



SDRconnect WebSocket API 1.0.3

Overview

SDRconnect exposes a bidirectional, real-time API over WebSockets.

The following specification applies equally to SDRconnect GUI and SDRconnect Headless.

All TEXT messages are JSON-encoded and follow a request-response pattern for operations that require acknowledgment and an event pattern for push-only notifications. A WebSocket message should contain a single JSON message only.

All BINARY messages are either signed little endian 16-bit integers or 8-bit unsigned integers depending on the content type.

In case of a dual tuner device, the terms Primary and Secondary Device refer to the first and second tuners channels respectively. When the device message field is not explicitly set the API shall assume Primary.

Message Structure

All JSON messages share a common envelope:

Field	Type	Required
event_type	String	Yes
property	String	Yes
value	String	Yes
device	String	No

Push Messages from SDRconnect

Valid event_type field values for messages from SDRconnect

event_type	Purpose	value
property_changed	Push notification of internal property value change	Internal property value
get_property_response	Response containing requested property value	Internal property value

Example push message from SDRconnect

```
{
  "event_type": "property_changed",
  "property": "device_center_frequency",
  "device": "primary",
  "value": "100000000"
}
```

Example response to get_property from SDRconnect

```
{
  "event_type": "get_property_response",
  "property": "device_vfo_frequency",
  "device": "primary",
  "value": "101000000"
}
```

Client Messages to SDRconnect

Valid event_type field values for messages to SDRconnect

event_type	Purpose	value
set_property	Modify internal property	"<property value>"
get_property	Request internal property value	N/A
iq_stream_enable	Enable IQ data streaming over WebSocket	"true" "false"
audio_stream_enable	Enable demodulated audio streaming over WebSocket	"true" "false"
spectrum_enable	Enable spectrum bins streaming over WebSocket	"true" "false"
device_stream_enable	Set streaming status of selected device	"true" "false"
selected_device	Set active device by index	"<unsigned integer index>"
selected_device_serial	Set active device by serial number	"<device serial>" "<device serial>:<network mode>" (Full IQ, IQ Lite, Compact)
selected_device_name	Set active device by display name	"<display name>" "<device serial>:<network mode>" (Full IQ, IQ Lite, Compact)
start_recording	Start specified recording type	"iq" "audio" "compressed_audio"
stop_recording	Stop recording	N/A
apply_device_profile	Apply specified device profile	"<device profile name>"
set_primary_device_enable	Filter events and binary data from Primary Device	"true" "false"
set_secondary_device_enable	Filter events and binary data from Secondary Device	"true" "false"

Example message from client to set property

```
{  
  "event_type": "set_property",  
  "property": "device_center_frequency",  
  "device": "primary",  
  "value": "101000000"  
}
```

Example message from client to enable audio streaming

```
{  
  "event_type": "audio_stream_enable",  
  "property": "",  
  "value": "true"  
}
```

Example message from client to select device by index

```
{
  "event_type": "selected_device",
  "property": "",
  "value": "1"
}
```

Valid property field values

Property	Description	Read-Only	Value Type
device_center_frequency	Hardware LO frequency	No	Uint64 (Hz)
device_sample_rate	Hardware sample rate	No	Double (Hz)
device_vfo_frequency	VFO tuned frequency	No	Uint64 (Hz)
lna_state	RF gain	No	Uint32
lna_state_min	RF gain minimum	Yes	Uint32
lna_state_max	RF gain maximum	Yes	Uint32
filter_bandwidth	VFO bandwidth	No	Uint32 (Hz)
demodulator	VFO Demodulator Mode	No	Enum: (AM/USB/LSB/CW/SAM/NFM/WFM)
demod_max_bandwidth	VFO maximum filter bandwidth	Yes	Uint32 (Hz)
started	Device streaming status	Yes	Boolean
overload	Device overload status	Yes	Boolean
can_control	Device hardware control	Yes	Boolean
audio_volume_percent	Audio playback volume	No	Percent
audio_mute	Audio playback mute enable	No	Boolean
audio_limiters	Audio limiting enable	No	Boolean
audio_filter	Audio filters enable	No	Boolean
squelch_enable	Squelch enable	No	Boolean
squelch_threshold	Squelch threshold	No	
agc_enable	Audio AGC enable	No	Boolean
agc_threshold	Audio AGC threshold	No	
wfm_stereo_enable	WFM Stereo decode enable	No	Boolean
noise_reduction_enable	Noise Reduction enable	No	Boolean
noise_reduction_strength	Noise Reduction threshold	No	Percent

nfm_deemphasis_enable	NFM deemphasis enable	No	Boolean
spectrum_ref_level	Spectrum display top position	No	
spectrum_base	Spectrum display bottom position	No	
rds_ps	RDS Program Service Name	Yes	String
rds_pi	RDS PI Code	Yes	Uint16
rds_pty	RDS Program Type	Yes	Uint16
rds_radiotext	RDS Radio Text	Yes	String
rds_enable	RDS Decoder enable	No	Boolean
signal_power	Measured signal power	Yes	Double (dB)
signal_snr	Measured signal SNR	Yes	Double (dB)
wfm_stereo	WFM Stereo PLL lock status	Yes	Boolean
am_lowcut_frequency	AM demodulator low cut frequency	No	Uint32 (Hz)
ssb_lowcut_frequency	SSB demodulator low cut frequency	No	Uint32 (Hz)
nfm_lowcut_frequency	NFM demodulator low cur frequency	No	Uint32 (Hz)
valid_antennas	Comma delimited list of available antenna names	Yes	String
active_antenna	Name of active antenna	No	String
valid_devices	Comma delimited list of available device display names	Yes	String
active_device	Display name of currently in use device	Yes	String
api_version	WebSocket API version	Yes	String (e.g. "1.0.3")

Binary Messages

Binary messages start with a 2-byte payload type identifier that should be interpreted as a little endian unsigned 16-bit integer.

Message Type	Payload Type
1	Signed 16-bit PCM Stereo Audio @ 48kHz (LRLR) (Primary Device)
2	Signed 16-bit interleaved IQ (IQIQ) (Primary Device)
3	Unsigned 8-bit spectrum FFT bin normalised to visible range. (Primary Device)
4	Signed 16-bit PCM Stereo Audio @ 48kHz (LRLR) (Secondary Device)
5	Signed 16-bit interleaved IQ (IQIQ) (Secondary Device)
6	Unsigned 8-bit spectrum FFT bin normalised to visible range. (Secondary Device)