Testkit-Lite Quick Start

Contents

1. Introduction

Testkit-Lite is a test runner, on which you can automatically run test cases for either middleware components or WebAPIs. This document describes how to write and structure a test case to be run on Testkit-Lite.

2. Test Case Descriptor File

You need to write an .xml test case descriptor file for Testkit-Lite to learn which test cases to run. Below is an example of test case descriptor file for the test case named "HelloWorldTest".

```
<?xml version="1.0" encoding="UTF-8"?>
<test_definition launcher="WRTLauncher">
<suite name=" HelloWorldSuite">
  <set name="HelloWorldSet">
   <testcase component=”HelloWorldComponent" execution_type="auto" purpose=”Test
HelloWorld” id="HelloWorldTest">
    <description>
     <test_script_entry test_script_expected_result="0" timeout="90">/PATH/TO/HelloWorld
       </test_script_entry>
    </description>
  </testcase>
 </set>
</suite>
</test_definition>
```

- **<suite>** and **<set>**

Both **<suite>** and **<set>** function as container element in a test case descriptor file. **<suite>** is the root element of a test case descriptor file, which is parent of one or more **<set>** elements. A **<set>** element in turn contains one or more **<testcase>** child elements.

- **<testcase>**

The **<testcase>** element describes the key information of a test case, including ID, targeting component, test purpose, pre-conditions, test steps, and expected results. For an auto test case, you need to assign **auto** to **execution_type**, and provide at least one <**test_script_entry>** element to tell which commands to run and with what arguments. Be aware that you need to provide the absolute path of test scripts, programs, or HTML web test page, if it is not accessible with the system default **PATH** environment variable. Testkit-Lite verdicts test result to PASS or FAILURE by comparing the real result with value of the attribute **test_script_expected_result.**

- **<test_script_entry>**

<**test_script_entry**> supports two types of commands:

- Executable programs or scripts developed in programming or scripting languages, such as Python, C/C++, and Java.
- HTML web test page. It can either embed JavaSript test code in page content or link to separated files that contain the JavaScript test code.

A test case descriptor file can contain as many test cases as you want and organize them in test sets, allowing batch execution of test cases in one test cycle. You need to update the test case descriptor file when new test cases are ready.

3. Testkit-Lite Constraints

Testkit-Lite has to make assumptions on test cases to reduce the complexity of running them. Constraints are as follows:

- Testkit-Lite uses system built-in normal account, which is a non-root account, to run test executable files by default.
- The test script, program, or HTML web test page in <**test_script_entry>** should be accessible with the system default PATH environment variable. If not, you need to provide full path to make it accessible by Testkit-Lite.
- Testkit-Lite requires a **Web-Runtime Environment** to load web test pages for running WebAPI test cases. A utility tool, **WRTLauncher,** is available for launching and running WebAPI test cases easily.

  Note: WRTLauncher takes the name of widget achieve as input. To launch a WebAPI test case, run the following command:

```
$ WRTLauncher  <widget_name>
```

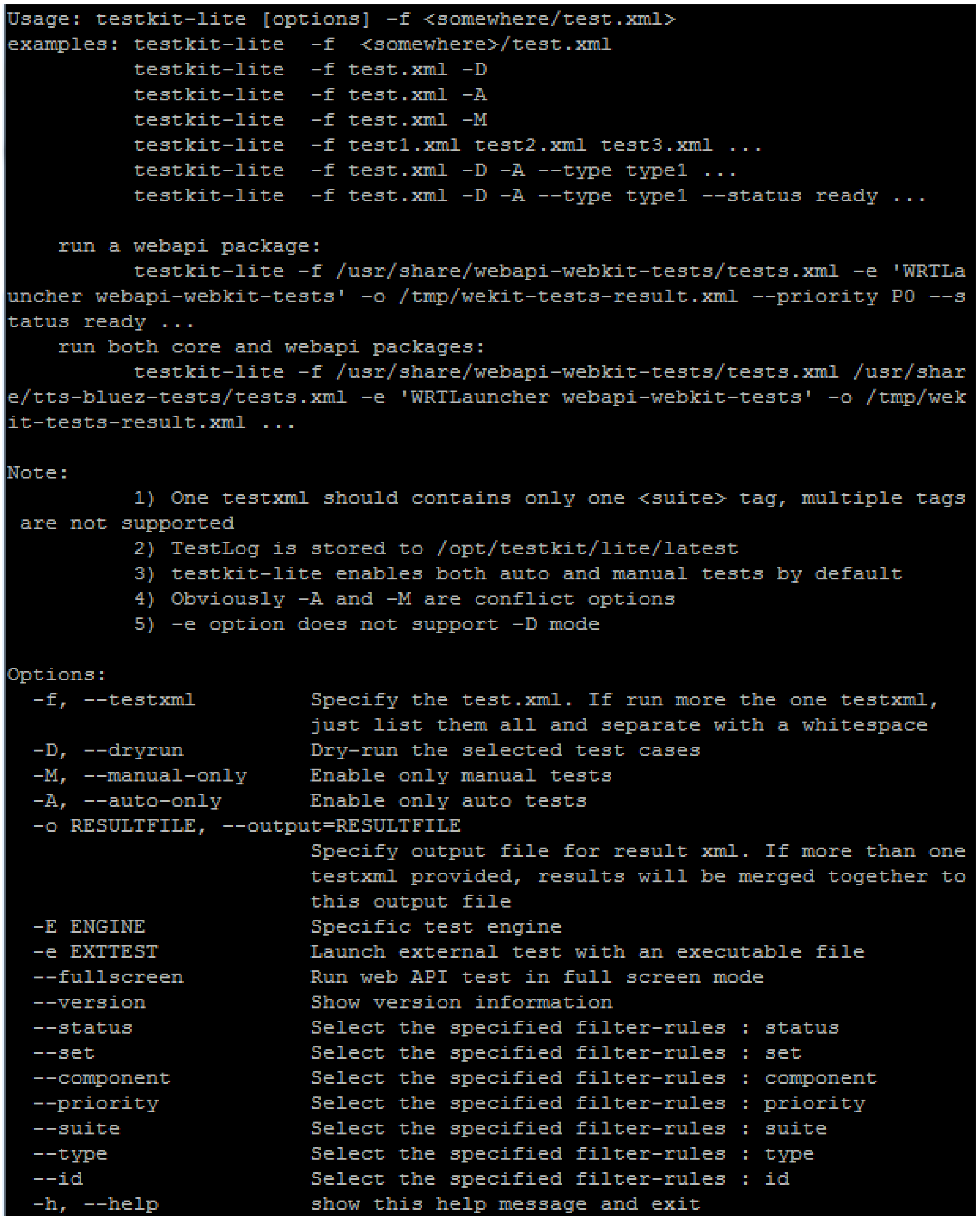
4. Running Test Case

To run a test case, perform the following steps:

1. Check that Testkit-Lite has already been installed on the target test device. On terminal, run the following command:

```
$ testkit-lite   --help
```

The Testkit-Lite help information displays, as shown in Figure 3-1.



```
Usage: testkit-lite [options] -f <somewhere/test.xml>
examples: testkit-lite  -f  <somewhere>/test.xml
          testkit-lite  -f test.xml -D
          testkit-lite  -f test.xml -A
          testkit-lite  -f test.xml -M
          testkit-lite  -f test1.xml test2.xml test3.xml ...
          testkit-lite  -f test.xml -D -A --type type1 ...
          testkit-lite  -f test.xml -D -A --type type1 --status ready ...

    run a webapi package:
          testkit-lite -f /usr/share/webapi-webkit-tests/tests.xml -e 'WRTLa
uncher webapi-webkit-tests' -o /tmp/wekit-tests-result.xml --priority P0 --s
tatus ready ...
    run both core and webapi packages:
          testkit-lite -f /usr/share/webapi-webkit-tests/tests.xml /usr/shar
e/tts-bluez-tests/tests.xml -e 'WRTLauncher webapi-webkit-tests' -o /tmp/wek
it-tests-result.xml ...

Note:
          1) One testxml should contains only one <suite> tag, multiple tags
 are not supported
          2) TestLog is stored to /opt/testkit/lite/latest
          3) testkit-lite enables both auto and manual tests by default
          4) Obviously -A and -M are conflict options
          5) -e option does not support -D mode

Options:
  -f, --testxml         Specify the test.xml. If run more the one testxml,
                        just list them all and separate with a whitespace
  -D, --dryrun          Dry-run the selected test cases
  -M, --manual-only     Enable only manual tests
  -A, --auto-only       Enable only auto tests
  -o RESULTFILE, --output=RESULTFILE
                        Specify output file for result xml. If more than one
                        testxml provided, results will be merged together to
                        this output file
  -E ENGINE             Specific test engine
  -e EXTTEST            Launch external test with an executable file
  --fullscreen          Run web API test in full screen mode
  --version             Show version information
  --status              Select the specified filter-rules : status
  --set                 Select the specified filter-rules : set
  --component           Select the specified filter-rules : component
  --priority            Select the specified filter-rules : priority
  --suite               Select the specified filter-rules : suite
  --type                Select the specified filter-rules : type
  --id                  Select the specified filter-rules : id
  -h, --help            show this help message and exit
```

Figure 3-1 Testkit-Lite help information

2. Deploy the following information to the target test device:
   - Test case descriptor file
   - Test scripts, programs, and HTML web test page
   - Dependency files or test data
3. Run test cases.
   - To run Non-WebAPI test cases:

```
$ testkit-lite   -f /PATH/TO/<test_descriptor_file>.xml
```

- To run WebAPI test cases:

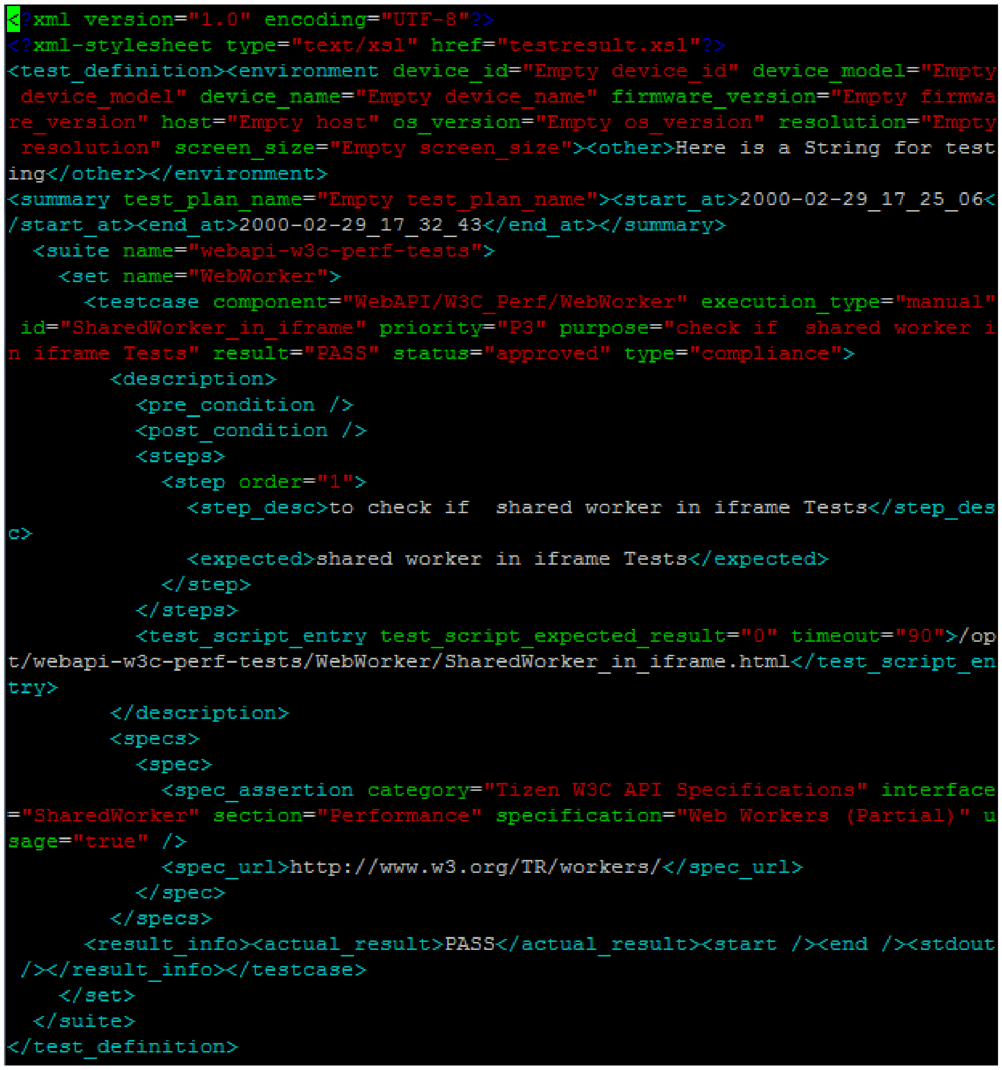
```
$ testkit-lite  -e  "WRTLauncher  <widget _name>" -f /PATH/TO/<test_descriptor_file>.xml
```

For details on more options and usages, see the *Testkit-Lite User Guide*.

5. Checking Test Report

Testkit-Lite creates an .xml test report and locates it under /opt/testkit/lite/latest after it completes executing all test cases successfully. Below is an example.

.xml Test Report, as shown in Figure 5-1



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="testresult.xsl"?>
<test_definition><environment device_id="Empty device_id" device_model="Empty
 device_model" device_name="Empty device_name" firmware_version="Empty firmwa
re_version" host="Empty host" os_version="Empty os_version" resolution="Empty
 resolution" screen_size="Empty screen_size"><other>Here is a String for test
ing</other></environment>
<summary test_plan_name="Empty test_plan_name"><start_at>2000-02-29_17_25_06<
/start_at><end_at>2000-02-29_17_32_43</end_at></summary>
  <suite name="webapi-w3c-perf-tests">
    <set name="WebWorker">
      <testcase component="WebAPI/W3C_Perf/WebWorker" execution_type="manual"
 id="SharedWorker_in_iframe" priority="P3" purpose="check if  shared worker i
n iframe Tests" result="PASS" status="approved" type="compliance">
        <description>
          <pre_condition />
          <post_condition />
          <steps>
            <step order="1">
              <step_desc>to check if  shared worker in iframe Tests</step_des
c>

              <expected>shared worker in iframe Tests</expected>
            </step>
          </steps>
          <test_script_entry test_script_expected_result="0" timeout="90">/op
t/webapi-w3c-perf-tests/WebWorker/SharedWorker_in_iframe.html</test_script_en
try>
        </description>
        <specs>
          <spec>
            <spec_assertion category="Tizen W3C API Specifications" interface
="SharedWorker" section="Performance" specification="Web Workers (Partial)" u
sage="true" />
            <spec_url>http://www.w3.org/TR/workers/</spec_url>
          </spec>
        </specs>
      <result_info><actual_result>PASS</actual_result><start /><end /><stdout
 /></result_info></testcase>
    </set>
  </suite>
</test_definition>
```

Figure 5-1 .xml test report