**Activity**

Files: activity.html, **activity.js**, sleep.activity.js

1. Screen: Activity , updateData(data, false)

2. Screen: Activity > Detail, updateData(data, true)

isDetail=true \*\*[allow filtering : cry, asleep, awake, away]

updateData(data);

**Simulated Breathscope**

Files: breathscope.html, **sleep.bpm.js**

3. Screen: Status setBpm(newBPM);

**Live Breathscope**

Files: livebreathscope.html, **livebreathscope.js**

4. Screen: Status > Detail addBsData(datapoint);

**Analysis**

Files: quality.html, **quality.js**, sleep.quality.js

5. Screen: Analysis updateData(data, false)

6. Screen: Analysis > Detail updateData(data, true) \*\*[return more value, average sound and temperature]

**Detail Analysis**

Files: analysis.html, **analysis.js**, sleep.analysis.js

7. Screen: Analisis > Day > Detail,

**Week/Month bar chart**

Files: summary.html, **summary.js**, sleep.summary.js

8. Screen: Analysis > Week updateData(data, 'week');

9. Screen: Analysis > Month updateData(data, 'month');

**Day/Week/Month Calendar**

Files: calendar.html, **calendar.js**

10. Screen: Analysis > Day initCalendar('day', 2015, 4, 1 , 1, null);

11. Screen: Analysis > Week initCalendar('week', 2015, 4, 1 , 1, null);

12. Screen: Analysis > Month initCalendar('month', 2015, 4, 1 , 1, null);

you can open the html file to view the chart,

example of how to pass data to the chart can be found at the bottom of **the file(highlighted red)**

you can replace NativeBridge.call in the file which is the callback method to pass data back to iOS from Web View

For Android, could replace it with JavaScriptInterface for the webview to get data

For website, could pass a function to the chart and callback once data return