Web Test Suite Developer Guide

Version 1.0

Contents

1 Overview

This document is intended for developers who contribute WebAPI test cases development.

You are supposed to have gained the following knowledge:

- Where to download WebAPI test source codes and how to run them
- How to download and run Testkit-lite
- What is W3C test harness, which is called by Testkit-lite to support the execution of WebAPI tests

  Note: Testkit-lite is a test execution framework. For details, see https://github.com/testkit/testkit-lite.

2 Test Suite Source Layout

The layout of test source codes should:

- Meet the requirements of Testkit-lite
- Meet the requirements of W3C test harness
- Meet project requirements, for example, support different package formats so that tests can be executed on various platforms

The test suite source layout is detailed as follows:

```
<webapi-xxx-tests>/
├── autogen
├── [common]/
├── configure.ac
├── config.xml.crx
├── config.xml.wgt
├── COPYING
├── icon.png
├── inst.sh.apk
├── inst.sh.ivi
├── inst.sh.wgt
├── inst.sh.xpk
├── Makefile.am
├── manifest.json
├── pack.sh
├── [README]
├── resources/
├── testkit/
├── testcase.xsl
├── testresult.xsl
├── tests.css
├── tests.xml
├── tests.full.xml
├── <testcasefolder>/
├── <webapi-xxx-tests.spec>
├── [utils]/
└── [data]/
```

- <webapi-xxx-tests>: name of WebAPI test package. The 'webapi-' prefix and the '-tests' suffix must be available, for example, webapi-shadowdom-w3c-tests, webapi-input-html5-tests.
- Documents:
    - README: an introduction of the test suite, and (optional) pre-/post-conditions.
    - COPYING: license and copying file
- Test-related files and folders:
    - <testcasefolder>/: a serial of source files or directories for test cases that are well organized by components or features to be tested, e.g. shadowdom/xxx, input/xxx
    - Tests.full.xml & tests.xml: a mandatory file to describe all test cases for this test suite. For details, see "Appendix 2 Tests.full.xml and tests.xml."
- W3C test harness support:
    - [common]/: (optional) integrated from https://github.com/w3c/web-platform-tests/tree/master/common to include common test functions
    - resources/: integrated from https://github.com/w3c/testharness.js to include W3C test harness as an API test framework
- Build/pack support:
    - autogen, configure.ac, and Makefile.am
    - pack.sh: script for generating a zip package
    - inst.sh.apk: script for installing the apk package on Android mobile.
    - inst.sh.ivi: script for installing the xpk package on Tizen IVI device.

    - inst.sh.wgt: script for installing the wgt package on Tizen mobile.
    - inst.sh.xpk: script for installing the xpk package on Tizen mobile.
    - config.xml.crx: configuration file for creating a .crx extension
    - config.xml.wgt: configuration file for creating a .wgt package
    - icon.png: Widget/Extension icon
    - manifest.json: manifest file for creating a .crx extension
    - <webapi-xxx-tests.spec>: specification file including version and configuration for setting suite signature; please set *src_file* to keep the source code files in packaged test suite and put specific files to be kept in *whitelist*. For WebAPI specifications, the 'webapi-' prefix and the '-tests' suffix must be available, for example, webapi-input-html5-tests.spec.
- Installation/execution support:
    - testkit/: web test runner for executing WebAPI test suite. It is integrated from and updated with Testkit-lite. For details, see https://github.com/testkit/testkit-lite.
- Misc:
    - [utils]/: (optional) contains utilities and tools if any
    - [data]/: (optional) contains small-sized data files (Large-sized data such as media content requires a separate package.)
    - Small-sized data files (a few Kbytes) should be included into the tests. Large-sized files should be made available separately. Instructions on how to obtain the data files must be provided in the README file.
    - Test data must be publicly available.

The following files and folders are mandatory in <webapi-xxx-tests>:

- autogen
- configure.ac
- config.xml.crx
- config.xml.wgt
- icon.png
- inst.sh.apk
- inst.sh.ivi
- inst.sh.wgt
- inst.sh.xpk
- Makefile.am
- manifest.json
- pack.sh
- resources/
- testkit/
- testcase.xsl
- testresult.xsl
- tests.css
- tests.xml
- tests.full.xml
- <webapi-xxx-tests.spec>

3 Test Case Coding Style

Test case developers shall follow the following rules:

- Comment each code block in a uniform way
- Return a clear pass/fail result
- Clean environment before exiting tests
- Automate test under condition of stability
- Keep test cases independent from each other
- Keep case independent from UX or vertical specific applications
- Avoid complicated code logic (comment it if unavoidable)
- Avoid duplicated code
- Remove redundant code

Please refer to the ***Coding_Style_Guide_CheatSheet_v1.0*** to get a quick start.

You can find detailed coding style instructions for specific languages from:

1) CSS & HTML: http://google-styleguide.googlecode.com/svn/trunk/htmlcssguide.xml
2) JavaScript: http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml
3) Python: http://google-styleguide.googlecode.com/svn/trunk/pyguide.html
4) Shell: http://google-styleguide.googlecode.com/svn/trunk/shell.xml
5) XML: ‘xmllint --format’ with default indent 2 spaces. See http://xmlsoft.org/xmllint.html

4 Test Case Naming Convention

Template

[SpecShortName]_<WebAPIInterface>_<ShortDescriptionForTestPurpose>

A test case should be named as per the following conventions:

- [SpecShortName] is optional, mostly for similar specifications, e.g. Selectors API Level 1, Selectors API Level 2
- <WebAPIInterface> and <ShortDescriptionForTestPurpose> are mandatory.
- Use lowercase, except API name and constant defined in spec
- Use descriptive names (e.g. ftp_file_send); Do not use numbers as tests name (e.g. _001, _002)
- Use ’_’ to connect words in file names (do not use @&- in case name, though W3C prefer ’-’ to ’_’)

Examples

bluetooth_BluetoothAdapter_discoverDevices_exists.html
Or BluetoothAdapter_discoverDevices_exists.html
webaudio_cancelScheduledValues_exists.html
Or webaudio_cancelScheduledValues_exists.html

5 Test Case Folder Naming Convention

A test case folder should be named as per the following conventions:

- Allow only letter, digit, and hyphen in test case folder name.
- For folder name, please also use lower-case with '-' if necessary.
- Name <testcasefolder> as a spec, component or sub-component, for example, style/, htmltemplates/.

6 Test Case Classification (<testcase> field in tests.xml)

Template

```
<testcase purpose="" type="" status="" component="" execution_type="" priority="" id="">
```

Test case created should be classified by the following rules:

- Purpose: test assertion; should be unique in whole tests.xml (no duplicate test case).
- Type: currently only support 'compliance'.
- Status: test case status
    - designed: test case is just designed but not ready for review.
    - ready: test case is ready for review.
    - approved: test case is reviewed and qualified to be released; currently only use this status when merge tests into test suites.
- Component: should comply with the WebAPI component name list. For details see "Appendix 3 WebAPI Component Name List".
- Execution_type:
    - auto:
    - manual:
- Priority: P0/P1/P2/P3

  - P0: use cases for feature to be tested, API use cases; P0 tests will be used in sanity testing.
  - P1: feature verification tests, API and its attribute presence and normal usage; P0+P1 tests will be used in feature verification testing.
  - P2: positive tests of extended feature tests, API parameter combination tests.
  - P3: negative tests of extended feature tests, API spec descriptive statement tests, complicated use cases, stress tests; P0+P1+P2+P3 tests will be used in full-pass testing.
  - Attribute & Method Coverage - cover each attribute or method at least once by using normal values to ensure the presence of all defined attributes and methods. P0+P1 tests are full tests of Attribute & Method Coverage.
  - Parameter Coverage - a superset of Attribute & Method coverage, which covers each parameter using minimum, maximum, normal, and error conditions of each range of values, parameter combination for the APIs with more than one parameter, and all return codes. P0+P1+P2 tests are full tests of Parameter Coverage.
  - Statement Coverage - a superset of parameter coverage, which covers testable statement, including common usage, error code (exceptions), code examples, and etc testable descriptive statements in each specification document. P0+P1+P2+P3 tests are full tests of Statement Coverage.

- Id: test case identification should be unique in whole tests.xml (no duplicate test case); can be simply as test case name without extension.

Example

```
<testcase purpose="Check if the Touch.screenY attribute exists" type="compliance"
status="approved" component="WebAPI/Device/Touch Events version 1 (Partial)"
execution_type="auto" priority="P1" id="Touch_screenY_exist">
```

7 Spec Coverage Assertion Rules (<specs> field in tests.full.xml)

Template

```
<specs>
  <spec>
    <spec_assertion element_type="" element_name="" interface="" specification="" section=""
category=""/>
    <spec_url></spec_url>
    <spec_statement></spec_statement>
  </spec>
```

```
</specs>

<specs>
  <spec>
    <spec_assertion usage= “” interface=“” specification=“” section=“” category=“”/>
    <spec_url></spec_url>
    <spec_statement></spec_statement>
  </spec>
</specs>
```

Spec coverage assertion should obey the following rules:

- <spec_assertion> field is mandatory.
    - < element_type>: ‘attribute’, ‘method’; only need for P0/P1/P2 test cases.
        - attribute:
        - method:
    - < element_name>: attribute/method name defined in <specification> comes together with < element_type>.
    - < usage>: ‘true’; only need for P3 test cases.
    - <interface>: interface name defined in <specification>.
    - <specification>: web api specification; the 3rd part by ”:” of ”Appendix 2 WebAPI Spec Name List.”
    - <section>: the 2nd part by ”:” of ”Appendix 2 WebAPI Spec Name List.”
        - Tizen
        - UI
        - Widget
        - Content(documents,graphics,multimedia)
        - CSS3
        - Device/OSIntegration
        - Network & Communication
        - Storage
        - Performance
        - ExtraHTML5
    - <category>: the 1st part by ”:” of ”Appendix 2 WebAPI Spec Name List.”
        - Tizen Device API Specifications
        - W3C HTML5 API Specifications
        - Supplementary API Specifications
- <spec_url> is mandatory, URL to public spec section being tested.
- <spec_statement> is optional, statements in spec being tested. It must be copied from the specification document.

Examples

```
<specs>
      <spec>
```

```
          <spec_assertion element_type="attribute" element_name="screenY" interface="Touch"
specification="Touch Events version 1 (Partial)" section="Device" category="Tizen W3C API
Specifications"/>
          <spec_url>http://www.w3.org/TR/2013/WD-touch-events-20130124/#idl-def-
Touch</spec_url>
          <spec_statement/>
        </spec>
</specs>


<specs>
        <spec>
          <spec_assertion element_type="attribute" element_name="clientY" interface="Touch"
specification="Touch Events version 1 (Partial)" section="Device" category="Tizen W3C API
Specifications"/>
          <spec_url>http://www.w3.org/TR/2013/WD-touch-events-20130124/#idl-def-
Touch</spec_url>
          <spec_statement/>
        </spec>
 </specs>
```

8 How to Add New Test Suite to WebAPI

To add a new suite to webapi, perform the following steps:

1) Fork and clone the webtest project from https://github.com/crosswalk-project/webtest
   Note: you must sign up for the GitHub first.
2) Copy a test suite to the spec under testing, for example, "webapi-style-css3-tests".
3) Replace bluetooth with real test case folder name in Makefile.am:

```
commondir = resources testkit
SUBDIRS = style $(commondir)
docdir = $(prefix)/opt/webapi-style-css3-tests
dist_doc_DATA = Changelog COPYING README tests.xml tests.full.xml
```

4) Replace bluetooth with the name of the real test suite and replace bluetooth with real test case folder name used in configure.ac:

```
AC_INIT([webapi-style-css3-tests], [5.34.1.1], [zhiqiang.zhang@intel.com])
AM_INIT_AUTOMAKE([-Wall -Werror foreign])

AC_CONFIG_FILES([Makefile \
style/Makefile \
style/csswg/Makefile \
style/csswg/support/Makefile \
style/csswg/reference/Makefile \
resources/Makefile testkit/Makefile])
AC_OUTPUT
```

5) Update config.xml.crx:

```
<widget xmlns="http://www.w3.org/ns/widgets">
</widget>
```

6) Update config.xml.wgt:

```
<widget id='http://tizen.org/test/webapi-style-css3-tests' xmlns='http://www.w3.org/ns/widgets'
xmlns:tizen='http://tizen.org/ns/widgets'>
  <access origin="*"/>
  <icon src="icon.png" height="117" width="117"/>
  <name>webapi-style-css3-tests</name>
  <tizen:application id="css3styles.WebAPIcss3styleTests" package="css3styles"
required_version="2.2"/>
  <tizen:setting screen-orientation="landscape"/>
</widget>
```

7) Update manifest.json:

```
{
    "version": "0.0.1",
    "name": "webapi-style-css3-tests",
    "permissions": ["tabs", "unlimited_storage", "notifications", "http://*/*", "https://*/*"],
    "description": "webapi-style-css3-tests",
    "file_name": "manifest.json",
    "app": {
        "launch": {
            "local_path": "index.html"
        }
    },
    "icons": {
        "128": "icon.png"
    }
}
```

8) Customize the .spec file based on the webapi-style-css3-tests.spec file.
9) Add new cases to the test suite. For details, see chapter 9 "How to Contribute New Cases to Test Suite Package."

9 How to Contribute New Cases to Test Suite Package

To contribute new cases to test suite package, perform the following steps:

1) Design new test case according to WebAPI Spec and add new case information to **tests.xml**. For details, see **"**Appendix 1 Tests.full.xml and tests.xml."

    - "Case name" should follow the test case naming convention. For details, see chapter 4 "Test Case Naming Convention".
    - "Specs" field should follow the Spec coverage assertion rules. For details, see chapter **Error! Reference source not found.** "Test Case Classification (<testcase> field in tests.xml)".
    - "Component" field should comply with the WebAPI component name list. For details see "Appendix 3 WebAPI Component Name List".

2) Develop test script by following the test case coding style and put it under <testcasefolder>.

Note:

- Each test should have an entry HTML file.
- The test script can be embedded into the HTML file or be used as separate JavaScript file.

- For details on how to use W3C test harness, see http://w3c-test.org/resources/testharness.js.

Example

```
 <!DOCTYPE html>
 <html>
  <head>
    <title>Audio Test: audio_MediaController_play_exists</title>
    <link rel=”author” title=”Intel” href=”http://www.intel.com” />
    <link rel=”help” href=”http://www.w3.org/TR/2012/WD-html5-20121025/media-
elements.html#mediacontroller” />
    <meta name=”flags” content=”” />
    <meta name=”assert” content=”Check if audio.MediaController.play exists”/>
    <script src=”../resources/testharness.js”></script>
    <script src=”../resources/testharnessreport.js”></script>
  </head>
  <body>
    <div id=”log”></div>
    <audio id = ”audio” src=”” mediagroup=”v”></audio>
    <script type=”text/javascript”>
        test(function (){
            var v = document.getElementById(”audio”);
            var controller = v.controller;
            assert_true(”play” in controller, ”audio.MediaController.play exists”);
        }, document.title);
    </script>
  </body>
</html>
```

Appendix 1 Tests.full.xml and tests.xml

Each test suite package has two dedicated .xml files (tests.full.xml and tests.xml), which defines all test cases in the package.

Tests.xml is a simplified version of tests.full.xml; it contains the minimum required elements when running the tests.

Note: The .xml files must comply with the rules in the test_definition.xsd file. For details, see https://github.com/testkit/testkit-lite/blob/master/xsd/test_definition.xsd.

Tests.full.xml Example:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<?xml-stylesheet type=”text/xsl” href=”./testcase.xsl”?>
<test_definition>
  <suite name=”webapi-style-css3-tests” launcher=”xwalk” category=”W3C API Specifications”>
    <set name=”Style”>
      <testcase purpose=”Test checks that a close brace (}) in the style attribute value does not terminate the
```

```
style data when there is no open brace delimiting the declaration list in the CSS style attribute syntax”
type=”compliance” status=”approved” component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute”
execution_type=”manual” priority=”P1” id=”style-attr-braces-001”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
001.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=“CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
   <testcase purpose=”Test checks that the braces must not be included and are therefore invalid when
style attribute values are the content of a declaration block” type=”compliance” status=”approved”
component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual” priority=”P0”
id=”style-attr-braces-002”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
002.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=“CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
   <testcase purpose=”Test checks that the braces in a style attribute are treated as an invalid tokens and
must be paired when dropping declarations” type=”compliance” status=”approved”
component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual” priority=”P1”
id=”style-attr-braces-003”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
003.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=“CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
   <testcase purpose=”Test checks that contains an !important declaration will override the corresponding
declaration” type=”compliance” status=”approved” component=”WebAPI/DOM, Forms and Styles/CSS Style
Attribute” execution_type=”manual” priority=”P1” id=”style-attr-cascade-001”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-
001.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=”CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
   <testcase purpose=”Test checks that an !important declaration takes precedence over a normal
declaration 1” type=”compliance” status=”approved” component=”WebAPI/DOM, Forms and Styles/CSS Style
Attribute” execution_type=”manual” priority=”P1” id=”style-attr-cascade-002”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-
002.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=”CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
   <testcase purpose=”Test checks that the declaration in the style attribute will override the one in the
style element” type=”compliance” status=”approved” component=”WebAPI/DOM, Forms and Styles/CSS
Style Attribute” execution_type=”manual” priority=”P1” id=”style-attr-cascade-003”>
    <description>
     <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-
003.htm</test_script_entry>
    </description>
    <specs>
     <spec>
      <spec_assertion element_type=”attribute” element_name=”style” interface=”CSS” specification=”CSS
Style Attributes” section=”DOM, Forms and Styles” category=”W3C API Specifications”/>
      <spec_url>http://www.w3.org/TR/css-style-attr/#syntax</spec_url>
      <spec_statement/>
     </spec>
    </specs>
   </testcase>
```

```
    </set>
  </suite>
</test_definition>
```

Tests.xml Example:

```
<?xml version=”1.0” encoding=”UTF-8”?>
<?xml-stylesheet type=”text/xsl” href=”./testcase.xsl”?>
<test_definition>
  <suite category=”W3C API Specifications” launcher=”xwalk” name=”webapi-style-css3-tests”>
    <set name=”Style”>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual”
id=”style-attr-braces-001” purpose=”Test checks that a close brace (}) in the style attribute value does not
terminate the style data when there is no open brace delimiting the declaration list in the CSS style attribute
syntax”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
001.htm</test_script_entry>
        </description>
      </testcase>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual”
id=”style-attr-braces-002” purpose=”Test checks that the braces must not be included and are therefore
invalid when style attribute values are the content of a declaration block”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
002.htm</test_script_entry>
        </description>
      </testcase>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual”
id=”style-attr-braces-003” purpose=”Test checks that the braces in a style attribute are treated as an
invalid tokens and must be paired when dropping declarations”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-braces-
003.htm</test_script_entry>
        </description>
      </testcase>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual”
id=”style-attr-cascade-001” purpose=”Test checks that contains an !important declaration will override the
corresponding declaration”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-
001.htm</test_script_entry>
        </description>
      </testcase>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual”
id=”style-attr-cascade-002” purpose=”Test checks that an !important declaration takes precedence over a
normal declaration 1”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-
002.htm</test_script_entry>
```

```
        </description>
      </testcase>
      <testcase component=”WebAPI/DOM, Forms and Styles/CSS Style Attribute” execution_type=”manual” id=”style-attr-cascade-003” purpose=”Test checks that the declaration in the style attribute will override the one in the style element”>
        <description>
          <test_script_entry>/opt/webapi-style-css3-tests/style/csswg/style-attr-cascade-003.htm</test_script_entry>
        </description>
      </testcase>
    </set>
  </suite>
</test_definition>
```

Appendix 2 WebAPI Spec Name List

- Tizen Device API Specifications:Tizen:Alarm
- Tizen Device API Specifications:Tizen:Application
- Tizen Device API Specifications:Tizen:Bluetooth
- Tizen Device API Specifications:Tizen:Bookmark
- Tizen Device API Specifications:Tizen:Calendar
- Tizen Device API Specifications:Tizen:Call History
- Tizen Device API Specifications:Tizen:Contact
- Tizen Device API Specifications:Tizen:Content
- Tizen Device API Specifications:Tizen:Data Control
- Tizen Device API Specifications:Tizen:Data Synchronization
- Tizen Device API Specifications:Tizen:Download
- Tizen Device API Specifications:Tizen:Filesystem
- Tizen Device API Specifications:Tizen:Message Port
- Tizen Device API Specifications:Tizen:Messaging
- Tizen Device API Specifications:Tizen:NFC
- Tizen Device API Specifications:Tizen:Network Bearer Selection
- Tizen Device API Specifications:Tizen:Notification
- Tizen Device API Specifications:Tizen:Package
- Tizen Device API Specifications:Tizen:Power
- Tizen Device API Specifications:Tizen:Push
- Tizen Device API Specifications:Tizen:Secure Element
- Tizen Device API Specifications:Tizen:System Information
- Tizen Device API Specifications:Tizen:System Setting
- Tizen Device API Specifications:Tizen:Time
- Tizen Device API Specifications:Tizen:Tizen
- Tizen Device API Specifications:Tizen:Web Setting
-
- W3C API Specifications:DOM Forms:HTML5 Forms
- W3C API Specifications:DOM Forms:Selectors API Level 1

- W3C API Specifications:DOM Forms:Selectors API Level 2 (Partial)
- W3C API Specifications:DOM Forms:WOFF File Format 1.0
- W3C API Specifications:DOM Forms:DOM/JS related HTML5 Enhancements
-
- W3C API Specifications:CSS3:Media Queries (Partial)
- W3C API Specifications:CSS3:CSS 2D Transforms
- W3C API Specifications:CSS3:CSS 3D Transforms Module Level 3
- W3C API Specifications:CSS3:CSS Animations Module Level 3
- W3C API Specifications:CSS3:CSS Transitions Module Level 3
- W3C API Specifications:CSS3:CSS Colors Module Level 3
- W3C API Specifications:CSS3:CSS Backgrounds and Borders Module Level 3
- W3C API Specifications:CSS3:CSS Flexible Box Layout Module
- W3C API Specifications:CSS3:CSS Multi-column Layout Module
- W3C API Specifications:CSS3:CSS Text Module Level 3 (Partial)
- W3C API Specifications:CSS3:CSS Basic User Interface Module Level 3
- W3C API Specifications:CSS3:CSS Fonts Module Level 3 (Partial)
-
- W3C API Specifications:Device:Touch Events version 1 (Partial)
- W3C API Specifications:Device:DeviceOrientation Event (Partial)
- W3C API Specifications:Device:Battery Status API
- W3C API Specifications:Device:Vibration API
- W3C API Specifications:Device:HTML5 Browser state
- W3C API Specifications:Device:Screen Orientation API
- W3C API Specifications:Device:Network Information API
-
- W3C API Specifications:Graphics:HTML5 canvas element
- W3C API Specifications:Graphics:HTML5 2D Canvas Context
- W3C API Specifications:Graphics:HTML5 SVG
-
- W3C API Specifications:Media:HTML5 audio element
- W3C API Specifications:Media:HTML5 video element
- W3C API Specifications:Media:getUserMedia
- W3C API Specifications:Media:Web Audio API (Partial)
- W3C API Specifications:Media:HTML Media Capture
-
- W3C API Specifications:Communication:The WebSocket API
- W3C API Specifications:Communication:XMLHttpRequest Level 2
- W3C API Specifications:Communication:HTML5 session history
- W3C API Specifications:Communication:Server-Sent Events
- W3C API Specifications:Communication:Web Messaging
-
- W3C API Specifications:Storage:Web Storage
- W3C API Specifications:Storage:File API
- W3C API Specifications:Storage:File API Directories and System
- W3C API Specifications:Storage:File API Writer
- W3C API Specifications:Storage:HTML5 Application Cache

- W3C API Specifications:Storage:Indexed Database API
- W3C API Specifications:Storage:Web SQL Database
-
- W3C API Specifications:Security:Cross-Origin Resource Sharing
- W3C API Specifications:Security:HTML5 iframe element
- W3C API Specifications:Security:Content Security Policy 1.0
-
- W3C API Specifications:UI:Clipboard API and events
- W3C API Specifications:UI:HTML5 Drag and drop
-
- W3C API Specifications:Performance:Web Workers (Partial)
- W3C API Specifications:Performance:Page Visibility
- W3C API Specifications:Performance:Animation Timing Control
- W3C API Specifications:Performance:Navigation Timing
-
- W3C API Specifications:Location:Geolocation API
-
- W3C API Specifications:Widget:Widget Packaging and XML Configuration
- W3C API Specifications:Widget:Widget Interface
- W3C API Specifications:Widget:XML Digital Signatures for Widgets
- W3C API Specifications:Widget:Widgets Access Request Policy
-
- Supplementary API Specifications:Typed Arrays - Khronos (Partial)
- Supplementary API Specifications:WebGL - Khronos (Partial)
- Supplementary API Specifications:Fullscreen API - Mozilla (Partial)
- Supplementary API Specifications:viewport Metatag - Apple (Partial)

Appendix 3 WebAPI Component Name List

Used by “Component” field in tests.full.xml and tests.xml

- TizenAPI/Application/Alarm
- TizenAPI/Application/Application
- TizenAPI/Application/Datacontrol
- TizenAPI/Application/Package
- TizenAPI/Communication/Bluetooth
- TizenAPI/Communication/Messaging
- TizenAPI/Communication/NBS
- TizenAPI/Communication/NFC
- TizenAPI/Communication/Push
- TizenAPI/Communication/SE
- TizenAPI/Content/Content
- TizenAPI/Content/Download
- TizenAPI/IO/Filesystem
- TizenAPI/IO/Messageport

- TizenAPI/Social/Bookmark
- TizenAPI/Social/Calendar
- TizenAPI/Social/Callhistory
- TizenAPI/Social/Contact
- TizenAPI/Social/Datasync
- TizenAPI/System/Power
- TizenAPI/System/Systeminfo
- TizenAPI/System/Systemsetting
- TizenAPI/System/Time
- TizenAPI/System/Websetting
- TizenAPI/Tizen Common/Tizen
- TizenAPI/UI/Notification
-
- WebAPI/Communication/HTML5 The session history of browsing contexts (Partial)
- WebAPI/Communication/HTML5 Web Messaging
- WebAPI/Communication/Server-Sent Events
- WebAPI/Communication/The WebSocket API
- WebAPI/Communication/XMLHttpRequest Level 2 (Partial)
- WebAPI/Device/Battery Status API
- WebAPI/Device/Device Orientation Event Specification (Partial)
- WebAPI/Device/HTML5 Browser state
- WebAPI/Device/The Network Information API
- WebAPI/Device/The Screen Orientation API
- WebAPI/Device/Touch Events version 1 (Partial)
- WebAPI/Device/Vibration API
- WebAPI/DOM, Forms and Styles/CSS 2D Transforms
- WebAPI/DOM, Forms and Styles/CSS 3D Transforms Module Level 3 (Partial)
- WebAPI/DOM, Forms and Styles/CSS Animations Module Level 3
- WebAPI/DOM, Forms and Styles/CSS Backgrounds and Borders Module Level 3 (Partial)
- WebAPI/DOM, Forms and Styles/CSS Basic User Interface Module Level 3 (CSS3 UI) (Partial)
- WebAPI/DOM, Forms and Styles/CSS Color Module Level 3
- WebAPI/DOM, Forms and Styles/CSS Flexible Box Layout Module (Partial)
- WebAPI/DOM, Forms and Styles/CSS Fonts Module Level 3 (Partial)
- WebAPI/DOM, Forms and Styles/CSS Multi-column Layout Module (Partial)
- WebAPI/DOM, Forms and Styles/CSS Style Attribute
- WebAPI/DOM, Forms and Styles/CSS Text Module Level 3 (Partial)
- WebAPI/DOM, Forms and Styles/CSS Transforms Module Level 1
- WebAPI/DOM, Forms and Styles/CSS Transitions Module Level 3
- WebAPI/DOM, Forms and Styles/DOM/JavaScript related HTML5 Enhancements
- WebAPI/DOM, Forms and Styles/HTML5 Forms (Partial)
- WebAPI/DOM, Forms and Styles/Media Queries (Partial)
- WebAPI/DOM, Forms and Styles/Selectors API Level 1
- WebAPI/DOM, Forms and Styles/Selectors API Level 2
- WebAPI/DOM, Forms and Styles/Selectors API Level 2 (Partial)

- WebAPI/Graphics/HTML5 SVG
- WebAPI/Graphics/HTML5 The canvas element (Partial)
- WebAPI/IVI/CSS
- WebAPI/IVI/DLNA
- WebAPI/IVI/Locale
- WebAPI/IVI/Message
- WebAPI/IVI/Notification
- WebAPI/IVI/Speech
- WebAPI/IVI/Vehicle
- WebAPI/JavaScript/Promises
- WebAPI/Location/Geolocation API Specification
- WebAPI/Media/getUserMedia (Partial)
- WebAPI/Media/HTML5 The audio element (Partial)
- WebAPI/Media/HTML5 The video element (Partial)
- WebAPI/Media/HTML Media Capture
- WebAPI/Media/Web Audio API
- WebAPI/Networking/WebRTC
- WebAPI/Performance and Optimization/High Resolution Time
- WebAPI/Performance and Optimization/Navigation Timing
- WebAPI/Performance and Optimization/Page Visibility
- WebAPI/Performance and Optimization/Resource Timing
- WebAPI/Performance and Optimization/Timing control for script-based animations
- WebAPI/Performance and Optimization/User Timing
- WebAPI/Performance and Optimization/Web Workers (Partial)
- WebAPI/Responsive Design/CSS Device Adaptation
- WebAPI/Runtime(short-term)/App URI
- WebAPI/Runtime(short-term)/Runtime
- WebAPI/Screen Presentation APIs/WebScreen
- WebAPI/Security/Content Security Policy
- WebAPI/Security/Cross-Origin Resource Sharing
- WebAPI/Security/HTML5 The iframe element
- WebAPI/Storage/File API
- WebAPI/Storage/File API: Directories and System (Partial)
- WebAPI/Storage/File API: Writer (Partial)
- WebAPI/Storage/HTML5 Application caches
- WebAPI/Storage/Indexed Database API (Partial)
- WebAPI/Storage/Web SQL Database
- WebAPI/Storage/Web Storage (Partial)
- WebAPI/Supplementary API Reference/Fullscreen API - Mozilla (Partial)
- WebAPI/Supplementary API Reference/SIMD
- WebAPI/Supplementary API Reference/Typed Arrays - Khronos (Partial)
- WebAPI/Supplementary API Reference/WebGL - Khronos (Partial)
- WebAPI/Supplementary API Reference/WebGL Specification
- WebAPI/System-level APIs/Contacts Manager
- WebAPI/System-level APIs/Device Capabilities
- WebAPI/System-level APIs/Messaging

- WebAPI/System-level APIs/Raw Sockets
- WebAPI/Tizen/Configuration
- WebAPI/UI/HTML5 Drag and drop
- WebAPI/UI/HTML5 telephone, email and URL state of input element
- WebAPI/UI/Web Notifications (Partial)
- WebAPI/UI/Web Speech
- WebAPI/W3C_EXTRAHTML5/Attributes
- WebAPI/W3C_EXTRAHTML5/Base64
- WebAPI/W3C_EXTRAHTML5/Browsers
- WebAPI/W3C_EXTRAHTML5/Dom
- WebAPI/W3C_EXTRAHTML5/ForeignContent
- WebAPI/W3C_EXTRAHTML5/Rendering
- WebAPI/W3C_EXTRAHTML5/Semantics
- WebAPI/W3C_EXTRAHTML5/Xhtml5
- WebAPI/Web Components/HTML Imports
- WebAPI/Web Components/HTML Templates
- WebAPI/Web Components/Shadow DOM
- WebAPI/Widget/Widget Access Request Policy
- WebAPI/Widget/Widget Interface
- WebAPI/Widget/Widget Packaging and XML Configuration