

# Websocket Structure

The websocket messages have a defined structure that must be followed when both sending and receiving a message from a unit/server.

A message has two components, a header with some meta data and the payload, with the contents of the message. All keys in this structure are camelcase.

## Header [object]

The header of the message contains 7 fields as follows:

Field	Type	Structure	Purpose	Example
fromID	string	First letter of 'type' _ 'timestamp of device initialisation' _ version	the ID of the original sender of the message	S_1707151018347_1.0.9
timestamp	number	unixtimestamp	The millisecond unix timestamp of when the message was sent	1707160733725
version	string	version.major.minor	Semantic version of the API version in use	1.0.9
type	string	Capitalised string	The type of the sender, to help identify what the purpose of the device is	Server/Browser
active	boolean	true/false	To determine if the original sending device is still online on the network. The message could be a relayed then cached message	true/false
messageID	number	unixtimestamp	The ID of the original message, this is typically the unixtimestamp at the time of sending, but could be any number. For detecting message duplication	1707160733725

recipients	array	[fromID, fromID]	An array of IDs that have been confirmed to have received this message. This is used in mesh networks to ensure messages don't get stuck in loops. Can also be used to track the path of a message through the network	[S_1707184718392_1.0.9, B_1707174713595_1.0.8]
------------	-------	------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------

## Payload [object]

The structure of the payload is less set as there are options based on usage.

### Module [string]

There will always be a 'module' field in all messages except the register message, as the register message is unique.

The module field is used to determine what module the message is from/for. This field is a string

Current valid options are: prompt, playback, system, config, network

### Command [string]

Like module, there will always be a command field, event register has the command field!

The contains the actual 'command' you are sending to the module, e.g. set, update, delete e.t.c.

### Key/Value [string/any]

When the data you are sending is of the form of a single key/value pair, you can specify a 'key' and a 'value' field and put your values in as a key [string] and value [any]. Generally it is preferred to use the 'data' field instead.

### Data [object/any]

In most cases there will be a 'data' property which contains the actual data. In many cases this is a series of key value pairs, however, in some cases that can be a large complex object and in others simply just a long string.

## Register

To register you send a message the typical header structure and payload with the command "register" and a special payload block "subscribe" which should be an array containing modules names you wish to send and receive messages from. As follows:

```
{
  "header": {
    "fromID": "B_1707160733672_0.1.9",
    "timestamp": 1707160733738,
    "version": "0.1.9",
    "type": "Browser",
    "system": "SignalEnc-XXXX",
    "active": true,
    "messageID": 1707160733738
  },
  "payload": {
    "command": "register",
    "subscribe": [
      "flows",
      "network",
      "system",
      "prompt"
    ]
  }
}
```

## Ping/Pong

There is a background ping/pong every second run by the server to all its attached clients. This should be responded to or your client will be marked as inactive and you will stop receiving messages. If this occurs, sending a new message should re-activate your connection. If you are inactive for too long your connection will be closed by the server.

A ping message will contain a single command with the value "ping", it should be responded to with a single command "pong" as seen below:

```
{
  "header": {
    "fromID": "S_1707151018347_0.1.9",
    "timestamp": 1707160734618,
    "version": "0.1.9",
```

```
"type": "Server",
"active": true,
"messageID": 1707160734618,
"recipients": []
},
"payload": {
  "command": "ping"
}
}
```

## Examples:

### Setting the system name

```
{
  "header": {
    "fromID": "S_1707151018347_1.0.9",
    "timestamp": 1707160733725,
    "version": "1.0.9",
    "type": "Server",
    "active": true,
    "messageID": 1707160733725,
    "recipients": []
  },
  "payload": {
    "module": "system",
    "command": "config",
    "key": "systemName",
    "value": "SygnalEnc-XXXX"
  }
}
```

### Recieveing a flow

```
{
  "header": {
    "fromID": "S_1707151018347_0.1.9",
    "timestamp": 1707160734618,
    "version": "0.1.9",
```

```
"type": "Server",
"active": true,
"messageID": 1707160734618,
"recipients": []
},
"payload": {
  "module": "flows",
  "command": "status",
  "data": {
    "name": "Hardware Encoder #1",
    "ID": "800fcb9b-7627-42be-b447-30c137a828ef",
    "type": "Hardware-Encoder",
    "flowType": "ENCODE",
    "profile": {
      "codec": "h265",
      "profile": "66",
      "bps": "",
      "bpsMin": "",
      "bpsMax": "",
      "gop": "",
      "rcMode": "0",
      "rotation": "0",
      "qpInit": "26",
      "qpMax": "0",
      "qpMin": "0",
      "qpStep": "-1"
    },
    "enabled": false,
    "active": false,
    "caps": {},
    "debug": false
  }
}
}
```

## Ping

```
{
  "header": {
    "fromID": "S_1707151018347_0.1.9",
```

```
"timestamp": 1707160734618,
"version": "0.1.9",
"type": "Server",
"active": true,
"messageID": 1707160734618,
"recipients": []
},
"payload": {
  "command": "ping"
}
}
```

## Receiving the current network interface info

```
{
  "header": {
    "fromID": "S_1707151018347_0.1.9",
    "timestamp": 1707160733723,
    "version": "0.1.9",
    "type": "Server",
    "active": true,
    "messageID": 1707160733723,
    "recipients": []
  },
  "payload": {
    "module": "network",
    "command": "ifaces",
    "data": [
      {
        "ifindex": 1,
        "ifname": "lo",
        "flags": [
          "LOOPBACK",
          "UP",
          "LOWER_UP"
        ],
        "mtu": 65536,
        "qdisc": "noqueue",
```

```
"operstate": "UNKNOWN",
"group": "default",
"txqlen": 1000,
"link_type": "loopback",
"address": "00:00:00:00:00:00",
"broadcast": "00:00:00:00:00:00",
"addr_info": [
  {
    "family": "inet",
    "local": "127.0.0.1",
    "prefixlen": 8,
    "scope": "host",
    "label": "lo",
    "valid_life_time": 4294967295,
    "preferred_life_time": 4294967295
  },
  {
    "family": "inet6",
    "local": "::1",
    "prefixlen": 128,
    "scope": "host",
    "valid_life_time": 4294967295,
    "preferred_life_time": 4294967295
  }
],
{
  "ifindex": 2,
  "ifname": "enP2p33s0",
  "flags": [
    "BROADCAST",
    "MULTICAST",
    "UP",
    "LOWER_UP"
  ],
  "mtu": 1500,
  "qdisc": "mq",
  "operstate": "UP",
  "group": "default",
  "txqlen": 1000,
```

```
"link_type": "ether",
"address": "7a:b2:4e:29:01:79",
"broadcast": "ff:ff:ff:ff:ff:ff",
"permaddr": "3e:9f:56:b0:fe:c0",
"addr_info": [
  {
    "family": "inet",
    "local": "192.168.1.29",
    "prefixlen": 24,
    "broadcast": "192.168.1.255",
    "scope": "global",
    "dynamic": true,
    "noprfixroute": true,
    "label": "enP2p33s0",
    "valid_life_time": 2775,
    "preferred_life_time": 2775
  },
  {
    "family": "inet6",
    "local": "2a0d:3344:1d6:9210:3f:39e3:590c:a04b",
    "prefixlen": 64,
    "scope": "global",
    "temporary": true,
    "dynamic": true,
    "valid_life_time": 1194,
    "preferred_life_time": 594
  },
  {
    "family": "inet6",
    "local": "2a0d:3344:1d6:9210:bbab:1a3:272a:8c49",
    "prefixlen": 64,
    "scope": "global",
    "dynamic": true,
    "mngtmpaddr": true,
    "noprfixroute": true,
    "valid_life_time": 1194,
    "preferred_life_time": 594
  },
  {
    "family": "inet6",
```



```
"local": "fdb4:be22:9e07:10::f5b",
"prefixlen": 128,
"scope": "global",
"dadfailed": true,
"tentative": true,
"noprefixroute": true,
"valid_life_time": 4294967295,
"preferred_life_time": 4294967295
},
{
  "family": "inet6",
  "local": "fdb4:be22:9e07:10:cb85:5d3d:4f:f4ac",
  "prefixlen": 64,
  "scope": "global",
  "temporary": true,
  "dynamic": true,
  "valid_life_time": 600378,
  "preferred_life_time": 81483
},
{
  "family": "inet6",
  "local": "fdb4:be22:9e07:10:d098:9a1e:6396:72be",
  "prefixlen": 64,
  "scope": "global",
  "mngtmpaddr": true,
  "noprefixroute": true,
  "valid_life_time": 4294967295,
  "preferred_life_time": 4294967295
},
{
  "family": "inet6",
  "local": "fe80::4408:7058:8fe9:54f9",
  "prefixlen": 64,
  "scope": "link",
  "noprefixroute": true,
  "valid_life_time": 4294967295,
  "preferred_life_time": 4294967295
}
],
"gateway": "192.168.1.1"
```

```
},
{
  "ifindex": 3,
  "ifname": "enP4p65s0",
  "flags": [
    "BROADCAST",
    "MULTICAST",
    "UP",
    "LOWER_UP"
  ],
  "mtu": 1500,
  "qdisc": "mq",
  "operstate": "UP",
  "group": "default",
  "txqlen": 1000,
  "link_type": "ether",
  "address": "46:62:0d:1d:e2:5b",
  "broadcast": "ff:ff:ff:ff:ff:ff",
  "permaddr": "aa:04:f5:fd:4e:35",
  "addr_info": [
    {
      "family": "inet",
      "local": "192.168.88.223",
      "prefixlen": 24,
      "broadcast": "192.168.88.255",
      "scope": "global",
      "dynamic": true,
      "noprefixroute": true,
      "label": "enP4p65s0",
      "valid_life_time": 419,
      "preferred_life_time": 419
    },
    {
      "family": "inet6",
      "local": "fe80::51d7:4511:97ea:b968",
      "prefixlen": 64,
      "scope": "link",
      "noprefixroute": true,
      "valid_life_time": 4294967295,
      "preferred_life_time": 4294967295
    }
  ]
}
```

```
}
],
"gateway": "192.168.88.1"
},
{
  "ifindex": 4,
  "ifname": "wlp3p49s0",
  "flags": [
    "BROADCAST",
    "MULTICAST",
    "UP",
    "LOWER_UP"
  ],
  "mtu": 1500,
  "qdisc": "noqueue",
  "operstate": "UP",
  "group": "default",
  "txqlen": 1000,
  "link_type": "ether",
  "address": "a0:02:a5:bd:d4:36",
  "broadcast": "ff:ff:ff:ff:ff:ff",
  "addr_info": [
    {
      "family": "inet",
      "local": "192.168.1.39",
      "prefixlen": 24,
      "broadcast": "192.168.1.255",
      "scope": "global",
      "dynamic": true,
      "noprefixroute": true,
      "label": "wlp3p49s0",
      "valid_life_time": 3373,
      "preferred_life_time": 3373
    },
    {
      "family": "inet6",
      "local": "2a0d:3344:1d6:9210::f5b",
      "prefixlen": 128,
      "scope": "global",
      "dynamic": true,
```

```
"noprfixroute": true,
"valid_life_time": 994,
"preferred_life_time": 394
},
{
  "family": "inet6",
  "local": "2a0d:3344:1d6:9210:996e:cd23:3952:2114",
  "prefixlen": 64,
  "scope": "global",
  "temporary": true,
  "dynamic": true,
  "valid_life_time": 1194,
  "preferred_life_time": 594
},
{
  "family": "inet6",
  "local": "2a0d:3344:1d6:9210:3f13:80eb:c42b:56d3",
  "prefixlen": 64,
  "scope": "global",
  "dynamic": true,
  "mngtmpaddr": true,
  "noprfixroute": true,
  "valid_life_time": 1194,
  "preferred_life_time": 594
},
{
  "family": "inet6",
  "local": "fdb4:be22:9e07:10::f5b",
  "prefixlen": 128,
  "scope": "global",
  "noprfixroute": true,
  "valid_life_time": 4294967295,
  "preferred_life_time": 4294967295
},
{
  "family": "inet6",
  "local": "fdb4:be22:9e07:10:9cd6:5d24:63b6:896a",
  "prefixlen": 64,
  "scope": "global",
  "temporary": true,
```

```
"dynamic": true,
"valid_life_time": 600319,
"preferred_life_time": 81343
},
{
  "family": "inet6",
  "local": "fdb4:be22:9e07:10:90c1:4b8f:4d19:68e0",
  "prefixlen": 64,
  "scope": "global",
  "mngtmpaddr": true,
  "noprefixroute": true,
  "valid_life_time": 4294967295,
  "preferred_life_time": 4294967295
},
{
  "family": "inet6",
  "local": "fe80::a750:aedd:b0d8:d2c2",
  "prefixlen": 64,
  "scope": "link",
  "noprefixroute": true,
  "valid_life_time": 4294967295,
  "preferred_life_time": 4294967295
}
],
"gateway": "192.168.1.1"
}
]
}
}
```

---

Revision #1

Created 13 February 2024 18:22:14 by Admin

Updated 13 February 2024 19:00:22 by Admin