

WifiController

API Reference

© 2018 All rights reserved by Metrological

This document contains information which is proprietary and confidential to Metrological. It is provided with the expressed understanding that the recipient will not divulge its content to other parties or otherwise misappropriate the information contained herein. This information is furnished for guidance; specifications and availability of goods mentioned in it are subject to change without notice. No part of this publication may be reproduced, stored in a database, retrieval system or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the written prior permission of Metrological.

History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.1 | 7-01-2018 | P. Wielders | Initial version |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Scope 4

1.2 Case sensitivity 4

1.3 Acronyms, Abbreviations and Terms 4

1.4 Standards 4

1.5 References 5

1.6 Open Issues 5

1.7 Limitations 5

2. WebKitBrowser Plugin 6

2.1 Configuration 6

2.2 Application Programming Interface (API) 7

2.2.1 General information 7

2.2.2 State changes 7

2.2.3 URL actions 7

2.3 Events 8

2.3.1 URL changes 8

2.3.2 State changes 8

2.4 JSON definitions 8

2.4.1 General information (browser\_info) 8

2.4.2 URL information (url\_info) 8

# Introduction

## Scope

This document describes the Plugin WifiController API interface. This plugin can be configured to be loaded and executed in the WPEFramework and offers REST full based access to the wpa\_supplicant component running on the system. For details on the WPEFramework API, refer to: [WPEF]

## Case sensitivity

All identifiers on the interface described here are case-sensitive. E.g. an id known in the plugin as 'C0FFEE' is not the same as 'c0ffee'.

All keywords, entities, properties, relations and actions should be treated as case-sensitive.

## Acronyms, Abbreviations and Terms

The next list provides an overview of acronyms and abbreviations used in this document and their definitions.

|  |  |
| --- | --- |
| **Acronym** | **Definitions** |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| UTC | Coordinated Universal Time |

Below terms are listed with their definitions, as used in this document.

|  |  |
| --- | --- |
| **Term** | **Definitions** |
| Callsign | The callsign is the name given to an instance of a plugin. One plugin can be instantiated multiple times, but each instance the instance name, callsign, must be unique. |
| Proxy | An object in one process space representing the “real” object in another process space. The Proxy takes care of marshalling the parameters. |
| Stub | An object in the process space that contains the actual object. The stub takes care of un-marshalling the request from the Proxy and executes the call, on behave of the Proxy object, on the real object |

## Standards

Date time formats between the systems shall be in UTC time and W3C (ISO 8601 profile) formatting [ISO 8601], e.g.: 2004-11-05T13:15:30Z. This way time discontinuities can be avoided due to daylight savings. Note that all interfacing systems must decode/encode the date time to the correct local time.

Languages used in the WPEFramework will be conform [ISO 639-1] using two letter language codes. If WPEFramework encounters a language code it does not recognize, it will use ‘xx’ instead. For a list of available two letter ISO language codes, please visit:  
<http://www.loc.gov/standards/iso639-2/php/code_list.php>

## References

This section lists the references made in this document:

|  |  |
| --- | --- |
| [WPEF] | WPEFramework API Reference  <https://github.com/WebPlatformForEmbedded/WPEFramework> |
| [HTTP] | Hypertext Transfer Protocol  <http://www.w3.org/Protocols> |
| [ISO 8601] | Date and time format  http://www.iso.org/iso/date\_and\_time\_format |
| [ISO-3166] | Country code specification  <http://www.iso.org/iso/country_codes.htm> |
| [ISO-639-1] | Language code specification (Alpha-2 code)  <http://www.loc.gov/standards/iso639-2/php/code_list.php> |
| [JSON] | JavaScript Object Notation  http://www.json.org |
| [URLENC] | URL Encoding  <http://www.w3schools.com/tags/ref_urlencode.asp> |

## Open Issues

This is a list of open issues that needs to be resolved:

* This document is still a work in progress.

## Limitations

The information described in this document is preliminary and subject to change in the future.

Legend:

****

**Be aware of:** implementation choice is needed or side-effect needs to be handled.



**Implementation advice:** Guide line for implementation mostly related to performance.

# WifiController Plugin

## Configuration

|  |  |
| --- | --- |
| callsign | [string] the instance name for the plugin e.g. YouTube. Default: WebKitBrowser. |
| classname | [string] WebKitBrowser. |
| locator | [string] libWPEWebKitBrowser.so |
| autostart | [bool] should the browser plugin be instantiated at the moment the WPEFramework is starts up. |
| configuration | [JSON] JSON object specifying the exact configuration for this plugin. See the next paragraph for details. |

Configuration of the plugin:

|  |  |
| --- | --- |
| interface | [string] the name of the wireless device, typically wlan0. |
| connector | [string] path to the domain socket that exposes the wpa\_supplicant functionality. |

## Application Programming Interface (API)

### General information

Using this method, actual running information can be retrieved from the wpa\_supplicant client.

|  |  |
| --- | --- |
| Request: | GET /Service/WifiController |
| Success: | HTTP/1.1 200 OK  { status } |

|  |  |
| --- | --- |
| Request: | GET /Service/WifiController/Networks |
| Success: | HTTP/1.1 200 OK  { networks } |

|  |  |
| --- | --- |
| Request: | GET /Service/WifiController/Configs |
| Success: | HTTP/1.1 200 OK  { configs } |

### State changes

Using these methods, the browser state (suspend/resume, hidden/visible) can be toggled.

|  |  |
| --- | --- |
| Request: | POST /Service/WebKitBrowser/Suspend |
| Success: | HTTP/1.1 200 OK |

|  |  |
| --- | --- |
| Request: | POST /Service/WebKitBrowser/Resume |
| Success: | HTTP/1.1 200 OK |

|  |  |
| --- | --- |
| Request: | POST /Service/WebKitBrowser/Hide |
| Success: | HTTP/1.1 200 OK |

|  |  |
| --- | --- |
| Request: | POST /Service/WebKitBrowser/Show |
| Success: | HTTP/1.1 200 OK |

### URL actions

Using this method, a new URL can be set. A different page is loaded accordingly.

|  |  |
| --- | --- |
| Request: | PUT /Service/WebKitBrowser /URL  { url\_info } |
| Success: | HTTP/1.1 200 OK |
| Failure | HTTP/1.1 400 Body is missing, incorrect URL. |

## Events

Events are autonomous events, triggered by the internals of the plugin. These events will be broadcasted as JSON to all the connected web socket connections that where opened to this plugin.

### Scan completed

The loading of a page takes time, hence why the reporting of the currently loaded page is signaled via the web socket interface. The following event might be reported:

|  |  |
| --- | --- |
| url | [string] the new URL that is about to be loaded, or has been loaded. |
| loaded | [bool] If the URL has been completely loaded, this element is relayed and set to true. If the element is not present or false, it just means that a change of the URL has been requested. |

### Connected/Disconnected network

Requesting a state change is A-synchronous. The actual state transitionare reported as a state change.

|  |  |
| --- | --- |
| suspended | [bool] is set to true, in case the browser has reached the suspended state. Set to false if the browser reached a resumed state. |

## JSON definitions

### General information (status)

|  |  |
| --- | --- |
| scanning | [bool] true is a network scan is currently in progress. |
| connected | [string] name (SSID) of the network that is currently connected. |

### Network information (networks)

|  |  |
| --- | --- |
| networks | [array] Array of network JSON structs . |

Network definition:

|  |  |
| --- | --- |
| Bssid | [string] Binary SSID associated with this radio. . |
| frequency | [uint32] number indicating the frequency this radio is broadcasting on. |
| signal | [sint32] Signal strength of this radio in dB. . |
| key | [string] Type of keys supported for the pairing (TKIP, CCM etc). Comma separated list of types. . |
| pair | [string] Type of pairing supported by this radio (WPA/WPA2/WEP/unsecure/Enterpise). Comma separated list of types. |
| ssid | [string] Name as this radio can be identified. |

### Config information (configs)

|  |  |
| --- | --- |
| configs | [array] Array of config JSON structs . |

Network definition:

|  |  |
| --- | --- |
| ssid | [string] Name as this radio can be identified. |
| Type | [enum] type of the radio, WPA/ |
| pair | [string] Type of pairing supported by this radio (WPA/WPA2/WEP/unsecure/Enterpise). Comma separated list of types. |
|  |  |