```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>データ</title>
    <script src="js/chart.js"></script>
    <script src="js/mylib.js"></script>
  </head>
<body>
<h1>My データ・サーバ : データ表示</h1>
<h2 id="channel_name">>>チャネル:<h2>
<div class="main" id="main_area"></div>
<br>
<button onclick="window.close()">閉じる</button>

<script>

  /*
    この画面では、
    ・出力期間は絞らない
    ・更新間隔は、1 分
    という仕様で実装する。
    API 的には細かくデータ取得できるので凝った画面を作ってみてください。
  */

  var channel_id = null;
  let elm = document.URL.split('?');
  if (elm[1]) {
      let param = elm[1].split('=');
      if (param[0] == "channel_id" && param[1] ) {
          channel_id=param[1];
      }
  }
  console.log(channel_id);

  let cn = document.getElementById("channel_name");
  cn.innerHTML = "チャネル：" + channel_id;


  // 間隔の初期値
```

```
41	var interval = 60 * 1000;
42	var mon_h = null;
43
44	function start_monitor() {
45		let dt = new Date();
46		console.log("start_monitor", dt);
47		mon_h = setInterval(process, interval, interval);
48		process(interval);
49	}
50
51	function process(offset) {
52		console.log("process", offset)
53		load_chart(channel_id);
54	}
55
56	function load_chart(channel_id) {
57		console.log("load_chart:", channel_id);
58		/*
59			画面仕様としては、
60		*/
61		let measurement = null;
62		let from = null;
63		let to = null;
64		let m = null;
65		let f = null;
66		let v = null;
67		get_influxdb_data(channel_id, null, null, null, function(list){  /* callback of get_influxdb_data */
68			console.log("enter get_data callback");
69			let m_data = {};
70			let fields = {};
71			let times = {};
72			let times_idx = {}
73			let timestamp_p = {}
74
75			for (let rec of list['data']) {
76				console.log(rec);
77				for (let key in rec) {
78					console.log(key);
79					console.log(rec[key]);
80					if (key == "measurement") {
```

```
81                                  m = rec[key];
82                              }
83                              else if (key == "timestamp") {
84                                  ts = rec[key];
85                              }
86                              else if (key == "source") {
87                                  s = rec[key];
88                              }
89                              else {
90                                  f = key;
91                                  v = rec[key];
92                              }
93                          }
94                          /* measurement */
95                          if (m_data[m] == null) {
96                              m_data[m] = {};
97                              times_idx[m] = -1;
98                              timestamp_p[m] = "20220101000000";
99                              times[m] = [];
100                         }
101                         let idx = times[m].indexOf(ts);
102                         if (idx == -1) {
103                             times_idx[m]++;
104                             idx = times_idx[m];
105                             times[m][times_idx[m]] = ts
106                         }
107                         /* field */
108                         fields = m_data[m];
109                         if (fields[f] == null) {
110                             fields[f] = [];
111                         }
112                         fields[f][idx] = v
113                     }
114                     console.log(times);
115                     console.log(times_idx);
116                     console.log(fields);
117                     console.log(m_data);
118
119                     // 表示領域を全部削除
120                     remove_chart_area();
```

```
121
122                     for (let m in m_data) {
123                         // 表示する領域を並べる
124                         add_chart_area(m);
125
126                         // ここで ID から名前に書き換える
127                         let c = document.getElementById(m+"_caption");
128                         c.innerHTML = m;
129
130                         // チャート表示
131                         color_list = ["rgb(0,0,255)", "rgb(255,0,0)", "rgb(0,128,0)", "rgb(0,255,255)",
132                                         "rgb(255,0,255)", "rgb(0,255,0)", "rgb(255,255,0)", "rgb(128,128,0)", "rgb(128,128,128)",
133                                         "rgb(128,0,128)", "rgb(128,0,0)", "rgb(0,0,128)", "rgb(0,128,128)", "rgb(0,0,0)" ];
134                         data = [];
135                         let i = 0;
136                         for (let key in m_data[m]) {
137                             data.push({
138                                 label: key,
139                                 fill:false,
140                                 lineTension:0,
141                                 data:m_data[m][key],
142                                 borderColor: color_list[i]
143
144                             });
145                             i++;
146                         }
147                         let cv = document.getElementById(m + "_chart");
148                         let mychart = new Chart(cv, {
149                             type: 'line',
150                             data: {
151                                 labels:times[m],
152                                 datasets:data
153                             }
154                         });
155
156                     }
157             });    /* callback of get_influxdb_data*/
158
159         }
160
```

```
161
162         // 動的に要素を追加
163         function add_chart_area(monitor_id) {
164                 let div_element = document.createElement("div");
165                 div_element.setAttribute("class", "monitor");
166                 div_element.setAttribute("id", monitor_id + "_area");
167                 div_element.innerHTML = '<p class="caption" id="' + monitor_id + '_caption' +'">' + monitor_id + '</p><p
class="chart"><canvas id="' + monitor_id + '_chart"></canvas></p>';
168                 let parent_object = document.getElementById("main_area");
169                 parent_object.appendChild(div_element);
170         }
171
172         function remove_chart_area() {
173             let main_area = document.getElementById("main_area");
174             while(main_area.firstChild){
175                 main_area.removeChild(main_area.firstChild);
176             }
177         }
178
179         start_monitor();
180         load_chart(channel_id);
181
182
183     </script>
184     </body>
185     </html>
```