

WebSocket API

2019.7.3

Doc V: 0.10.8.1

Websocket url: ws://localhost:31278/ws

Find Dongle:

API	Client Request	Server Response
setting		
request	<pre>{ "type": { "type": "request", "target_type": "device", "target_name": "dongle" }, "name": null, "contents": "find_dongle" }</pre>	<pre>{ "type": { "type": "response", "source_type": "device", "source_name": "dongle" }, "name": null, "contents": { "dongle_list" : ["STEEG_00", "STEEG_01"] } }</pre>
streaming		

Connect Device:

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "device", "target_name": "dongle" }, "name": null, "contents": { "on_off": true, (true >> turn on false >> turn off) "target_id": "DONGLE_00", "ch_config": 19 } }</pre> <p>(0 >> default 19 >> use 32ch as 19ch)</p>	<pre>{ "type": { "type": "ack", "source_type": "device", "source_name": "dongle" }, "name": null, "contents": { "result": true, "status": true } }</pre>
request		
streaming		

Device’s Information:

API	Client Request	Server Response
setting		
request	<pre>{ "type": { "type": "request", "target_type": "device", "target_name": "device" }, "name": null, "contents": "device_info" }</pre>	<pre>{ "type": { "type": "response", "source_type": "device", "source_name": "device" }, "name": null, "contents": { "sampling_rate": 1000, "resolution": 24, "ch_num": 8, "ch_label": ["Fp1", "Fp2", "Pz", ..., "O2"] "battery": 87, "message": "" } }</pre> ("message" will restore latest system message)
streaming		

System Message: (system responds automatically)

API	Client Request	Server Response
setting		
request		<pre>{ "type": { "type": "response", "source_type": "system", "source_name": "system" }, "name": null, "contents": { "message": "something....." } }</pre>
streaming		

System Shut Down:

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "system", "target_name": "system" }, "name": null, "contents": { "shut_down": true } }</pre>	
request		
streaming		

Contact Impedance :

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "device", "target_name": "impedance" }, "name": null, "contents": { "enable": true, } }</pre>	<pre>{ "type": { "type": "ack", "source_type": "device", "source_name": "impedance" }, "name": null, "contents": { "result": true } }</pre>
request	<pre>{ "type": { "type": "request", "target_type": "device", "target_name": "impedance" }, "name": null, "contents": { "requirement": ["enable", "sps_origin", "ch_num", "ch_label"] } }</pre>	<pre>{ "type": { "type": "response", "source_type": "device", "source_name": "impedance" }, "name": null, "contents": { "enable": true, "sps_origin": 1000, "ch_num": 8, "ch_label": ["Fp1", "Fp2", "Pz", ..., "O2"] } }</pre>
streaming		<pre>{ "type": { "type": "data", "source_type": "device", "source_name": "impedance" }, "name": null, "contents": { "sync_tick": 4456, "impedance": [[50.2, 46.9, 252.3, ..., 605.6], [1, 1, 1, ..., 1]] } }</pre> <p>(The first array is for impedance measurement, the second array is for Lead-on/off detection.)</p>

Raw Data :

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "raw", "target_name": "raw" }, "name": null, "contents": { "enable": true, "chunk_size": 4 } }</pre>	<pre>{ "type": { "type": "ack", "source_type": "raw", "source_name": "raw" }, "name": null, "contents": { "result": true } }</pre>
request	<pre>{ "type": { "type": "request", "target_type": "raw", "target_name": "raw" }, "name": null, "contents": { "requirement": ["enable", "sps_origin", "ch_num", "chunk_size", "ch_label"] } }</pre>	<pre>{ "type": { "type": "response", "source_type": "raw", "source_name": "raw" }, "name": null, "contents": { "enable": true, "sps_origin": 1000, "ch_num": 8, "chunk_size": 50, "ch_label": ["Fp1", "Fp2", "Pz", ..., "O2"] } }</pre>
streaming	<pre>"contents": [{sample-1}, {sample-2}, {sample-3}, {sample-4}]</pre>	<pre>{ "type": { "type": "data", "source_type": "raw", "source_name": "raw" }, "name": null, "contents": [{ "sync_tick": 4456, "eeg": [-1.051, 7.113, 11.512, ..., -3.123], "event": { "event_id": [125, 126], "event_duration": [0.5, 1] }, "g_sensor": {"X": 10, "Y": 1, "Z": -2}, "gyro": null, "battery_power": null, "machine_info": null }, {.....}, {.....}, { "sync_tick": 4459, "eeg": [-2.055, -5.654, 0.123, ..., 1.234], "event": { "event_id": [], "event_duration": [] }, "g_sensor": {"X": 20, "Y": 0, "Z": 1}, "gyro": null, "battery_power": null, "machine_info": null }] }</pre> <div data-bbox="806 1118 1320 1813" style="border: 2px dashed red; padding: 10px; margin-top: 10px;">data chunk</div>

Decimated Data :

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "algorithm", "target_name": "decimation" }, "name": null, "contents": { "enable": true, "use_clean_data": true, "decimate_num": 2, "notch_filter": 0 } }</pre> <p>(0 >> None 50 >> 50Hz 60 >> 60Hz)</p>	<pre>{ "type": { "type": "ack", "source_type": "algorithm", "source_name": "decimation" }, "name": null, "contents": { "result": true } }</pre>
request	<pre>{ "type": { "type": "request", "target_type": "algorithm", "target_name": "decimation" }, "name": null, "contents": { "requirement": ["enable", "sps_origin", "sps_decimated", "decimate_num", "use_clean_data", "notch_filter", "ch_num", "ch_label"] } }</pre>	<pre>{ "type": { "type": "response", "source_type": "algorithm", "source_name": "decimation" }, "name": null, "contents": { "enable": true, "sps_origin": 1000, "sps_decimated": 500, "decimate_num": 2, "use_clean_data": true, "notch_filter": 0, "ch_num": 8, "ch_label": ["Fp1", "Fp2", "Pz", ..., "O2"] } }</pre>
streaming		<div><pre>{ "type": { "type": "data", "source_type": "algorithm", "source_name": "decimation" }, "name": null, "contents": [{ "sync_tick": 4456, "data": [-1.051, 7.113, 11.512, ..., -3.123], "event": null }, {...}, {...}, { "sync_tick": 4462, "data": [-1.051, 7.113, 11.512, ..., -3.123], "event": { "event_id": 125, "event_duration": [] } }] }</pre></div> <p>data chunk</p>

Spectral Analysis (FFT) :

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "algorithm", "target_name": "FFT" }, "name": null, "contents": { "enable": true, "window_size": 2, "window_interval": 0.5, "freq_range": [0,30], "notch_filter": 0 (0 >> None 50 >> 50Hz 60 >> 60Hz) } }</pre>	<pre>{ "type": { "type": "ack", "source_type": "algorithm", "source_name": "FFT" }, "name": null, "contents": { "result": true } }</pre>
request	<pre>{ "type": { "type": "request", "target_type": "algorithm", "target_name": "FFT" }, "name": null, "contents": { "requirement": ["enable", "window_size", "window_interval", "freq_range", "freq_label", "data_size", "ch_label", "notch_filter"] } }</pre>	<pre>{ "type": { "type": "response", "source_type": "algorithm", "source_name": "FFT" }, "name": null, "contents": { "enable": true, "window_size": 2, "window_interval": 0.5, "freq_range": [0,30], "freq_label": [0,0.5,1,...,30], "data_size": [8,61], ([# of chs., # of freq.]) "notch_filter": 0 } }</pre>
streaming	<p>(psd = [[ch1-freq0, ch1-freq1, ch1-freq2,..., ch1-freq61], [ch2-freq0, ch2-freq1,..., ..., ..., ch8-freq61];)</p> <p>(power = [[ch1-delta, ch1-theta, ch1-alphaLow, ch1-alphaHigh, ch1-betaLow, ch1-betaMid, ch1-betaHigh, ch1-gamma], [ch2-delta, ch2-theta,...], ..., [ch8-gamma];) ➤ Each channel has 8 power bands.</p>	<pre>{ "type": { "type": "data", "source_type": "algorithm", "source_name": "FFT" }, "name": null, "contents": { "sync_tick": 3999, "psd": [[0.051,...,0.113],..., [0.512,...,0.123]] ← "band": { "power": [[0.051,...,0.113],..., ← [0.512,...,0.123]] "z_score_all": [[0.051,...,0.113], ..., [0.512,...,0.123]] "z_score_each": [[0.051,...,0.113], ..., [0.512,...,0.123]] } } }</pre>

BCI Index :

API	Client Request	Server Response
setting	<pre>{ "type": { "type": "setting", "target_type": "algorithm", "target_name": "BCI" }, "name": null, "contents": { "enable": true, "ch_used": [0,1,6,7] } }</pre>	<pre>{ "type": { "type": "ack", "source_type": "algorithm", "source_name": "BCI" }, "name": null, "contents": { "result": true } }</pre>
request	<pre>{ "type": { "type": "request", "target_type": "algorithm", "target_name": "BCI" }, "name": null, "contents": { "requirement": ["enable","ch_used"] } }</pre>	<pre>{ "type": { "type": "response", "source_type": "algorithm", "source_name": "BCI" }, "name": null, "contents": { "enable": true, "ch_used": [0,1,6,7] } }</pre>
streaming		<pre>{ "type": { "type": "data", "source_type": "algorithm", "source_name": "BCI" }, "name": null, "contents": { "sync_tick": 3999, "Index":{ "Concentration": 1.2, "EMG": 1.1, "Blink": 0.6 } } }</pre>