An introduction to

Robot Framework

http://robotframework.org
Fast facts

• Generic test automation framework
  - Utilizes the keyword-driven testing approach
  - Suitable for both “normal” test automation and ATDD

• Implemented with Python
  - Runs also on Jython (JVM) and IronPython (.NET)
  - Can be extended natively using Python or Java
  - Other languages supported via a remote interface

• Open source
  - Apache 2.0 license
  - Sponsored by Nokia Siemens Networks
  - Active development and growing community
High level architecture

- Test Data
  - Test data syntax
- Robot Framework
  - Test library API
- Test Libraries
- Test Tools
  - Application interfaces
- System Under Test
Simple tabular syntax

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Action</th>
<th>Argument</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>User can create an account and log in</td>
<td>Create Valid User</td>
<td>fred</td>
<td>P4ssw0rd</td>
</tr>
<tr>
<td></td>
<td>Attempt to Login with Credentials</td>
<td>fred</td>
<td>P4ssw0rd</td>
</tr>
<tr>
<td></td>
<td>Status Should Be</td>
<td>Logged In</td>
<td></td>
</tr>
<tr>
<td>User cannot log in with bad password</td>
<td>Create Valid User</td>
<td>betty</td>
<td>P4ssw0rd</td>
</tr>
<tr>
<td></td>
<td>Attempt to Login with Credentials</td>
<td>betty</td>
<td>wrong</td>
</tr>
<tr>
<td></td>
<td>Status Should Be</td>
<td>Access</td>
<td>Denied</td>
</tr>
</tbody>
</table>
# Data-driven test cases

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Action</th>
<th>Password</th>
<th>Expected error message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Too short password</td>
<td>Creating user with invalid password should fail</td>
<td>abCD5</td>
<td>${PWD INVALID LENGTH}</td>
</tr>
<tr>
<td>Too long password</td>
<td>Creating user with invalid password should fail</td>
<td>abCD567890123</td>
<td>${PWD INVALID LENGTH}</td>
</tr>
<tr>
<td>Password without lowercase letters</td>
<td>Creating user with invalid password should fail</td>
<td>123DEFG</td>
<td>${PWD INVALID LENGTH}</td>
</tr>
<tr>
<td>Password without capital letters</td>
<td>Creating user with invalid password should fail</td>
<td>abcd56789</td>
<td>${PWD INVALID CONTENT}</td>
</tr>
<tr>
<td>Password without numbers</td>
<td>Creating user with invalid password should fail</td>
<td>AbCdEfGh</td>
<td>${PWD INVALID CONTENT}</td>
</tr>
<tr>
<td>Password with special characters</td>
<td>Creating user with invalid password should fail</td>
<td>abCD56+</td>
<td>${PWD INVALID CONTENT}</td>
</tr>
</tbody>
</table>
## Executable specifications

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>User can change password</td>
<td>Given a user has a valid account</td>
</tr>
<tr>
<td></td>
<td>when she changes her password</td>
</tr>
<tr>
<td></td>
<td>then she can log in with the new password</td>
</tr>
<tr>
<td></td>
<td>and she cannot use the old password anymore</td>
</tr>
</tbody>
</table>
## Higher level keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Action</th>
<th>Argument</th>
<th>Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear login database</td>
<td>Remove file</td>
<td>${DATABASE FILE}</td>
<td></td>
</tr>
<tr>
<td>Create valid user</td>
<td>[Arguments]</td>
<td>${username}</td>
<td>${password}</td>
</tr>
<tr>
<td>Create user</td>
<td>${username}</td>
<td>${password}</td>
<td></td>
</tr>
<tr>
<td>Status should be</td>
<td>SUCCESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creating user with invalid password should fail</td>
<td>[Arguments]</td>
<td>${password}</td>
<td>${error}</td>
</tr>
<tr>
<td>Create user</td>
<td>example</td>
<td></td>
<td>${password}</td>
</tr>
<tr>
<td>Status should be</td>
<td>Creating user failed:</td>
<td></td>
<td>Creating user failed:</td>
</tr>
<tr>
<td></td>
<td>${error}</td>
<td></td>
<td>${error}</td>
</tr>
<tr>
<td>Login</td>
<td>[Arguments]</td>
<td>${username}</td>
<td>${password}</td>
</tr>
<tr>
<td></td>
<td>Attempt to login with credentials</td>
<td>${username}</td>
<td>${password}</td>
</tr>
<tr>
<td></td>
<td>Status should be</td>
<td>Logged In</td>
<td></td>
</tr>
</tbody>
</table>
import os
import sys

class LoginLibrary:

    def __init__(self):
        sut_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), '...', 'sut', 'login.py')
        self._command_prefix = '%s %s' % (sys.executable, sut_path)
        self._status = ''

    def create_user(self, username, password):
        self._run_command('create', username, password)

    def change_password(self, username, old_pwd, new_pwd):
        self._run_command('change-password', username, old_pwd, new_pwd)

    def attempt_to_login_with_credentials(self, username, password):
        self._run_command('login', username, password)

    def status_should_be(self, expected_status):
        if expected_status != self._status:
            raise AssertionError("Expected status to be '%s' but was '%s'" % (expected_status, self._status))

    def _run_command(self, command, *args):
        command = '%s %s %s' % (self._command_prefix, command, ' '.join(args))
        process = os.popen(command)
        self._status = process.read().strip()
        process.close()
## Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>${USERNAME}</td>
<td>janedoe</td>
</tr>
<tr>
<td>${PASSWORD}</td>
<td>J4n3D0e</td>
</tr>
<tr>
<td>${NEW PASSWORD}</td>
<td>e0D3n4J</td>
</tr>
<tr>
<td>${DATABASE FILE}</td>
<td>${TEMPDIR}${/}robotframework-quickstart-db.txt</td>
</tr>
<tr>
<td>${PWD INVALID LENGTH}</td>
<td>Password must be 7-12 characters long</td>
</tr>
<tr>
<td>${PWD INVALID CONTENT}</td>
<td>Password must be a combination of lowercase and uppercase letters and numbers</td>
</tr>
</tbody>
</table>

### From command line:

```
$ pybot --variable USERNAME:johndoe --variable PASSWORD:J0hnD0e my_tests.html
```
Tagging

- Free metadata to categorize test cases
- Statistics by tags collected automatically
- Select test cases to be executed
- Specify which test cases are considered critical
Clear reports

### Login Tests Test Report

**Summary Information**

- **Status:** 6 critical tests failed
- **Start Time:** 20080613 14:12:08.445
- **End Time:** 20080613 14:12:39.666
- **Elapsed Time:** 00:00:31.221

#### Test Statistics

<table>
<thead>
<tr>
<th>Total Statistics</th>
<th>Total</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Tests</td>
<td>10</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>All Tests</td>
<td>10</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Statistics by Tag

<table>
<thead>
<tr>
<th>Statistics by Tag</th>
<th>Total</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>regression</td>
<td>10</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>smoke</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Statistics by Suite

**Login Tests**

<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login Tests</td>
<td>10</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>L. Invalid Login</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>L. Simple Login</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**L. Higher Level Login**

<table>
<thead>
<tr>
<th>Name</th>
<th>Total</th>
<th>Pass</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>L. Higher Level Valid Login</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

### Test Details by Suite

- **Login Tests**
  - **Documented:** Demo test cases for Robot Framework using Selenium test library
  - **Passed:** 16 critical tests, 13 passed, 0 failed
  - **Failed:** 16 tests total, 10 passed, 6 failed
  - **Started:** 20080613 13:38:36
  - **Elapsed:** 00:00:32

- **L. Invalid Login**
  - **Passed:** 3 critical tests, 3 passed, 0 failed
  - **Failed:** 3 tests total, 3 passed, 0 failed
  - **Started:** 20080613 13:38:36
  - **Elapsed:** 00:00:17

- **L. Higher Level Login**
  - **Passed:** 3 critical tests, 3 passed, 0 failed
  - **Failed:** 3 tests total, 3 passed, 0 failed
  - **Started:** 20080613 13:38:36
  - **Elapsed:** 00:00:05

- **L. Invalid Username**
  - **Passed:** 6 critical tests, 6 passed, 0 failed
  - **Failed:** 6 tests total, 6 passed, 0 failed
  - **Started:** 20080613 13:38:36
  - **Elapsed:** 00:00:10

- **L. Invalid Password**
  - **Passed:** 6 critical tests, 6 passed, 0 failed
  - **Failed:** 6 tests total, 6 passed, 0 failed
  - **Started:** 20080613 13:38:36
  - **Elapsed:** 00:00:01
### Detailed logs

#### TEST SUITE: Higher Level Login

- **Full Name:** Login Tests.Higher Level Login
- **Source:** /home/jth/workspace/seleniumlib/demo/login_tests/higher_level_login.html
- **Start / End / Elapsed:** 2009/04/15 07:36:29.500 / 2009/04/15 07:36:55.480 / 00:00:25.980
- **Overall Status:** FAIL
- **Message:** 3 critical tests, 0 passed, 3 failed
  3 tests total, 0 passed, 3 failed

#### TEST CASE: Higher Level Valid Login

- **Full Name:** Login Tests.Higher Level Login.Higher Level Valid Login
- **Tags:** regression, smoke
- **Start / End / Elapsed:** 2009/04/15 07:36:29.533 / 2009/04/15 07:36:36.412 / 00:00:06.879
- **Status:** FAIL (critical)
- **Message:** Location should have been 'http://localhost:7272/welcome.html' but was 'http://localhost:7272/error.html'

- **SETUP:** `resource.Open Login Page`
- **KEYWORD:** `resource.Input Username` demo
- **KEYWORD:** `resource.Input Password` mode
- **KEYWORD:** `resource.Click Login Button`

- **KEYWORD:** `resource.Welcome Page Should Be Open`

- **Start / End / Elapsed:** 2009/04/15 07:36:36.340 / 2009/04/15 07:36:36.374 / 00:00:00.034

- **KEYWORD:** `SeleniumLibrary.Location Should Be` `{WELCOME URL}`
  - **Documentation:** Verifies that current URL is exactly `url`.
  - **Start / End / Elapsed:** 2009/04/15 07:36:36.341 / 2009/04/15 07:36:36.374 / 00:00:00.033

- **INFO:** Verifying current location is 'http://localhost:7272/welcome.html'.
  - **FAIL:** Location should have been 'http://localhost:7272/welcome.html' but was 'http://localhost:7272/error.html'

- **TEARDOWN:** `SeleniumLibrary.Close Browser`
Easy integration

• Test suites are created from files and directories
  – Trivial to store into any version control system
• Simple command line interface
  – Easy to start test execution from CI systems or other tools
• Output also in XML format
  – All information in machine readable format
  – Outputs from different test runs can be combined
• Plugins for common CI and build tools
  – Jenkins/Hudson, Ant, Maven
Different test libraries

- **Standard libraries**
  - Included in normal installation
  - OperatingSystem, Telnet, Screenshot, String, …

- **External libraries**
  - Must be installed separately
  - SeleniumLibrary, SwingLibrary, DatabaseLibrary, AutoItLibrary, SSHLibrary, HTTPPLibrary, …

- **Project and team specific libraries**
Separate test data editor (RIDE)
For more information

- **Project pages**
  - The main information hub
  - [http://robotframework.org](http://robotframework.org)
- **Quick Start Guide**
  - Executable guide demonstrating the key features
- **User Guide**
  - Detailed information about all features of the framework
- **SeleniumLibrary demo**
  - Demonstrates web testing with Robot Framework