

# カスタム動作で自作のPythonファイルを使用する

## GOAL

- 1.自作Pythonファイル関数を使用したカスタム動作を作成できるようにする
- 2.作成したカスタム動作でテスト実行できることを確認する

## 使用した環境・バージョン

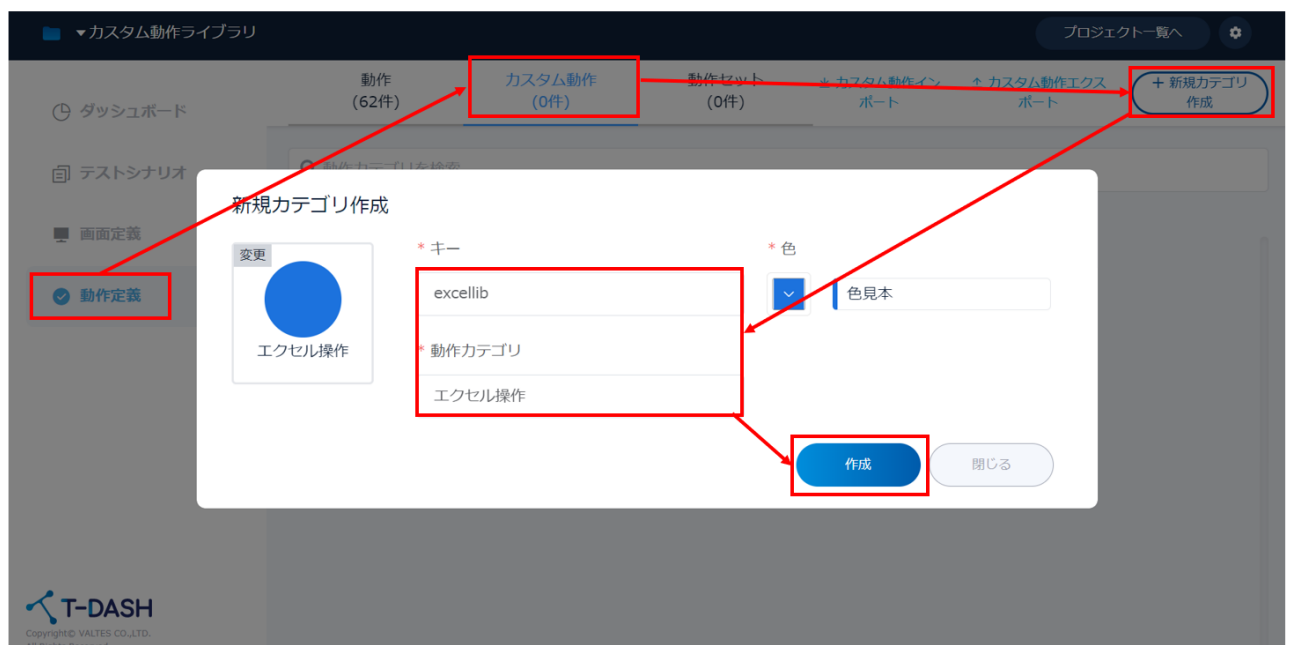
OS : Windows10 64bit

ブラウザ : Chrome 103.0.5060.134

T-DASH : 1.4.000

## 手順

1. 動作定義-> カスタム動作を選択
2. +新規カテゴリ作成ボタンを押下して任意のカテゴリを作成



3. +カスタム動作を追加ボタンを押下



#### 4. 動作名、フォーマットを任意で入力 ※今回例では設定値を使用するキーワードフォーマットを作成します

カスタム動作  
動作関数作成

保存する

× 閉じる

\* 動作名

エクセルファイルを作成する

利用変数

☐ 「画面名」 ☐ 「要素名」 ☒ 「設定値」

動作種類

☒ 操作関数 ☐ 検証関数

\* フォーマット

「設定値」でエクセルファイルを作成する

説明 入力すると?の吹き出しで表示されます

説明を入力してください

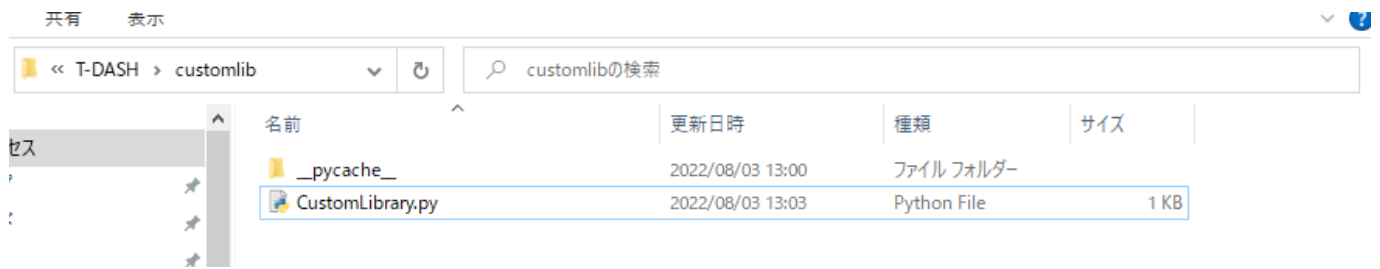
+ ライブラリを追加する

	A	B	C	D	E	F
1	「\${value}」でエクセルファイルを作成する					
2		[Arguments]	\${value}			
3						
4						
5						

#### 5. +ライブラリを追加するボタンを押下して自作ライブラリのフォルダを開くリンクを押下



## 6. 自作したPythonファイルを格納



今回自作したPythonファイルの中身

```
from robot.api.deco import keyword
from openpyxl import Workbook

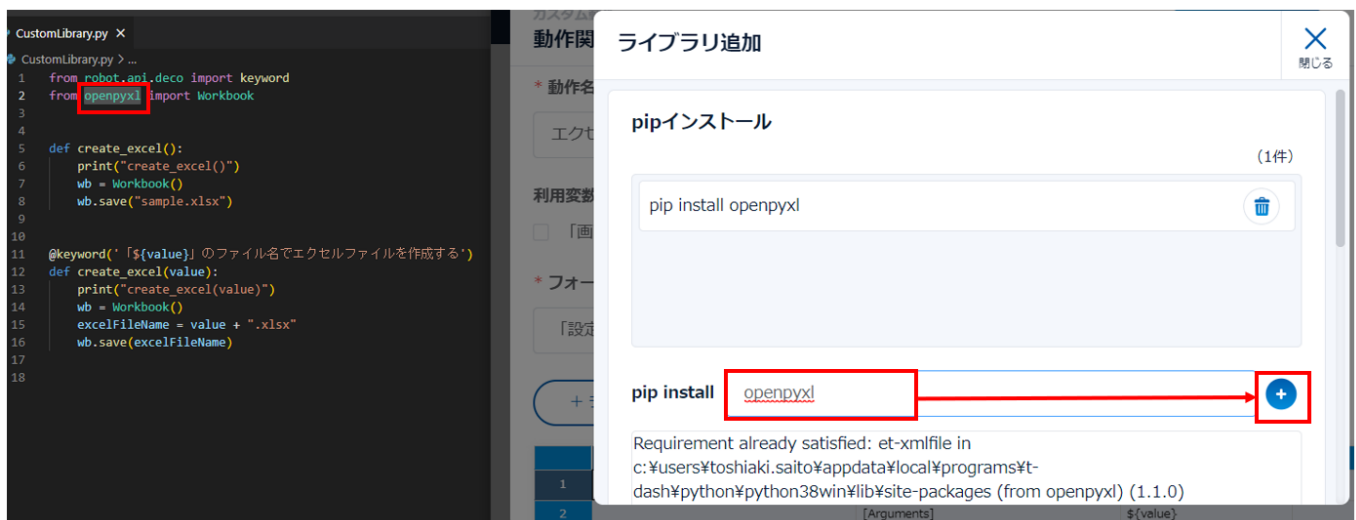
def create_excel1():
    print("create_excel1")
    wb = Workbook()
    wb.save("sample.xlsx")

@keyword('「${value}」のファイル名でエクセルファイルを作成する')
def create_excel2(value):
    print("create_excel2")
    wb = Workbook()
    excelFileName = value + ".xlsx"
    wb.save(excelFileName)
```

7. 再読み込みリンク -> +ボタン押下で自作Pythonファイルを追加



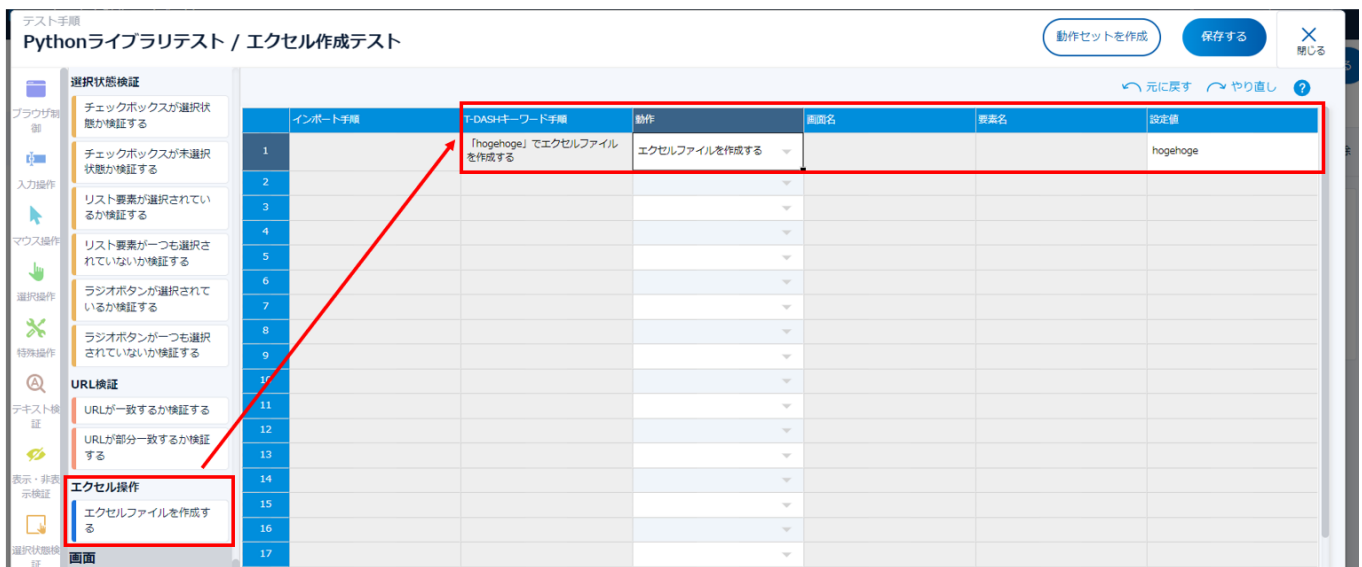
8. 自作Pythonファイルで使用しているライブラリをインストール ※Python標準のライブラリであれば追加不要



9. テストシナリオを選択して任意のテストシナリオ、テストケースを作成



10. **テストケースリンク**を押下->作成したカスタム動作手順を追加して保存 ※今回の例では設定値に入力した値でエクセルファイルが作られます



11. **選択したテストを実行する**を押下してテスト実行



12. **テストレポートを開く**を押下->LOGボタン押下でカスタム動作の関数が呼ばれていることを確認

テスト実行

テスト実行を完了する

テストレポートを開く

閉じる

1件のテストケース

100%

※実行中のテスト前後 10件 を表示します

インデックス	テストシナリオ	テストケース	手順	テスト結果
1	1	Pythonライブラリテスト	エクセル作成テスト	成功

完了時刻:  
20220804 10:23:24.409

LOG

日本語

## テストレポート

```
CustomLibrary.py X
CustomLibrary.py > create_excel2
1 from robot.api.deco import keyword
2 from openpyxl import Workbook
3
4
5 def create_excel1():
6     print("create_excel1")
7     wb = Workbook()
8     wb.save("sample.xlsx")
9
10
11 @keyword('「${value}」のファイル名でエクセルファイルを作成する')
12 def create_excel2(value):
13     print("create_excel2")
14     wb = Workbook()
15     excelFileName = value + ".xlsx"
```

message:

PASS = 1.Force Stop

- KEYWORD excellib. 「hogehoge」でエクセルファイルを作成する  
Start / End / Elapsed: 20220804 10:23:23.364 / 20220804 10:23:23.394 / 00:00:00.030
- KEYWORD CustomLibrary. 「\${value}」のファイル名でエクセルファイルを作成する  
Start / End / Elapsed: 20220804 10:23:23.365 / 20220804 10:23:23.382 / 00:00:00.017
- KEYWORD CustomLibrary. Create Excel1  
Start / End / Elapsed: 20220804 10:23:23.383 / 20220804 10:23:23.394 / 00:00:00.011

今回の例ではtdash.exe配下にエクセルが作成される

AppData > Local > Programs > T-DASH					T-DASHの検索	
	名前	更新日時	種類	サイズ		
	projects	2022/08/03 13:19	ファイル フォルダ			
	Python	2022/03/30 13:47	ファイル フォルダ			
	._log.txt.lock	2022/08/04 10:09	LOCK ファイル	0 KB		
	.env	2022/08/03 12:34	ENV ファイル	1 KB		
	alembic.ini	2022/03/30 11:30	構成設定	4 KB		
	chromedriver.exe	2022/08/04 10:09	アプリケーション	11,606 KB		
	geckodriver.exe	2022/04/12 9:28	アプリケーション	3,652 KB		
notification	hogehoge.xlsx	2022/08/04 10:23	Microsoft Excel ワ...	5 KB		
	IEDriverServer.exe	2022/07/14 13:20	アプリケーション	3,010 KB		
	information_checked_state.pkl	2022/08/04 10:08	PKL ファイル	1 KB		
Files	log.txt	2022/08/04 10:09	TXT ファイル	357 KB		
	msedgedriver.exe	2022/08/01 9:44	アプリケーション	14,820 KB		
	nodeunlock.lic	2022/06/30 16:48	LIC ファイル	1 KB		
	param.robot	2022/08/02 15:02	ROBOT ファイル	1 KB		
	proxy.conf	2022/07/28 11:55	CONF ファイル	1 KB		
	root.db	2022/08/04 10:09	Data Base File	64 KB		
	sample.xlsx	2022/08/04 10:23	Microsoft Excel ワ...	5 KB		
	tdash.exe	2022/08/03 10:01	アプリケーション	193,269 KB		