RobotFramework_Testsuites

v. 0.2.2

Mai Dinh Nam Son

18.07.2022

Contents

Chapter 1

Introduction

RobotFramework_Testsuites

!!! add introduction - if wanted !!

Chapter 2

Description

The RobotFramework AIO testsuites management:

TO BE CONTINUED

Chapter 3

CConfig.py

3.1 Class: dotdict

Imported by:

```
from RobotFramework_Testsuites.Config.CConfig import dotdict
```

Subclass: dotdict Subclass of dict, with ”dot” (attribute) access to keys.

3.2 Class: CConfig

Imported by:

```
from RobotFramework_Testsuites.Config.CConfig import CConfig
```

Class: CConfig

> Defines the properties of configuration
>
> Holds the identified config files.
>
> Level1 is highest priority, Level4 is lowest priority.
>
> **Level1:** Handed over by command line argument. **Level2:** Read from content of json config file .. code:: json
>
> > **{ ”default”: {** ”name”: ”robot_config.json”, ”path”: ”.../config/”
> > }, ”variant_0”: { ”name”: ”robot_config.json”, ”path”: ”.../config/” }, ”variant_1”: { ”name”: ”robot_config_variant_1.json”, ”path”: ”.../config/” }, ...
> >
> > }
> >
> > According to the `ConfigName`, Testsuites-Management package will choose the corresponding config file.

`".../config/"` indicats the relative path to json config file, Testsuites-Management will recursively find the `config` folder.

> **Level3:** Read in testsuite folder `/config/robot_config.json` **Level4:** Read from RobotFramework AIO install folder `/RobotFramework/defaultconfig/robot_config.json`

3.2.1 Method: loadCfg

Method: loadCfg This loadCfg method uses to load configuration’s parameters from json files.

Arguments:

- No input parameter is required

Returns:

- No return variable

3.2.2 Method: updateCfg

Method: updateCfg This updateCfg method updates preprocessor, global or local params base on RobotFramework AIO local

config or any json config file according to purpose of specific testsuite.

Arguments:

- `sUpdateCfgFile`

  / *Condition*: required / *Type*: string
  The path of json file which wants to update configuration parameters.

Returns:

- No return variable

3.2.3 Method: verifyRbfwVersion

Method: verifyRbfwVersion

This verifyRbfwVersion validates the current RobotFramework AIO version with maximum and minimum

version (if provided in the configuration file).

In case the current version is not between min and max version, then the execution of

testsuite is terminated with "unknown" state

Arguments:

- No input parameter is required

Returns:

- No return variable

3.2.4 Method: bValidateMinVersion

Method: bValidateMinVersion This bValidateMinVersion validates the current version with required minimun version.

Arguments:

- `tCurrentVersion`

  / *Condition*: required / *Type*: tuple
  Current RobotFramework AIO version.

- `tMinVersion`

  / *Condition*: required / *Type*: tuple
  The minimum version of RobotFramework AIO.

Returns:

- `True or False`

3.2.5 Method: bValidateMaxVersion

Method: bValidateMaxVersion This bValidateMaxVersion validates the current version with required minimun version.

Arguments:

- `tCurrentVersion`

  / *Condition*: required / *Type*: tuple
  Current RobotFramework AIO version.

- `tMinVersion`

  / *Condition*: required / *Type*: tuple
  The minimum version of RobotFramework AIO.

Returns:

- `True or False`

3.2.6 Method: bValidateSubVersion

Method: bValidateSubVersion This bValidateSubVersion validates the format of provided sub version and parse it into sub tuple

for version comparision.

Arguments:

- `sVersion`

  / *Condition*: required / *Type*: string
  The version of RobotFramework AIO.

Returns:

- `lSubVersion`

  / *Type*: tuple /

3.2.7 Method: tupleVersion

Method: tupleVersion

This tupleVersion returns a tuple which contains the (major, minor, patch) version.

(remaining content needs to be fixed and restored)

Arguments:

- `sVersion`

  / *Condition*: required / *Type*: string
  The version of RobotFramework AIO.

Returns:

- `lVersion`

  / *Type*: tuple /

3.2.8 Method: versioncontrol_error

Method: versioncontrol_error

Wrapper version control error log:

> Log error message of version control due to reason and set to unknown state.
> reason can only be ”conflict_min”, ”conflict_max” and ”wrong_minmax”.

Arguments:

- `reason`

  / *Condition*: required / *Type*: string

- `version1`

  / *Condition*: required / *Type*: string

- `version2`

  / *Condition*: required / *Type*: string

Returns:

- No return variable

Chapter 4

COnFailureHandle.py

4.1 Class: COnFailureHandle

Imported by:

```
from RobotFramework_Testsuites.Keywords.COnFailureHandle import COnFailureHandle
```

4.1.1 Method: is_noney

4.1.2 Method: register_keyword_run_on_failure

TBD

Chapter 5

CSetup.py

5.1 Class: CSetupKeywords

Imported by:

```
from RobotFramework_Testsuites.Keywords.CSetup import CSetupKeywords
```

Class: CSetupKeywords

This CSetupKeywords class uses to define the setup keywords which are using in suite setup and teardown of

robot test script.

`Testsuite Setup` keyword loads the RobotFramework AIO configuration, checks the version of RobotFramework AIO,

and logs out the basic information of the robot run.

`Testsuite Teardown` keyword currently do nothing, it's defined here for future requirements.

`Testcase Setup` keyword currently do nothing, it's defined here for future requirements.

`Testcase Teardown` keyword currently do nothing, it's defined here for future requirements.

5.1.1 Method: testsuite_setup

Method: testsuite_setup

This testsuite_setup defines the `Testsuite Setup` which is used to loads the RobotFramework AIO configuration,

checks the version of RobotFramework AIO, and logs out the basic information of the robot run.

Arguments:

* `sTestsuiteCfgFile`

/ *Condition*: required / *Type*: string

`sTestsuiteCfgFile=''` and vairiable `config_file` is not set Robotframework AIO will check

for configuration level 3, and level 4.

`sTestsuiteCfgFile` is set with a <json_config_file_path> and vairiable `config_file` is not set

Robotframework AIO will load configuration level 2.

Returns:

- No return variable

5.1.2 Method: testsuite_teardown

Method: testsuite_teardown

This testsuite_teardown defines the `Testsuite Teardown` keyword, currently this keyword does nothing,

it's defined here for future requirements.

5.1.3 Method: testcase_setup

Method: testcase_setup

This testcase_setup defines the `Testcase Setup` keyword, currently this keyword does nothing,

it's defined here for future requirements.

5.1.4 Method: testcase_teardown

Method: testcase_teardown

This testcase_teardown defines the `Testcase Teardown` keyword, currently this keyword does nothing,

it's defined here for future requirements.

5.1.5 Method: update_config

Method: update_config This update_config defines the `Update Config` keyword which is using update the configuration object

of RobotFramework AIO.

Arguments:

- `sCfgFile`

  / *Condition*: required / *Type*: string
  The path of Json configuration file.

Returns:

- No return variable

5.2 Class: CGeneralKeywords

Imported by:

```
from RobotFramework_Testsuites.Keywords.CSetup import CGeneralKeywords
```

Class: CGeneralKeywords

This CGeneralKeywords class uses to define the keywords which will be using in RobotFramework AIO test script.

`Get Config` keyword gets the current config object of robot run.

`Load Json` keyword loads json file then return json object.

In case, new robot keyword is required, it will be defined and implemented in this class.

5.2.1 Method: get_config

Method: get_config

This get_config defines the `Get Config` keyword gets the current config object of RobotFramework AIO.

Arguments:

- No parameter is required

Returns:

- `oConfig.oConfigParams`

  / *Type*: json /

5.2.2 Method: load_json

Method: load_json

This load_json defines the `Load Json` keyword which loads json file then return json object.

Arguments:

- `jsonfile`

  / *Condition*: required / *Type*: string
  The path of Json configuration file.

- `level`

  / *Condition*: required / *Type*: int
  Level = 1 -> loads the content of jsonfile.
  level != 1 -> loads the json file which is set with variant (likes loading config level2)

Returns:

- `oJsonData`

  / *Type*: json /

Chapter 6

CStruct.py

6.1 Class: CStruct

Imported by:

```
from RobotFramework_Testsuites.Utils.CStruct import CStruct
```

Class: CStruct

This `CStruct` class creates the given attributes dynamically at runtime.

Chapter 7

Event.py

7.1 Class: Event

Imported by:

```
from RobotFramework_Testsuites.Utils.Events.Event import Event
```

7.1.1 Method: trigger

Chapter 8

ScopeEvent.py

8.1 Class: ScopeEvent

Imported by:

```
from RobotFramework_Testsuites.Utils.Events.ScopeEvent import ScopeEvent
```

8.1.1 Method: trigger

8.2 Class: ScopeStart

Imported by:

```
from RobotFramework_Testsuites.Utils.Events.ScopeEvent import ScopeStart
```

8.3 Class: ScopeEnd

Imported by:

```
from RobotFramework_Testsuites.Utils.Events.ScopeEvent import ScopeEnd
```

Chapter 9

__init__.py

9.1 Function: on

9.2 Function: dispatch

9.3 Function: register_event

Chapter 10

LibListener.py

10.1 Class: LibListener

Imported by:

```
from RobotFramework_Testsuites.Utils.LibListener import LibListener
```

Class: LibListener

This `LibListener` class defines the hook methods.

`_start_suite` hooks to every starting testsuite of robot run.

`_end_suite` hooks to every ending testsuite of robot run.

`_start_test` hooks to every starting test case of robot run.

`_end_test` hooks to every ending test case of robot run.

Chapter 11

__init__.py

11.1 Class: RobotFramework_Testsuites

Imported by:

```
from RobotFramework_Testsuites.__init__ import RobotFramework_Testsuites
```

RobotFramework_Testsuites is the Bosch testing library for Robot Framework. RobotFramework_Testsuites control peripheral devices, tools and target under testing. Method: run_keyword -------------------

11.1.1 Method: get_keyword_tags

11.1.2 Method: get_keyword_documentation

11.1.3 Method: failure_occurred

11.2 Class: CTestsuitesCfg

Imported by:

```
from RobotFramework_Testsuites.__init__ import CTestsuitesCfg
```

Chapter 12

version.py

12.1 Function: robfwaio_version

Return testsuitemanagement version as Robot framework AIO version

Chapter 13

Appendix

About this package:

Table 13.1: Package setup

Setup parameter	Value
Name	RobotFramework_Testsuites
Version	0.2.2
Date	18.07.2022
Description	Functionality to manage RobotFramework testsuites
Package URL	robotframework-testsuitesmanagement
Author	Mai Dinh Nam Son
Email	son.maidinhnam@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 14

History

0.1.0	06/2022
Initial version	

RobotFramework_Testsuites.pdf

Created at 13.10.2022 - 17:52:26

by GenPackageDoc v. 0.32.0