

```
1  import {Log} from './modules/log.js';
2  import {buildRow, formatPlaybackTime} from './modules/helper.js';
3  import {EditableField} from './modules/editablefield.js';
4
5  window.onload = setup;
6  google.charts.load('current', {'packages':['corechart']});
7
8  var id;
9
10 function setup() {
11
12     const params = new URLSearchParams(location.search);
13     id = params.get('id');
14
15     if (id) {
16         loadLog(id);
17     }
18
19 }
20
21
22
23 async function loadLog(id) {
24     // Load the log
25     let log = await Log.fromId2(id);
26
27     const date = `${log.startTime.getMonth()}/${log.startTime.getDate()}/${log.startTime.getFullYear()}`;
28
29     // Update the page Titles
30     document.title = `Swift Log ${date}`;
31     document.getElementById('date').innerHTML = date;
32
33     // Add the count
34     document.getElementById('count').innerHTML = log.currentCount;
35
36     // Setup notes and weather fields
37     let notesField = new EditableField(
38         id,
39         'notes',
40         document.getElementById('notes'),
41         document.getElementById('editNotes'),
42         log.notes,
43         "No notes entered"
44     )
45
46     let weatherField = new EditableField(
47         id,
48         'weather',
49         document.getElementById('weather'),
50         document.getElementById('editWeather'),
51         log.weather,
52         "No weather information entered"
53     )
54
55     // document.getElementById('notes').innerHTML = log.notes ? log.notes : "No notes entered";
56     // document.getElementById('weather').innerHTML = log.weather ? log.weather : "No weather information entered";
57
58     // Update the edit log link
59     let editLogLink = document.getElementById('editLog');
60     if (log.logType == 'video') editLogLink.href = `videolog.html?id=${id}`;
61     else {
```

```
62     let parent = editLogLink.parentElement;
63     parent.parentElement.removeChild(parent);
64 }
65
66 // Download link
67 let downloadLogLink = document.getElementById('downloadLog');
68 downloadLogLink.href = `api/logtospreadsheet.php?id=${id}`;
69
70 // Output the raw data
71 let rawDataTable = document.getElementById('rawLog');
72 log.data.forEach(entry => {
73     rawDataTable.appendChild(buildRow([formatPlaybackTime(new Date(log.startTime.getTime() + entry.time)),
74 entry.count]))
75 })
76
77
78 // Get the count by minute
79 // let minuteCount = log.getCountByInterval(10);
80 let minuteCount = log.getTotalByInterval(60);
81
82 let chartData = minuteCount.map(entry => {
83     return [entry.time, entry.count, entry.runningTotal];
84 })
85
86 drawChart(chartData)
87
88 // Output the minute count
89 const tbody = document.getElementById('log');
90
91 minuteCount.forEach(minute => {
92     tbody.appendChild(buildRow([formatPlaybackTime(minute.time), minute.count, minute.runningTotal]))
93 })
94
95 }
96
97
98 function drawChart(tempData) {
99
100     var data = new google.visualization.DataTable();
101     data.addColumn('date', 'Time');
102     data.addColumn('number', 'Count');
103     data.addColumn('number', 'Running Total');
104
105     data.addRows(tempData);
106
107     var options = {
108         title: 'Recorded Data',
109         seriesType: 'bars',
110         series: {
111             0: {type: 'bar',
112                 targetAxisIndex: 0},
113             1: {type: 'line',
114                 targetAxisIndex: 1}
115         },
116         vAxes: {
117             0: {title: 'count'},
118             1: {title: 'total'}
119         }
120     };
121
122     var chart = new google.visualization.ComboChart(document.getElementById('dataChart'));
123
```

```
124     chart.draw(data, options);  
125 }  
126  
127 function downloadLog() {  
128  
129  
130  
}
```