

TestSuite Log

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	3	3	0	00:00:01	
All Tests	3	3	0	00:00:01	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
TestSuite	3	3	0	00:00:01	

Test Execution Log

☐ SUITE

TestSuite

00:00:01.105

Full Name:

TestSuite

Documentation:

Suite description

Source:

D:\Projetos\robot\robot_db\UNISAL\TestSuite\TestSuite.robot

Start / End / Elapsed:

20201202 18:52:41.148 / 20201202 18:52:42.253 / 00:00:01.105

Status:

3 critical test, 3 passed, 0 failed
3 test total, 3 passed, 0 failed

☐ TEST

1 Connect to GitHub and log Issues with label=question and state=open

00:00:00.682

Full Name:

TestSuite.1 Connect to GitHub and log Issues with label=question and state=open

Start / End / Elapsed:

20201202 18:52:41.485 / 20201202 18:52:42.167 / 00:00:00.682

Status:

PASS (critical)

☐ SETUP

RequestsLibrary.Create Session Session, \${HOST_GITHUB}, verify=True

00:00:00.001

Documentation:

Create Session: create a HTTP session to a server

Start / End / Elapsed:

20201202 18:52:41.485 / 20201202 18:52:41.486 / 00:00:00.001

18:52:41.486

INFO

Creating Session using : alias=Session,
url=https://api.github.com/repos/davimargelo/robot_db, headers={},
cookies={}, auth=None, timeout=None, proxies=None, verify=True,
debug=0

☐ KEYWORD

Teste_Keywords.Connect to Git Repository and log issues

00:00:00.678

Start / End / Elapsed:

20201202 18:52:41.487 / 20201202 18:52:42.165 / 00:00:00.678

☐ KEYWORD

\${resp} = RequestsLibrary.Get Request Session, /issues?labels=question&state=open

00:00:00.644

Documentation:

Send a GET request on the session object found using the given `alias`

Start / End / Elapsed:

20201202 18:52:41.487 / 20201202 18:52:42.131 / 00:00:00.644

18:52:42.131

INFO

GET Request : url=https://api.github.com/repos/davimargelo/robot_db/issues?
labels=question&state=open
path_url=/repos/davimargelo/robot_db/issues?labels=question&state=open
headers={'User-Agent': 'python-requests/2.24.0', 'Accept-Encoding': 'gzip,
deflate', 'Accept': '*/*', 'Connection': 'keep-alive'}
body=None

18:52:42.131

INFO

GET Response : url=https://api.github.com/repos/davimargelo/robot_db/issues?
labels=question&state=open
status=200, reason=OK
body=
[{"url": "https://api.github.com/repos/davimargelo/robot_db/issues/1", "repository
_url": "https://api.github.com/repos/davimargelo/robot_db", "labels_url": "https://
api.github.com/repos/davimargelo/robot_db/issues/1/labels{/name}", "comments_url"
: "https://api.github.com/repos/davimargelo/robot_db/issues/1/comments", "events_u
rl": "https://api.github.com/repos/davimargelo/robot_db/issues/1/events", "html_ur
l": "https://github.com/davimargelo/robot_db/issues/1", "id": 755619119, "node_id": "
MDU6SXNzdWU3NTU2MTkxMTk=", "number": 1, "title": "Question and Opened", "user":
{ "login": "davimargelo", "id": 56364191, "node_id": "MDQ6VXNlcjU2MzY0MTkx", "avatar_ur
l": "https://avatars0.githubusercontent.com/u/56364191?
v=4", "gravatar_id": "", "url": "https://api.github.com/users/davimargelo", "html_url"
: "https://github.com/davimargelo", "followers_url": "https://api.github.com/users
/davimargelo/followers", "following_url": "https://api.github.com/users/davimargel
o/following{/other_user}", "gists_url": "https://api.github.com/users/davimargelo/
gists{/gist_id}", "starred_url": "https://api.github.com/users/davimargelo/starred
{/owner}
{/repo}", "subscriptions_url": "https://api.github.com/users/davimargelo/subscript
ions", "organizations_url": "https://api.github.com/users/davimargelo/orgs", "repos

```
_url": "https://api.github.com/users/davimargelo/repos", "events_url": "https://api.github.com/users/davimargelo/events{/privacy}", "received_events_url": "https://api.github.com/users/davimargelo/received_events", "type": "User", "site_admin": false}, "labels": [{"id": 2552070519, "node_id": "MDU6TGFiZWwNTUyMDcwNTE5", "url": "https://api.github.com/repos/davimargelo/robot_db/labels/question", "name": "question", "color": "d876e3", "default": true, "description": "Further information is requested"}], "state": "open", "locked": false, "assignee": null, "assignees": [], "milestone": null, "comments": 0, "created_at": "2020-12-02T21:22:19Z", "updated_at": "2020-12-02T21:22:19Z", "closed_at": null, "author_association": "OWNER", "active_lock_reason": null, "body": "", "performed_via_github_app": null}]
```

18:52:42.131 INFO \${resp} = <Response [200]>

☐ KEYWORD `JSONLibrary`. **Get Value From Json** \${resp.json()}, \$.title 00:00:00.033

Documentation: Get Value From JSON using JSONPath

Start / End / Elapsed: 20201202 18:52:42.131 / 20201202 18:52:42.164 / 00:00:00.033

18:52:42.164 INFO \${title} = ['Question and Opened']

☐ KEYWORD `BuiltIn`. **Log To Console** \${title} 00:00:00.001

Documentation: Logs the given message to the console.

Start / End / Elapsed: 20201202 18:52:42.164 / 20201202 18:52:42.165 / 00:00:00.001

☐ TEST **2 Create folder with txt and log it then delete it** 00:00:00.008

Full Name: TestSuite.2 Create folder with txt and log it then delete it

Start / End / Elapsed: 20201202 18:52:42.169 / 20201202 18:52:42.177 / 00:00:00.008

Status: PASS (critical)

☐ SETUP `RequestsLibrary`. **Create Session** Session, \${HOST_GITHUB}, verify=True 00:00:00.001

Documentation: Create Session: create a HTTP session to a server

Start / End / Elapsed: 20201202 18:52:42.170 / 20201202 18:52:42.171 / 00:00:00.001

18:52:42.170 INFO Creating Session using : alias=Session, url=https://api.github.com/repos/davimargelo/robot_db, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=True, debug=0

☐ KEYWORD `Teste_Keywords`. **Create folder with txt** 00:00:00.005

Start / End / Elapsed: 20201202 18:52:42.171 / 20201202 18:52:42.176 / 00:00:00.005

☐ KEYWORD `OperatingSystem`. **Create File** D:/robot/teste.txt, Hello, world! 00:00:00.002

Documentation: Creates a file with the given content and encoding.

Start / End / Elapsed: 20201202 18:52:42.171 / 20201202 18:52:42.173 / 00:00:00.002

18:52:42.173 INFO Created file 'D:\robot\teste.txt'.

☐ KEYWORD `OperatingSystem`. **List Directory** D:/robot/ 00:00:00.002

Documentation: Returns and logs items in a directory, optionally filtered with pattern.

Start / End / Elapsed: 20201202 18:52:42.173 / 20201202 18:52:42.175 / 00:00:00.002

18:52:42.173 INFO Listing contents of directory 'D:\robot'.

18:52:42.175 INFO 2 items:
PROVA_ARQUIVOS
teste.txt

☐ KEYWORD `OperatingSystem`. **Remove File** D:/robot/teste.txt 00:00:00.001

☐ TEST **2 Connect to database and log users from table 'ACCOUNT'** 00:00:00.072

Full Name: TestSuite.2 Connect to database and log users from table 'ACCOUNT'

Start / End / Elapsed: 20201202 18:52:42.180 / 20201202 18:52:42.252 / 00:00:00.072

Status: PASS (critical)

☐ SETUP `RequestsLibrary`. **Create Session** Session, \${HOST_GITHUB}, verify=True 00:00:00.001

Documentation: Create Session: create a HTTP session to a server

Start / End / Elapsed: 20201202 18:52:42.181 / 20201202 18:52:42.182 / 00:00:00.001

18:52:42.182 INFO Creating Session using : alias=Session, url=https://api.github.com/repos/davimargelo/robot_db, headers={}, cookies={}, auth=None, timeout=None, proxies=None, verify=True, debug=0

☐ KEYWORD `Teste_Keywords`. **Connect to database prova_robotframework and execute script** 00:00:00.068

Start / End / Elapsed: 20201202 18:52:42.183 / 20201202 18:52:42.251 / 00:00:00.068

☐ KEYWORD `DatabaseLibrary`. **Connect To Database** psycopg2, prova_robotframework, postgres, postgres, localhost, 5500 00:00:00.030

Documentation: Loads the DB API 2.0 module given `dbapiModuleName` then uses it to connect to the database using `dbName`, `dbUsername`, and `dbPassword`.

Start / End / Elapsed: 20201202 18:52:42.183 / 20201202 18:52:42.213 / 00:00:00.030

18:52:42.200 INFO Connecting using : psycopg2.connect(database=prova_robotframework,

user=postgres, password=postgres, host=localhost, port=5500)

☐ **KEYWORD** DatabaseLibrary. **Execute Sql Script** ./create_table_accounts.sql 00:00:00.005

Documentation: Executes the content of the `sqlScriptFileName` as SQL commands. Useful for setting the database to a known state before running your tests, or clearing out your test data after running each a test. Set optional input `sansTran` to True to run command without an explicit transaction commit or rollback.

Start / End / Elapsed: 20201202 18:52:42.214 / 20201202 18:52:42.219 / 00:00:00.005

18:52:42.215 INFO Executing : Execute SQL Script | ./create_table_accounts.sql

☐ **KEYWORD** DatabaseLibrary. **Execute Sql String** INSERT INTO account(user_id, username, password, email) VALUES (1, 'Joao Silva', 'robot987', 'joaosilva@unisal.br') 00:00:00.006

Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input `sansTran` to True to run command without an explicit transaction commit or rollback.

Start / End / Elapsed: 20201202 18:52:42.219 / 20201202 18:52:42.225 / 00:00:00.006

18:52:42.219 INFO Executing : Execute SQL String | INSERT INTO account(user_id, username, password, email) VALUES (1, 'Joao Silva', 'robot987', 'joaosilva@unisal.br')

☐ **KEYWORD** @{queryResult} = DatabaseLibrary. **Query** Select * from account 00:00:00.004

Documentation: Uses the input `selectStatement` to query for the values that will be returned as a list of tuples. Set optional input `sansTran` to True to run command without an explicit transaction commit or rollback. Set optional input `returnAsDict` to True to return values as a list of dictionaries.

Start / End / Elapsed: 20201202 18:52:42.225 / 20201202 18:52:42.229 / 00:00:00.004

18:52:42.225 INFO Executing : Query | Select * from account

18:52:42.229 INFO @{queryResult} = [(1, 'Joao Silva', 'robot987', 'joaosilva@unisal.br')]

☐ **KEYWORD** BuiltIn. **Log To Console** @{queryResult} 00:00:00.000

Documentation: Logs the given message to the console.

Start / End / Elapsed: 20201202 18:52:42.229 / 20201202 18:52:42.229 / 00:00:00.000

☐ **KEYWORD** DatabaseLibrary. **Execute Sql String** TRUNCATE TABLE ACCOUNT 00:00:00.021

Documentation: Executes the sqlString as SQL commands. Useful to pass arguments to your sql. Set optional input `sansTran` to True to run command without an explicit transaction commit or rollback.

Start / End / Elapsed: 20201202 18:52:42.230 / 20201202 18:52:42.251 / 00:00:00.021

18:52:42.230 INFO Executing : Execute SQL String | TRUNCATE TABLE ACCOUNT