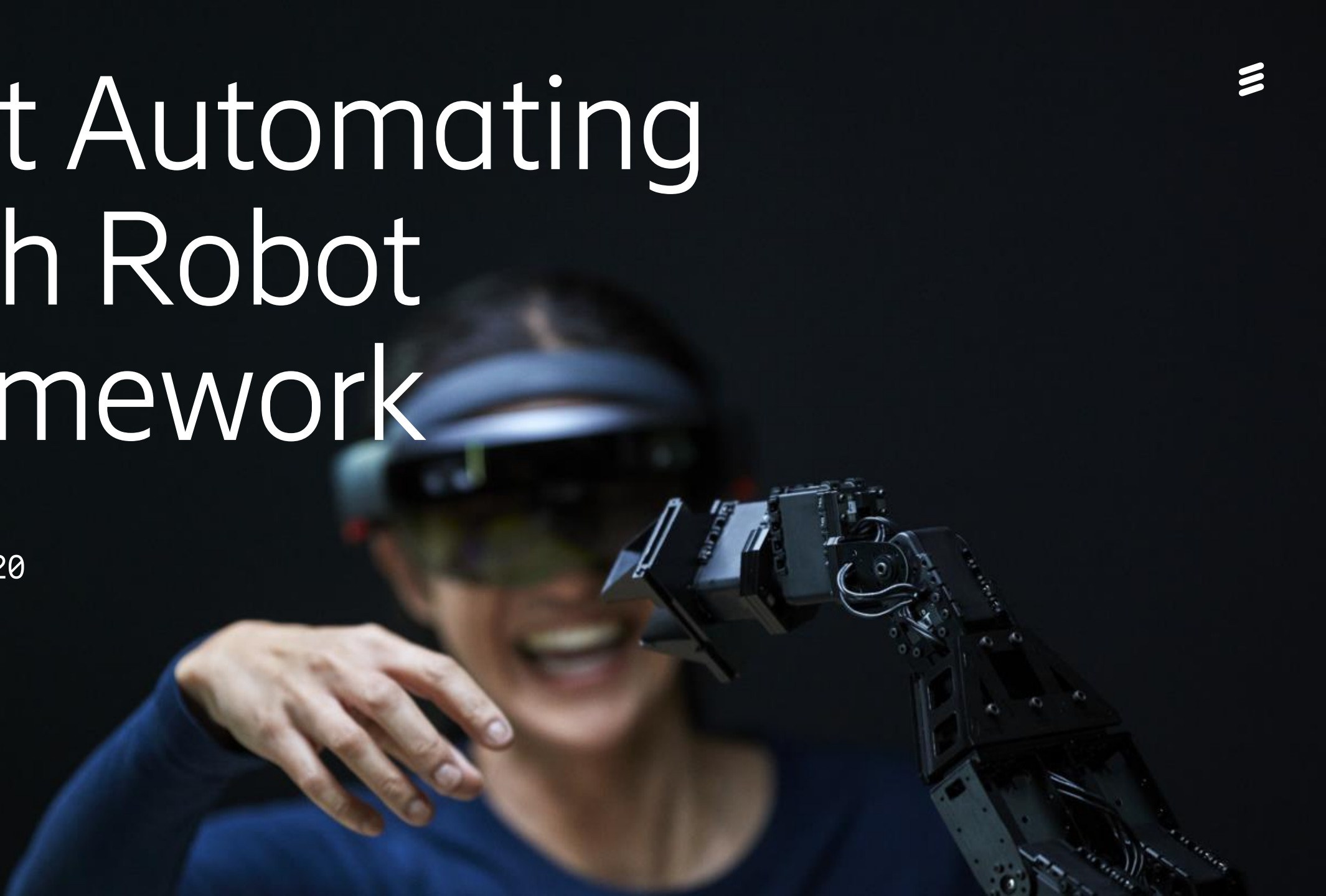


Test Automating With Robot Framework



October 2, 2020

Ge Liu
Amir Ingher



Agenda

12:00 – 12:20 Basics of Robot Framework

12:20 – 12:30 CI/CD with Robot Framework

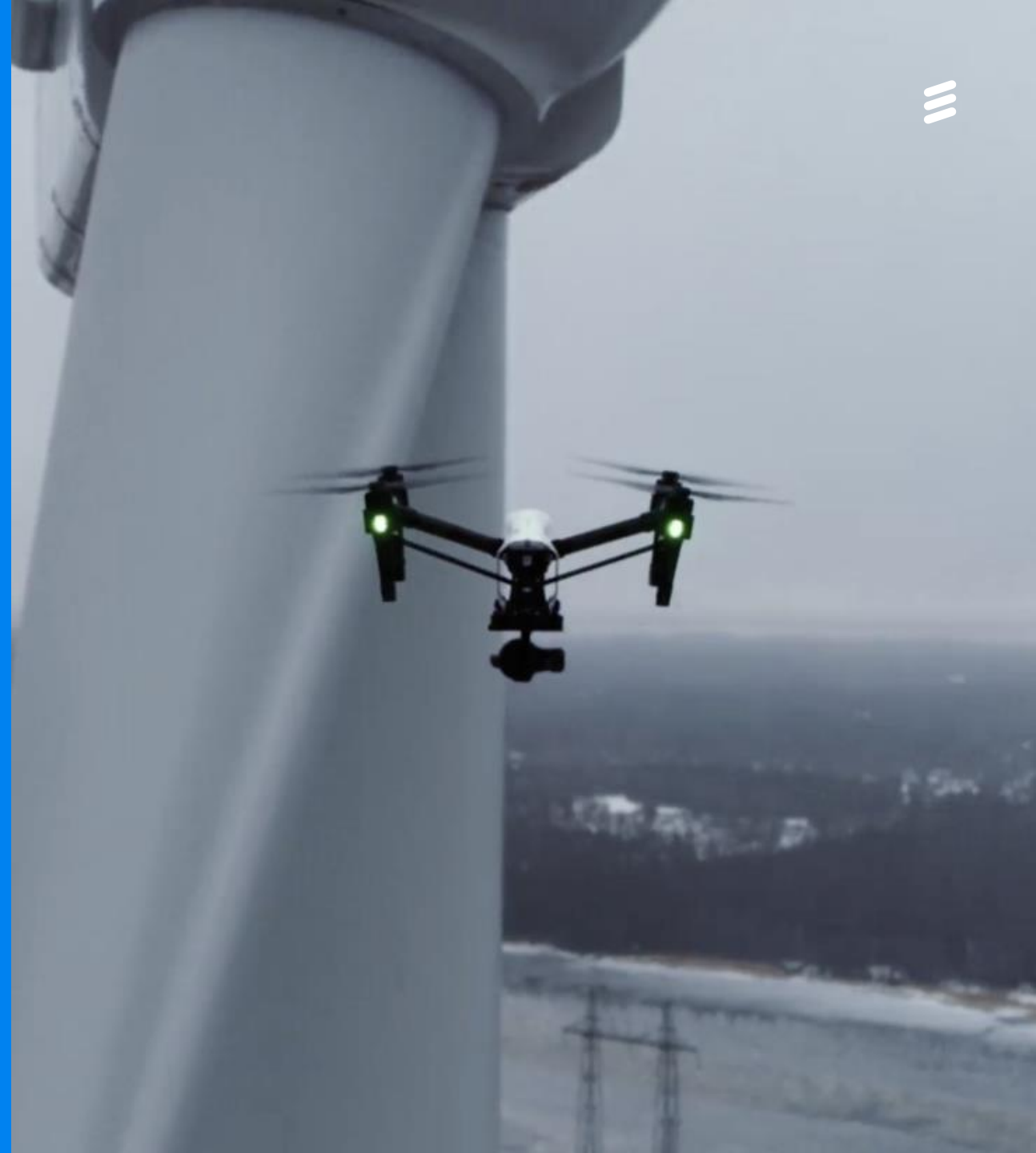
12:30 – 12:50 Device Automation with Appium

12:50 – 13.00 Q&A

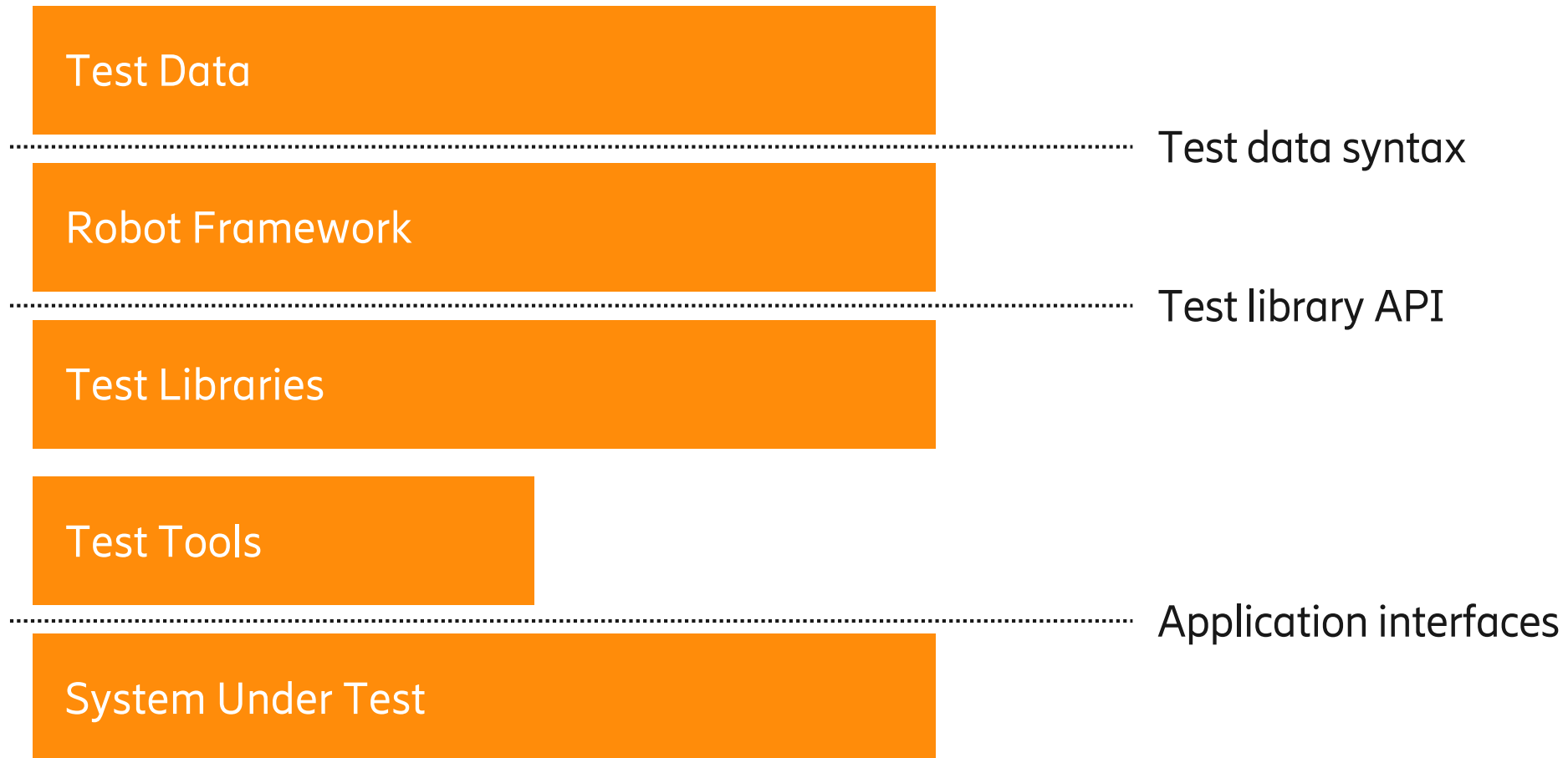


Robot Framework

- Robot Framework is a Data driven acceptance test automation framework
- Started as a Master Thesis sponsored by Nokia Networks back in 2005, became Open Source in 2008
- Implemented with Python, it also has a Java-based implementation
- Widely used in the industries
- Well integrated with common CI framework (e.g. Jenkins)



High level architecture



Basic Structure



- Normally A Robot test suite contains these sections:
 - Settings
 - Variables
 - Test Cases
 - Keywords

```
*** Settings ***
Force Tags    if-cases
Library      Selenium2Library

*** Variables ***
${BROWSER}   GC
${USERNAME}  demo
${PASSWORD}  mode
${CONFITION} ${True}

*** Test Cases ***
Demo how to use a loop in Robot cases
  Run Keyword If    ${CONFITION} == ${True}    Login multiple times    3

*** Keywords ***
Login multiple times
  [Arguments]    ${loop_count}
  FOR    ${index}    IN RANGE    0    ${loop_count}
    Open Login Page
    Login to Portal    ${USERNAME}    ${PASSWORD}
  END

Open Login Page
  Open Browser    http://localhost:7272    ${BROWSER}
  Page should contain    Login Page

Login to Portal
  [Arguments]    ${username}    ${password}
  [Teardown]    Close Browser
  Input Text    username_field    ${username}
  Input Text    password_field    ${password}

  Click Element    login_button
  Sleep    5
```

Keyword-Driven



- Robot is a simple keyword-driven test framework

```
*** Test Cases ***
Login with valid username and password
Open Browser      http://localhost:7272
Input Text        username_field      demo
Input Text        password_field     mode
Click Element     login_button
Page Should Contain Welcome Page
Page Should Contain Login succeeded. Now you can logout.
Close Browser
```

Data-Driven



- Robot Framework is data-driven.

*** Test Cases ***

```
Login with valid username and empty password  
Login with empty username and empty password  
Login with valid username and invalid password
```

USERNAME

```
${username}  
${EMPTY}  
${username}
```

PASSWORD

```
${EMPTY}  
${EMPTY}  
${invalid_password}
```

- Variables can be defined in resource files, python formatted variable files, or in command line

```
app_url= "http://localhost:7272"  
username= "demo"  
password= "mode"  
invalid_password= "dummy"  
  
Welcome_text_line1= "Welcome Page"  
Welcome_text_line2= "Login succeeded. Now you can logout."  
  
Error_text_line1= "Error Page"  
Error_text_line2= "Login failed. Invalid user name and/or password."
```

```
-V username:myname
```

Behavior Driven Development (Gherkin Syntax)



***** Test Cases *****

Valid Login

Given browser is opened to login page

When user "demo" logs in with password "mode"

Then welcome page should be open

Tagging

- Tags can be used to make the execution simpler

```
*** Settings ***
Force Tags    if-cases
Library      Selenium2Library

*** Variables ***
${BROWSER}   GC
${USERNAME}  demo
${PASSWORD}  mode
${CONFITION} ${True}

*** Test Cases ***
Demo how to use a loop in Robot cases
  Run Keyword If    ${CONFITION} == ${True}    Login multiple times    3

*** Keywords ***
Login multiple times
  [Arguments]    ${loop_count}
  FOR    ${index}    IN RANGE    0    ${loop_count}
    Open Login Page
    Login to Portal    ${USERNAME}    ${PASSWORD}
  END

Open Login Page
  Open Browser    http://localhost:7272    ${BROWSER}
  Page should contain    Login Page

Login to Portal
  [Arguments]    ${username}    ${password}
  [Teardown]    Close Browser
  Input Text    username_field    ${username}
  Input Text    password_field    ${password}

  Click Element    login_button
  Sleep    5
```

Use Python Function As Keyword



- Python function can be used directly in Robot test cases directly

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time

def open_browser_and_login():

    browser = webdriver.Chrome(executable_path="C:\\Python27\\chromedriver.exe")
    browser.get("http://localhost:7272")
    fname = browser.find_element_by_id('username_field')
    fname.clear()
    fname.send_keys('demo')
    lname = browser.find_element_by_id('password_field')
    lname.clear()
    lname.send_keys('mode')
    submit_button = browser.find_element_by_css_selector('input[type="submit"]')
    submit_button.send_keys(Keys.ENTER)
    time.sleep(2)
    browser.close()
```

```
*** Settings ***
Library      selenium_example.py

*** Variables ***
${BROWSER}   GC
${USERNAME}  demo
${PASSWORD}  mode
${CONFITION} ${True}

*** Test Cases ***
Demo how to use a loop in Robot cases
    open browser and login
```

Test Report



Robot Cases	1	1	0
Robot Cases.If Condition	1	1	0
Robot Cases.If Condition.Keywords	1	1	0

Test Execution Log

SUITE Robot Cases

Full Name: Robot Cases
Source: C:\robotdemo\robotdemo\robot_cases
Start / End / Elapsed: 20200920 21:14:16.103 / 20200920 21:14:59.327 / 00:00:43.224
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

SUITE If Condition

Full Name: Robot Cases.If Condition
Source: C:\robotdemo\robotdemo\robot_cases\if_condition
Start / End / Elapsed: 20200920 21:14:16.165 / 20200920 21:14:59.315 / 00:00:43.150
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

SUITE Keywords

Full Name: Robot Cases.If Condition.Keywords
Source: C:\robotdemo\robotdemo\robot_cases\if_condition\keywords.robot
Start / End / Elapsed: 20200920 21:14:16.177 / 20200920 21:14:59.311 / 00:00:43.134
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

TEST Demo how to use a loop in Robot cases

Full Name: Robot Cases.If Condition.Keywords.Demo how to use a loop in R
Tags: if-cases
Start / End / Elapsed: 20200920 21:14:16.494 / 20200920 21:14:59.308 / 00:00:42.814
Status: **PASS** (critical)

KEYWORD BuiltIn.Run Keyword If \${CONDITION} == \${True}, Login multiple times, 3

Documentation: Runs the given keyword with the given arguments, if condition is true.
Start / End / Elapsed: 20200920 21:14:16.496 / 20200920 21:14:59.307 / 00:00:42.811

KEYWORD Login multiple times 3

Start / End / Elapsed: 20200920 21:14:16.497 / 20200920 21:14:59.307 / 00:00:42.810

FOR \${index} IN RANGE [0 | \${loop_count}]

Start / End / Elapsed: 20200920 21:14:16.498 / 20200920 21:14:59.307 / 00:00:42.809

FOR \${index} = 0

Cases Report

Information

All tests passed
20200920 21:14:16.103
20200920 21:14:59.327
00:00:43.224
[log.html](#)

Statistics

Total Statistics	Total	Pass	Fail	Elapsed
	1	1	0	00:00:43.224
	1	1	0	00:00:43.224

Statistics by Tag	Total	Pass	Fail	Elapsed
	1	1	0	00:00:43.224

Statistics by Suite	Total	Pass	Fail	Elapsed
Condition	1	1	0	00:00:43.150
Condition Keywords	1	1	0	00:00:43.134

Filters

Tags Suites Search

Critical Tests
 All Tests

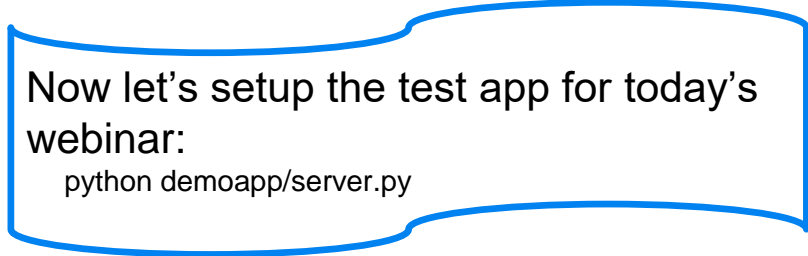
- Robot has pretty good test report and execution log

Libraries



- Link: <http://robotframework.org/#libraries>
- Including Standard and External
- Standard libraries will be installed when You install the Robot Framework
- External needs to be installed separately

Demo Time



Now let's setup the test app for today's webinar:
`python demoapp/server.py`



- Demo case 1: My First Case

```
python3 -m robot my_first_case.robot
```

```
python3 -m robot my_first_case_with_keywords.robot
```

- Demo case 2: How to use template

```
python3 -m robot -v BROWSER:gc -V ..\variables\variables.py use_template.robot
```

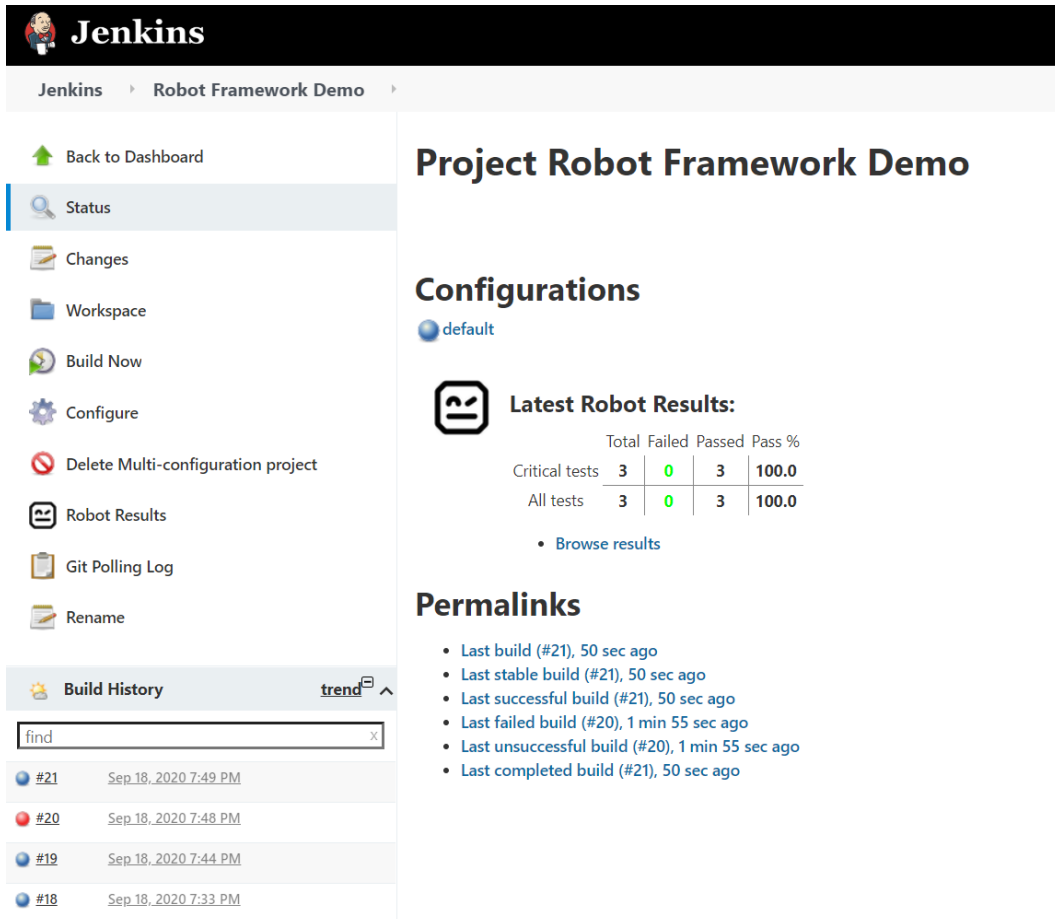
- Demo case 3: How to use if conditions and loop

```
python3 -m robot keywords.robot
```

- Demo case 4: How to use Python function as keywords

```
python3 -m robot use_python_function.robot
```

Continuous Integration with Robot Framework



Jenkins

Jenkins > Robot Framework Demo

Back to Dashboard

Status

Changes

Workspace

Build Now

Configure

Delete Multi-configuration project

Robot Results

Git Polling Log

Rename

Project Robot Framework Demo

Configurations

default

Latest Robot Results:

	Total	Failed	Passed	Pass %
Critical tests	3	0	3	100.0
All tests	3	0	3	100.0

[Browse results](#)

Permalinks

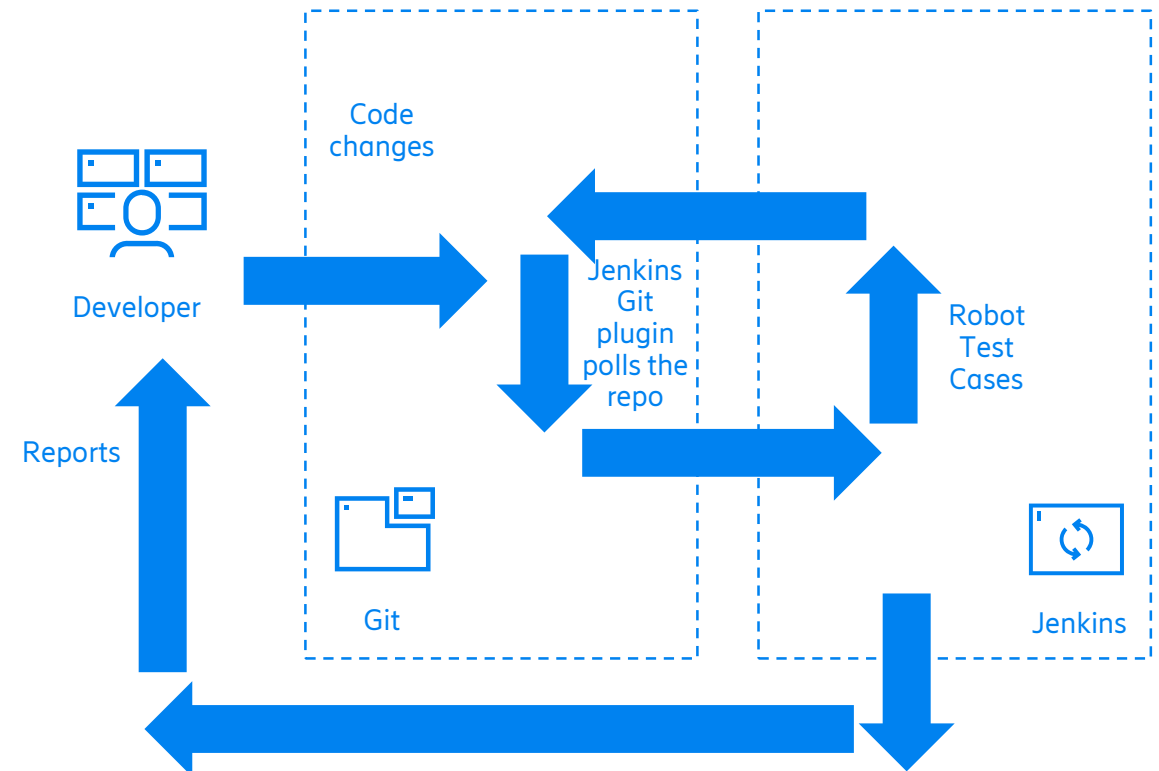
- Last build (#21), 50 sec ago
- Last stable build (#21), 50 sec ago
- Last successful build (#21), 50 sec ago
- Last failed build (#20), 1 min 55 sec ago
- Last unsuccessful build (#20), 1 min 55 sec ago
- Last completed build (#21), 50 sec ago

Build History

find

#21	Sep.18., 2020 7:49 PM
#20	Sep.18., 2020 7:48 PM
#19	Sep.18., 2020 7:44 PM
#18	Sep.18., 2020 7:33 PM

- Robot Framework works very well in Jenkins, which You can use it for Continuous Integration purposes.





<https://www.ericsson.com/>