
endTime (optional) : testcase end time.
duration (optional) : testcase duration.
testPC (optional) : test PC which executed testcase.
testBy (optional) : user ID who executed testcase.
lastlog (optional) : traceback information (for Failed testcase).
buildrecordID (optional) : Build Record ID for linking.
sTeam (optional) : team name for linking.
sOwnerID (optional) : user ID of testcase owner.

Returns: xml template as string.

3.2.15 Method: createBuildRecordTemplate

Return build record template from provided build name

Args: buildName : Build Record name.

Returns: xml template as string.

3.2.16 Method: createConfigurationTemplate

Return configuration - Test Environment template from provided configuration name

Args: buildName : configuration - Test Environment name.

Returns: xml template as string.

3.2.17 Method: createTSERTemplate

Return testsuite execution record (TSER) template from provided configuration name

Args: testsuiteID : testsuite ID.
testsuiteName : testsuite name.
testplanID : testplan ID for linking.
confID (optional) : configuration - Test Environment ID for linking.
sOwnerID (optional) : user ID of testsuite owner.

Returns: xml template as string.

3.2.18 Method: createTestsuiteResultTemplate

Return testsuite execution result template from provided configuration name

Args: testsuiteID : testsuite ID.
testsuiteName : testsuite name.
TSERID : testsuite execution record (TSER) ID for linking.
lTCER : list of testcase execution records (TCER) for linking.
lTCResults : list of testcase results for linking.
startTime (optional) : testsuite start time.
endTime (optional) : testsuite end time.
duration (optional) : testsuite duration.
sOwnerID (optional) : user ID of testsuite owner.

Returns: xml template as string.

3.2.19 Method: createResource

Create new resource with provided data from template by POST method.

Args: resourceType : resource type.

content: xml template as string.

Returns: returnObj: a response dictionary which contains status, ID, status_code and error message. { 'success' : False, 'id': None, 'message': ", 'status_code': " }

3.2.20 Method: createBuildRecord

Create new build record.

Note: Tool will check if build record is already existing or not (both on RQM and current execution).

Args: sBuildSWVersion : build version - Build Record name.

forceCreate (optional) : if True, force to create new build record without existing verification.

Returns: returnObj: a response dictionary which contains status, ID, status_code and error message. { 'success' : False, 'id': None, 'message': ", 'status_code': " }.

3.2.21 Method: createConfiguration

Create new configuration - test environment.

Note: Tool will check if configuration is already existing or not (both on RQM and current execution).

Args: sConfigurationName : configuration - Test Environment name.

forceCreate (optional) : if True, force to create new Test Environment without existing verification.

Returns: returnObj: a response dictionary which contains status, ID, status_code and error message. { 'success' : False, 'id': None, 'message': ", 'status_code': " }

3.2.22 Method: updateResourceByID

Update data of provided resource and ID by PUT method.

Args: resourceType : resource type.

id : resource id.

content : xml template as string.

Returns: res : response object from PUT request.

3.2.23 Method: linkListTestcase2Testplan

Link list of test cases to provided testplan ID.

Args: testplanID : testplan ID to link given testcase(s).

lTestcases : list of testcase(s) to be linked with given testplan. If None (as default), lTestcaseIDs property will be used as list of testcase.

Returns: res : response object which contains status and error message. { 'success' : False, 'message': " }

3.2.24 Method: linkListTestcase2Testsuite

Link list of test cases to provided testsuite ID

Args: testsuiteID : testsuite ID to link given testcase(s).

lTestcases : list of testcase(s) to be linked with given testsuite. If None (as default), lTestcaseIDs property will be used as list of testcase.

Returns: res : response object which contains status and error message. { 'success' : False, 'message': " }

Chapter 4

__init__.py

RobotResults2RQM package provides an interface to import robot result(s) from output.xml result file to IBM Rational Quality Manager (RQM).

Chapter 5

robot2rqm.py

5.1 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)

5.2 Function: convert_to_datetime

Convert time string to datetime.

Args: time : string of time.

Returns: dt : datetime object

5.3 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.

5.4 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.

5.5 Function: process_suite

process robot suite for importing to RQM.

Args: RQMClient : RQMClient object.
suite : Robot suite object.

Returns: None.

5.6 Function: process_test

process robot test for importing to RQM.

Args: RQMClient : RQMClient object.
test : Robot test object.

Returns: None.

5.7 Function: RobotResults2RQM

Import robot results from output.xml to RQM - IBM Rational Quality Manager.

Flow to import Robot results to RQM:

1. Process provided arguments from command line
2. Login Rational Quality Management (RQM)
3. Parse Robot results
4. Import results into RQM
5. Link all executed testcases to provided testplan/testsuite ID

Args: args : Argument parser object:

- outputfile : path to the output file or directory with output files to be imported.
- host : RQM host url.
- project : RQM project name.
- user : user for RQM login.
- password : user password for RQM login.
- testplan : RQM testplan ID.
- recursive : if True, then the path is searched recursively for log files to be imported.
- createmissing : if True, then all testcases without fcid are created when importing.
- updatetestcase : if True, then testcases information on RQM will be updated bases on robot testfile.
- dryrun : if True, then just check the RQM authentication and show what would be done.

Returns: None.

5.8 Class: Logger

```
RobotResults2RQM.robot2rqm
```

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.

5.8.1 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.

5.8.2 Method: log_warning

Write warning message to console/file output.

Args: msg : message to write to output.

Returns: None.

5.8.3 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	RobotResults2RQM
Version	1.1.1
Date	05.07.2022
Description	Imports robot result(s) to IBM Rational Quality Manager (RQM)
Package URL	robotframework-testresult2rqmtool
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	

RobotResults2RQM.pdf

Created at 05.07.2022 - 16:47:07

by GenPackageDoc v. 0.20.2