




Advance Robot Framework















Somkiat

Home









Somkiat Puisungnoen

Update Info
1


View Activity Log
10+

...


Timeline
About
Friends 3,138
Photos
More ▾



When did you work at Opendream?
×


...
22 Pending Items



Intro


Software Craftsmanship



Software Practitioner at สยามชำนาญกิจ พ.ศ. 2556



Agile Practitioner and Technical at SPRINT3r



Post


Photo/Video



Live Video


Life Event



What's on your mind?


Public ▾

Post



Somkiat Puisungnoen

15 mins · Bangkok ·


Java and Bigdata

...

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

Advance Robot Framework

3

Facebook interface for the page **somkiat.cc**. The top navigation bar includes the Facebook logo, a search bar, and user profile links for **Somkiat** and **Home**. The main navigation bar features **Page**, **Messages**, **Notifications** (with a red badge showing 3), **Insights**, **Publishing Tools**, **Settings**, and **Help**.

The page content area displays a large video of a man in a white Superman t-shirt with "SOMKIAT.CC" printed on it, posing against a white wall. A smaller thumbnail of the same video is shown in the left sidebar. The sidebar also includes the page name **somkiat.cc**, the handle **@somkiat.cc**, and a menu with **Home**, **Posts**, **Videos**, and **Photos**.

Below the video, there are interaction buttons: **Liked**, **Following**, **Share**, and a three-dot menu. A blue call-to-action button **+ Add a Button** is visible. A blue tooltip message reads: "Help people take action on this Page."



Agenda day 1

- Basic of python
- How to develop test library with python ?
- Type of test library
- Scope of test library
- How to use and publish test library ?
- Workshop



Agenda day 2

- Create keywords of test library
- How to generate documentation of test library ?
- Develop dynamic test library
- Develop test library by use case
- Workshop



Create library of Robot Framework



Programming language

Python
Java



Basic of Python



OOP with Python



Types of Library

Static library
Dynamic library
Hybrid library



Static Library



Hello World (1)

Create a new library with python

```
1  *** Settings ***
2  Library      HelloWorld.py
3
4  *** Testcases ***
5  First library
6  Say Hi
7
8  Second library with argument
9  Say Hi      somkiat
```



Hello World (2)

Create file **HelloWorld.py** and method **say_hi()**

```
1  def say_hi(name = ""):  
2      print("Say hi " + name)  
3
```



Hello World (3)

\$pybot test.robot

```
=====
Test
=====
First library | PASS |
-----
Second library with argument | PASS |
-----
Test | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
=====
```



Hello World (4)

See in [report.html](#)

Test Details

TotalsTagsSuitesSearch

Name:

Test

Status:

2 critical test, 2 passed, 0 failed
2 test total, 2 passed, 0 failed

Start / End Time:

20180603 22:28:28.540 / 20180603 22:28:28.578

Elapsed Time:

00:00:00.038

Log File:

[log.html#s1](#)

Name	Documentation
Test.First library	
Test.Second library with argument	



Improve naming of Library



Improve name of library

Need to change to HelloWorld

```
1  *** Settings ***
2  Library      HelloWorld
3
4  *** Testcases ***
5  First library
6      Say Hi
7
8  Second library with argument
9      Say Hi  somkiat
```



Run with python path

`$pybot --pythonpath . test.robot`

```
=====
Test
=====
First library                                     | PASS |
-----
Second library with argument                     | PASS |
-----
Test                                             | PASS |
2 critical tests, 2 passed, 0 failed
2 tests total, 2 passed, 0 failed
=====
```



Working with OOP



Hello World (2)

Create file **HelloWorld.py** and method **say_hi()**

```
1  class HelloWorld:
2      def say_hi(self, name = ""):
3          print("Say hi " + name)
4
5
```



Scope of test library



Scope of Test Library

TEST CASE (default)
TEST SUITE
GLOBAL



TEST CASE

Create a new instance for every test case



TEST SUITE

Create a new instance for every test suite



GLOBAL

Only one instance and shared by all test cases
and test suites

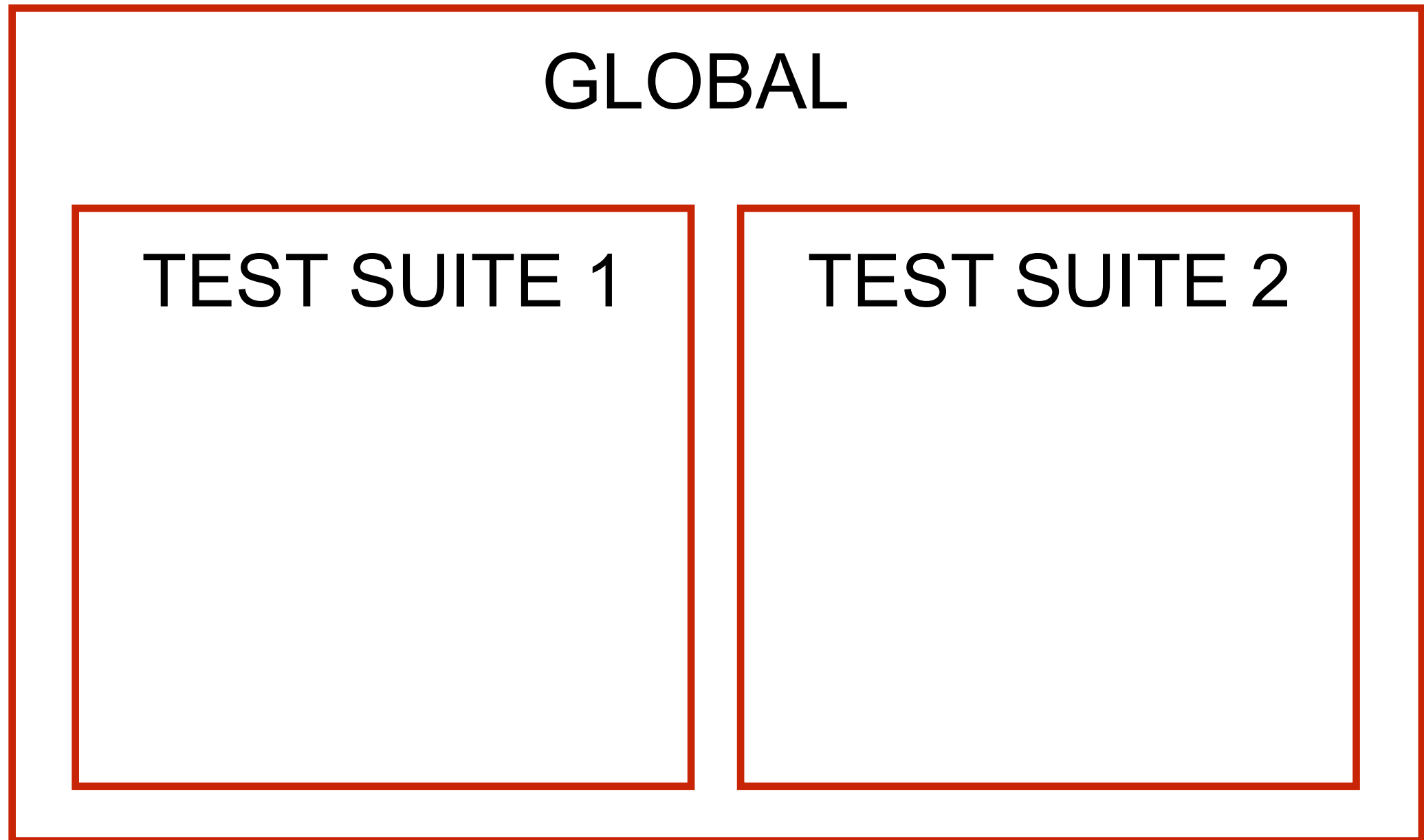


Scopes (1)

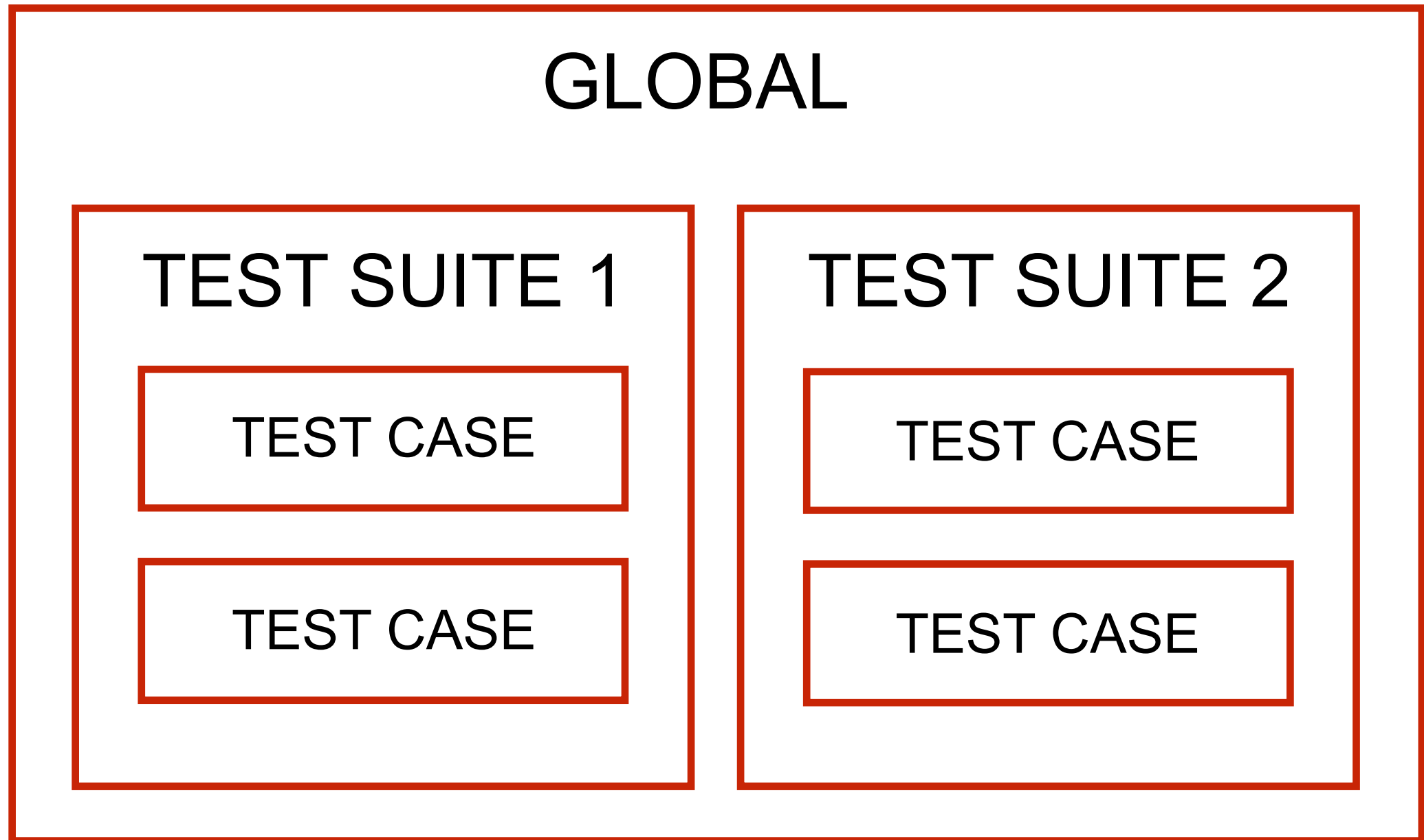
GLOBAL



Scopes (2)



Scopes (3)



Using TEST CASE scope

Create attribute ROBOT_LIBRARY_SCOPE

```
1  class HelloWorld:
2      ROBOT_LIBRARY_SCOPE = 'TEST CASE'
3
4      def __init__(self):
5          self.name = "Noname"
6
7      def say_hi(self):
8          print("Say hi " + self.name)
9
10     def say_hi2(self, name):
11         self.name = name
12         print("Say hi " + self.name)
13
```



My test cases

```
1    *** Settings ***
2    Library    HelloWorld
3
4    *** Testcases ***
5    First library
6        Say Hi
7
8    Second library with argument
9        Say Hi2    somkiat
10
11   Third library
12       Say Hi
```



Run with python path

☐ TEST First library

Full Name: Test.First library

Start / End / Elapsed: 20180603 23:22:02.490 / 20180603 23:22:02.491 / 00:00:00.001

Status: **PASS** (critical)

☐ KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:22:02.491 / 20180603 23:22:02.491 / 00:00:00.000

23:22:02.491 INFO Say hi Noname

Name = "Noname"

☐ TEST Second library with argument

Full Name: Test.Second library with argument

Start / End / Elapsed: 20180603 23:22:02.492 / 20180603 23:22:02.493 / 00:00:00.001

Status: **PASS** (critical)

☐ KEYWORD HelloWorld.Say Hi2 somkiat

Start / End / Elapsed: 20180603 23:22:02.492 / 20180603 23:22:02.493 / 00:00:00.001

23:22:02.492 INFO Say hi somkiat

Name = "somkiat"

☐ TEST Third library

Full Name: Test.Third library

Start / End / Elapsed: 20180603 23:22:02.493 / 20180603 23:22:02.494 / 00:00:00.001

Status: **PASS** (critical)

☐ KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:22:02.494 / 20180603 23:22:02.494 / 00:00:00.000

23:22:02.494 INFO Say hi Noname

Name = "Noname"



Using TEST SUITE scope

```
1  class HelloWorld:
2      ROBOT_LIBRARY_SCOPE = 'TEST SUITE'
3
4      def __init__(self):
5          self.name = "Noname"
6
7      def say_hi(self):
8          print("Say hi " + self.name)
9
10     def say_hi2(self, name):
11         self.name = name
12         print("Say hi " + self.name)
```



Run with python path

- TEST First library

Full Name: Test.First library

Start / End / Elapsed: 20180603 23:16:43.825 / 20180603 23:16:43.826 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:16:43.825 / 20180603 23:16:43.826 / 00:00:00.001
23:16:43.826 INFO Say hi Noname

Name = "Noname"

- TEST Second library with argument

Full Name: Test.Second library with argument

Start / End / Elapsed: 20180603 23:16:43.826 / 20180603 23:16:43.827 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi2 somkiat

Start / End / Elapsed: 20180603 23:16:43.827 / 20180603 23:16:43.827 / 00:00:00.000
23:16:43.827 INFO Say hi somkiat

Name = "somkiat"

- TEST Third library

Full Name: Test.Third library

Start / End / Elapsed: 20180603 23:16:43.828 / 20180603 23:16:43.829 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:16:43.828 / 20180603 23:16:43.828 / 00:00:00.000
23:16:43.828 INFO Say hi somkiat

Name = "somkiat"



Using GLOBAL scope

```
1  class HelloWorld:
2      ROBOT_LIBRARY_SCOPE = 'GLOBAL'
3
4      def __init__(self):
5          self.name = "Noname"
6
7      def say_hi(self):
8          print("Say hi " + self.name)
9
10     def say_hi2(self, name):
11         self.name = name
12         print("Say hi " + self.name)
13
```



Run with python path

- TEST First library

Full Name: Test.First library

Start / End / Elapsed: 20180603 23:16:43.825 / 20180603 23:16:43.826 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:16:43.825 / 20180603 23:16:43.826 / 00:00:00.001
23:16:43.826 INFO Say hi Noname

Name = "Noname"

- TEST Second library with argument

Full Name: Test.Second library with argument

Start / End / Elapsed: 20180603 23:16:43.826 / 20180603 23:16:43.827 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi2 somkiat

Start / End / Elapsed: 20180603 23:16:43.827 / 20180603 23:16:43.827 / 00:00:00.000
23:16:43.827 INFO Say hi somkiat

Name = "somkiat"

- TEST Third library

Full Name: Test.Third library

Start / End / Elapsed: 20180603 23:16:43.828 / 20180603 23:16:43.829 / 00:00:00.001

Status: **PASS** (critical)

- KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:16:43.828 / 20180603 23:16:43.828 / 00:00:00.000
23:16:43.828 INFO Say hi somkiat

Name = "somkiat"



Run with another test suite

Create new test suite => test2.robot

```
1  *** Settings ***  
2  Library      HelloWorld  
3  
4  *** Testcases ***  
5  Another test case  
6      Say Hi
```



Run with python path

\$pybot --pythonpath *.robot

– SUITE Test2

Full Name:	Test & Test2.Test2
Source:	/Users/somkiat/data/slide/robot-framework/adv
Start / End / Elapsed:	20180603 23:31:04.939 / 20180603 23:31:04.94
Status:	1 critical test, 1 passed, 0 failed 1 test total, 1 passed, 0 failed

TEST Another test case

Full Name:	Test & Test2.Test2.Another test case
Start / End / Elapsed:	20180603 23:31:04.941 / 20180603 23:31:04.941 / 00:00:00.000
Status:	PASS (critical)

KEYWORD HelloWorld.Say Hi

Start / End / Elapsed: 20180603 23:31:04.941 / 20180603 23:31:04.942

23:31:04.942 INFO Say hi somkiat

Name = "somkiat"



Publish Library



Publish Library

Git provider => Github
pypi.org



Publish Library with pypi.org

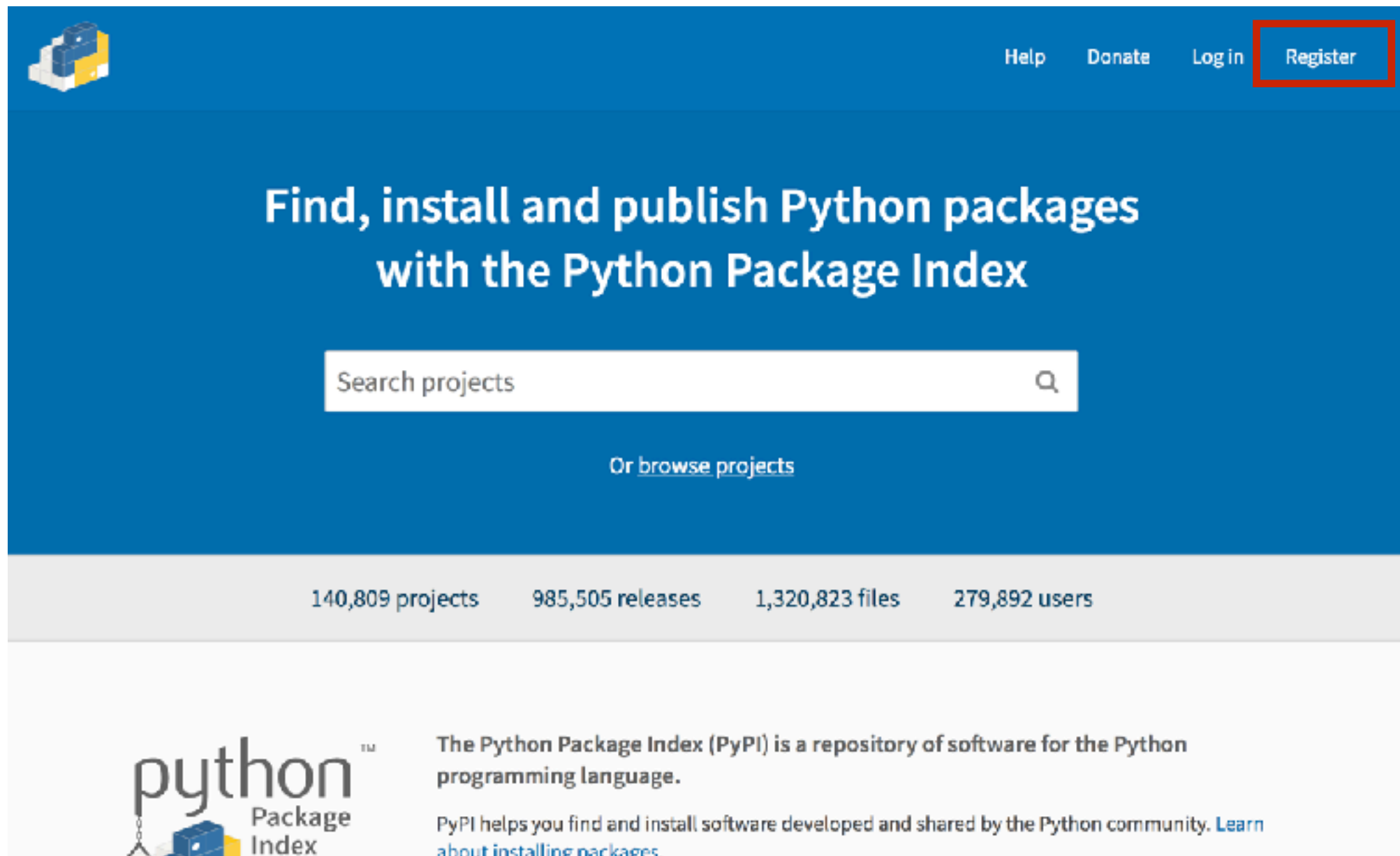


<https://packaging.python.org/guides/migrating-to-pypi-org/#uploading>



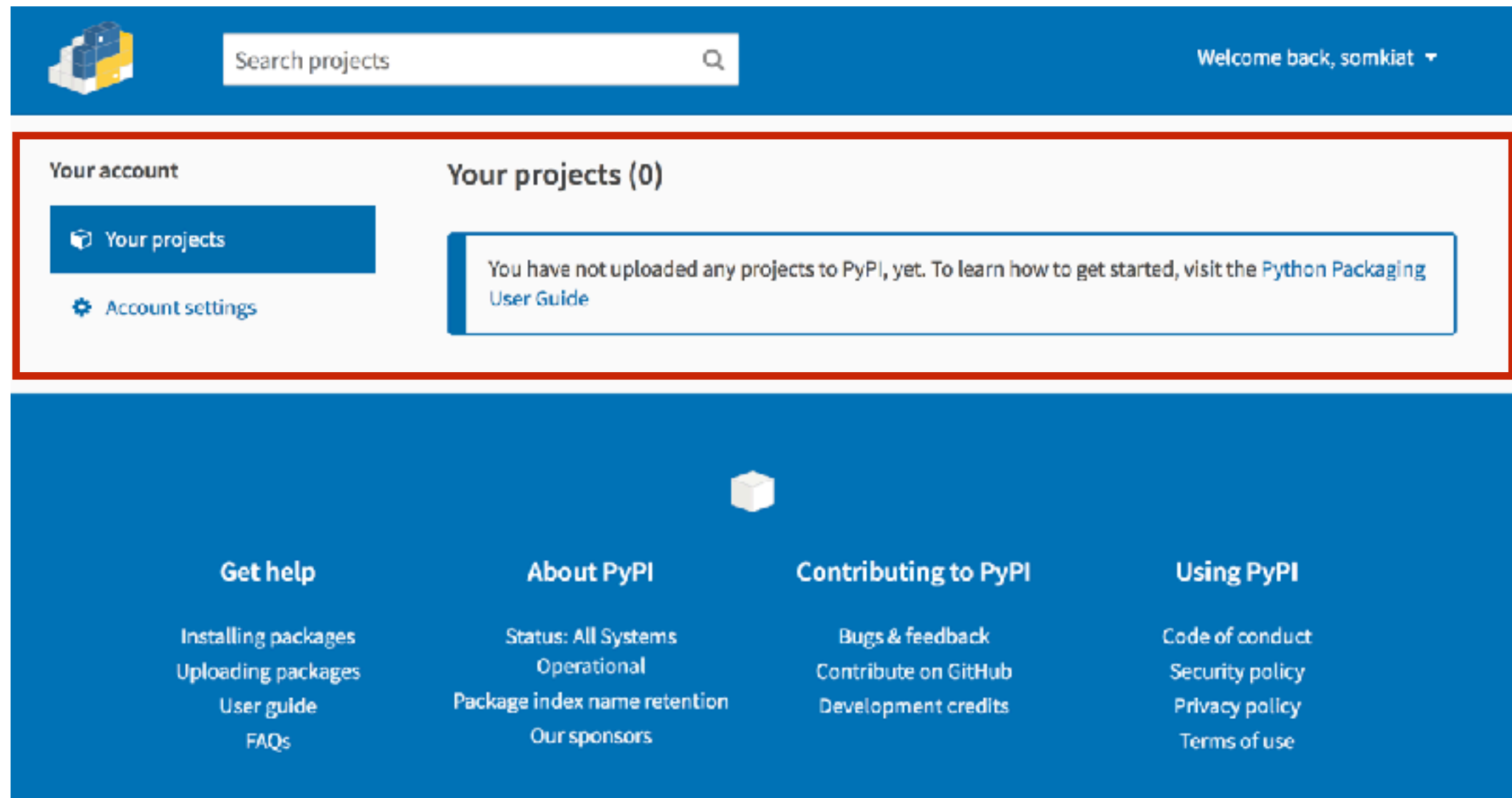
Step 1

Register account at <https://pypi.org/>



Step 2

Verify and see your project



The screenshot displays the PyPI user interface. At the top, there is a blue header bar with the PyPI logo on the left, a search bar labeled "Search projects" in the center, and a welcome message "Welcome back, somkiat" on the right. Below the header, the main content area is divided into two sections. On the left, under "Your account", there are two links: "Your projects" (highlighted with a blue background) and "Account settings". On the right, under "Your projects (0)", there is a message stating "You have not uploaded any projects to PyPI, yet. To learn how to get started, visit the [Python Packaging User Guide](#)". Below these sections, there is a large blue footer area with a white cube icon in the center. The footer is organized into four columns: "Get help" (with links for installing, uploading packages, user guide, and FAQs), "About PyPI" (with links for status, operational, package index name retention, and sponsors), "Contributing to PyPI" (with links for bugs & feedback, contribute on GitHub, and development credits), and "Using PyPI" (with links for code of conduct, security policy, privacy policy, and terms of use).



Step 3

Start to develop your package

```
— MANIFEST
— MANIFEST.in
— README.txt
— install.txt
— requirements.txt
— setup.py
— src
  — HelloWorld
    — HelloWorldKeywords.py
    — __init__.py
— test_suite.robot
```



Step 4 (1)

Create file **setup.py** to configure test library

```
— MANIFEST
— MANIFEST.in
— README.txt
— install.txt
— requirements.txt
— setup.py
— src
  — HelloWorld
    — HelloWorldKeywords.py
    — __init__.py
— test_suite.robot
```



Step 4 (2)

Specify name and version of library

```
1  from setuptools import setup
2
3  setup(
4      name="helloworld-library",
5      version='0.1',
6      package_dir={'': 'src'},
7      packages=['HelloWorld'],
8      url='https://github.com/up1/demo-helloworld-library',
9      author='Somkiat',
10     author_email='somkiat.p@gmail.com',
11 )
```



Step 4 (3)

Specify package structure and name

```
1  from setuptools import setup
2
3  setup(
4      name="helloworld-library",
5      version='0.1',
6      package_dir={'': 'src'},
7      packages=['HelloWorld'],
8      url='https://github.com/up1/demo-helloworld-library',
9      author='Somkiat',
10     author_email='somkiat.p@gmail.com',
11 )
```



Step 4 (4)

Required metadata of test library

```
1  from setuptools import setup
2
3  setup(
4      name="helloworld-library",
5      version='0.1',
6      package_dir={'': 'src'},
7      packages=['HelloWorld'],
8      url='https://github.com/up1/demo-helloworld-library',
9      author='Somkiat',
10     author_email='somkiat.p@gmail.com',
11 )
```



Step 5

Develop HelloWorld library



Structure of package

Create directory src/HelloWorld

```
— MANIFEST
— MANIFEST.in
— README.txt
— install.txt
— requirements.txt
— setup.py
— src
  └─ HelloWorld
      └─ HelloWorldKeywords.py
      └─ __init__.py
— test_suite.robot
```



Define keywords of library

Create file HelloWorldKeywords.py

```
1  class HelloWorldKeywords(object):
2      def __init__(self):
3          self.name = "Noname"
4
5      def say_hi(self):
6          print("Say hi " + self.name)
7
8      def say_hi2(self, name):
9          self.name = name
10         print("Say hi " + self.name)
11
```



Define keywords of library

Create file `__init__.py`

```
1  from HelloWorldKeywords import HelloWorldKeywords
2
3  class HelloWorld(HelloWorldKeywords):
4      ROBOT_LIBRARY_SCOPE = 'TEST_CASE'
5
```



Step 6

Publish library to pypi.org



Create file ~/.pypirc

Configuration for publish library to pypi.org

```
1  [distutils]
2  index-servers =
3      pypi
4
5  [pypi]
6  #repository=https://pypi.python.org/pypi
7  username=<your username>
8  password=<your password>
```



Publish library to pypi.org

`$pip install -U pip setuptools twine`

`$python setup.py sdist`

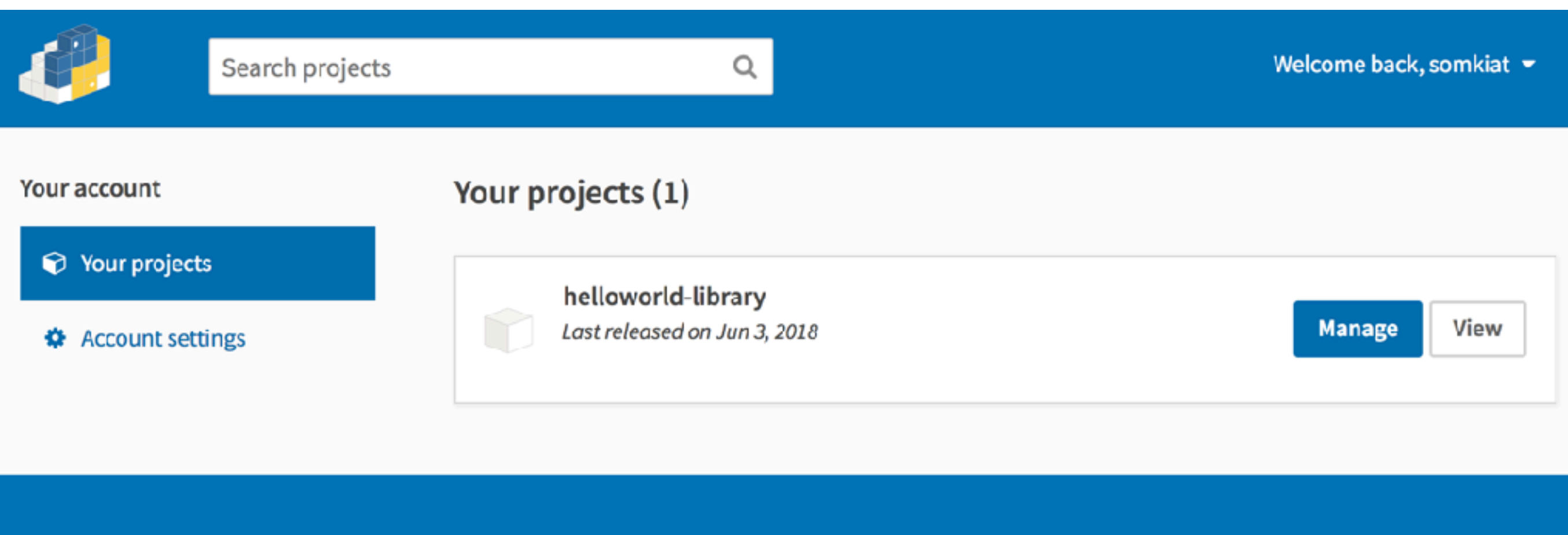
`$twine upload dist/*`

```
Writing helloworld-library-0.2/setup.cfg
Creating tar archive
removing 'helloworld-library-0.2' (and everything under it)
Uploading distributions to https://upload.pypi.org/legacy/
Uploading helloworld-library-0.2.tar.gz
100%|██████████| 3.54k/3.54k [00:01<00:00, 2.86kB/s]
```



Check your library (1)

Go to pypi.org





The screenshot displays the PyPI user interface. At the top, there is a blue header bar containing the PyPI logo (a stack of cubes), a search bar labeled "Search projects" with a magnifying glass icon, and a user greeting "Welcome back, somkiat" with a dropdown arrow. Below the header, the page is divided into two main sections. On the left, under the heading "Your account", there is a blue button labeled "Your projects" and a link labeled "Account settings" with a gear icon. On the right, under the heading "Your projects (1)", there is a single project entry for "helloworld-library". This entry includes a small cube icon, the project name, the text "Last released on Jun 3, 2018", and two buttons: "Manage" (in blue) and "View" (in white with a blue border).



Check your library (2)

helloworld-library 0.1

 Latest version


`pip install helloworld-library` 


Last released: About 5 hours ago.


No project description provided

Manage project


Navigation

 Project description

 Release history

 Download files

Project links

 Homepage

Project description

The author of this package has not provided a project description



Use HelloWorld library

\$pip install helloworld-library

```
Collecting helloworld-library
```

```
  Downloading https://files.pythonhosted.org/packages/8f/a2/92e220deb5cb908b1d6f358f1d36e8e8307e9211a8c382b91a0225/hello-world-library-0.2.tar.gz
```

```
Building wheels for collected packages: helloworld-library
```

```
  Running setup.py bdist_wheel for helloworld-library ... c
```

```
  Stored in directory: /Users/somkiat/Library/Caches/pip/wheels/b4/6b/db550e3f32243f1d2397f064d34ed13b3178cb7b90b29f4c5e
```

```
Successfully built helloworld-library
```

```
Installing collected packages: helloworld-library
```

```
Successfully installed helloworld-library-0.2
```



Use HelloWorld library

```
1    *** Settings ***
2    Library    HelloWorld
3
4    *** Testcases ***
5    First library
6        Say Hi
7
8    Second library with argument
9        Say Hi2    somkiat
10
```

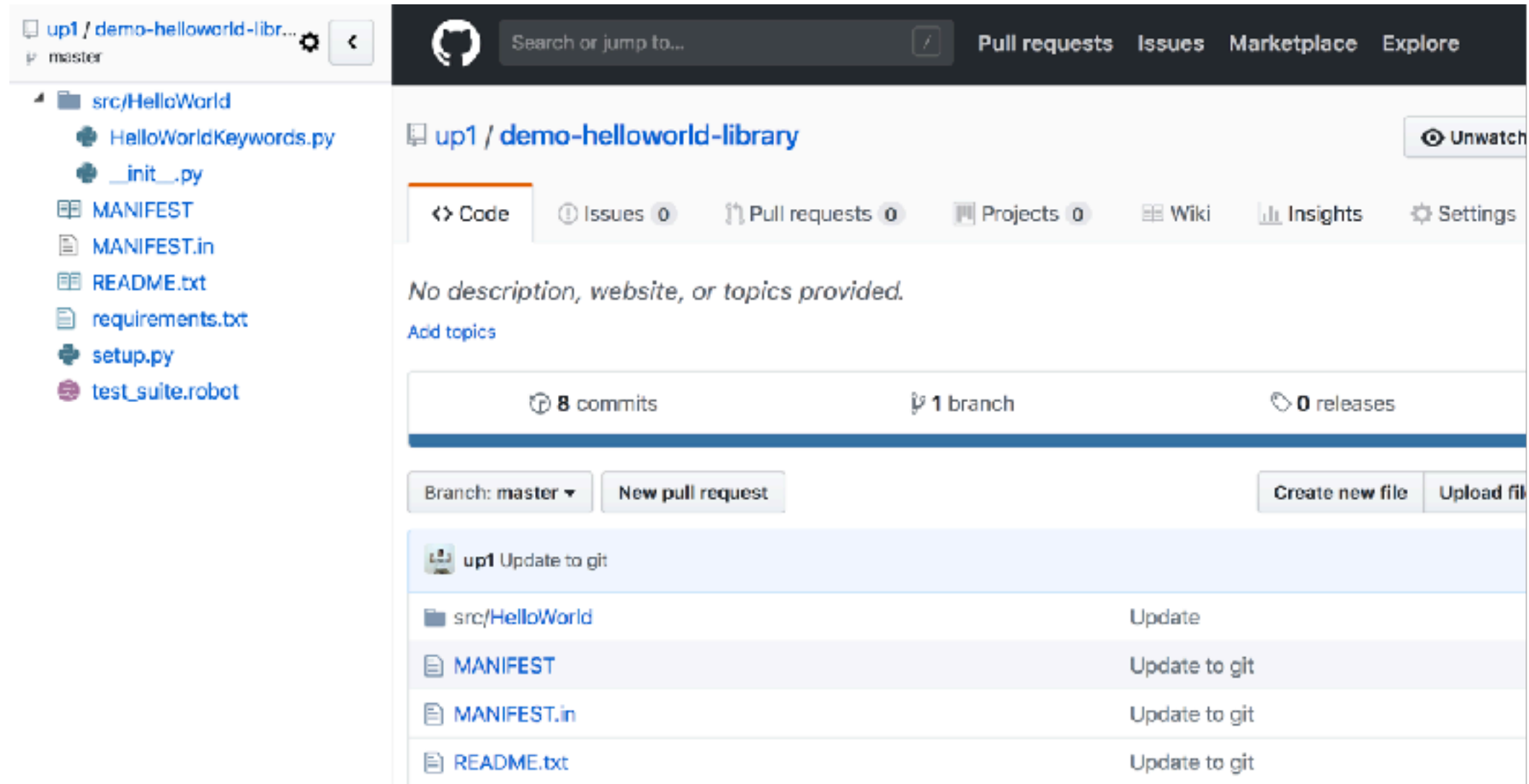


Publish Library with github



Publish library to github

1. Push your code to your Github repository



<https://github.com/up1/demo-helloworld-library>



2. Install library from Github (1)

```
$pip install -r requirements.txt
```

```
$pip uninstall -r requirements.txt
```

[git+https://github.com/up1/demo-helloworld-library.git#egg=helloworld-library](https://github.com/up1/demo-helloworld-library.git#egg=helloworld-library)

↑
Name of library



2. Install library from Github (2)

\$pip install -r requirements.txt

```
Collecting helloworld-library from git+https://github.com/up1/library.git#egg=helloworld-library (from -r requirements.txt)
  Cloning https://github.com/up1/demo-helloworld-library.git to /tmp/pip-install-d20doky/helloworld-library
  olders/t5/8kg23s_97z9dw44tfc1d6dqw0000gn/T/pip-install-d20doky/helloworld-library
```

```
Building wheels for collected packages: helloworld-library
  Running setup.py bdist_wheel for helloworld-library ... done
  Stored in directory: /private/var/folders/t5/8kg23s_97z9dw44tfc1d6dqw0000gn/T/pip-ephem-wheel-cache-svb7x4pk/wheels/6e/77/72/2c1098f915d8e47d24d0c0f106fe5b667
```

```
Successfully built helloworld-library
Installing collected packages: helloworld-library
Successfully installed helloworld-library-0.2
```



How to generate document of test library ?

<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#specifying-documentation-format>



Generate document of library

Robotframework 2.7.5+ use **Libdoc** to generate the documentation of library

<http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#libdoc>



Support formats

ROBOT (default)

HTML

TEXT (plain text)

reST (reStructuredText)



How to use ?

Example with ROBOT format

```
1  from HelloWorldKeywords import HelloWorldKeywords
2
3  class HelloWorld(HelloWorldKeywords):
4
5      """ A keyword library for Robot Framework. It provides keywords for
6          learning how to create a new library. For more information
7          on underlying methods and documentation, see:
8              http://eclipse.org/paho/clients/python/docs/
9          """
10
11     ROBOT_LIBRARY_SCOPE = 'TEST_CASE'
12
```



How to use ?

Document in each keyword

```
5     def say_hi(self):
6         """ Say hi with out argument
7         Examples:
8         | Say Hi |
9         """
10        print("Say hi " + self.name)
11
12    def say_hi2(self, name):
13        """ Say hi with a argument.
14        `name` Your name
15        Examples:
16        Say hi    <name>
17        | Say Hi | somkiat |
18        """
19        self.name = name
20        print("Say hi " + self.name)
```



Generate documentation

```
$pip install -U helloworld-library
```

```
$python -m robot.libdoc HelloWorld ./docs/  
HelloWorld-Library.html
```



Documentation of Library (1)

HelloWorld

Library scope: test case
Named arguments: supported

Introduction

A keyword library for Robot Framework. It provides keywords for learning how to create a new library. For more information on underlying

Shortcuts

Say Hi - Say Hi2

Keywords

Keyword	Arguments	
Say Hi		Say hi with out argument Examples: <div>Say Hi</div>
Say Hi2	<i>name</i>	Say hi with a argument. <i>name</i> Your name Examples <div>Say Hi somkiat</div>

Altogether 2 keywords.
Generated by [Libdoc](#) on 2018-06-04 00:34:08.



Documentation of Library (2)

HelloWorld

Library scope: test case
Named arguments: supported

Introduction

A keyword library for Robot Framework. It provides keywords for learning how to create a new library. For more information on underlying

Shortcuts

Say Hi - Say Hi2

Keywords

Keyword	Arguments	
Say Hi		Say hi with out argument Examples: <div>Say Hi</div>
Say Hi2	<i>name</i>	Say hi with a argument. <i>name</i> Your name Examples <div>Say Hi somkiat</div>

Altogether 2 keywords.
Generated by [Libdoc](#) on 2018-06-04 00:34:08.



Create keywords of test library

