Open *Source* | Open *Possibilities*

Testing the W3C Battery API with WebSockets

Presented by: Giri Mandyam

Presentation Date: February 2013

QUALCOMM INNOVATION CENTER, INC.

Introduction

- DAP has essentially completed specification of Battery API
- Battery API is a great opportunity for developers to tune the performance of their pages
- One potential use of Battery API for developer to adjust messaging frequency based on battery state
  - Current methods are indirect, e.g. Page Visbility (http://www.w3.org/TR/page-visibility/) spec from Web Performance Working Group
    - Example of webpage changing its messaging frequency based on whether page is currently in foreground (i.e. visible)
- Effect of app-layer keep alive over WebSockets on battery life discussed in Mandyam & Ehsan, “HTML5 Connectivity Methods and Mobile Web Power Consumption”, W3C Web Performance Workshop, November 2012
  - http://www.w3.org/2012/10/Qualcomm-paper.pdf
  - Upshot is that accurate battery readings can also be used to modulate messaging frequency for polling applications and extend battery life

Proposal

- Test battery API implementations specifically for networked applications
- Create test(s) that involve WebSocket connection with repeated sending of messages
  - Would require ws-compatible server to establish WebSocket connection
- Run test unplugged
  - Battery API should accurately track battery level, or at least replicate at any given instant in time an accurate representation of native power level reading

Test

- Run on desktop (Lenovo Thinkpad 420) with Firefox browser (ver. 18.0.1) using Battery API
- Server is at localhost
  - Accessible at ws://127.0.0.1:8080
  - Apache container
  - Message sent (every 3 minutes) by webpage is looped back
- PC started at full charge and was unplugged
- Results
  - Readings of battery API were not consistent with system power meter
  - Sample test results

Time Lapsed (minutes)	Battery.level (JS)	System Power Meter
0	98%	98%
3	98%	96%
6	98%	94%
9	98%	92%

Discussion

- Support among desktop versions of popular browsers for Battery API may be spotty and untested
  - Battery API tends to be viewed as more critical for handheld device browsers
- Premature at this point to conclude that there is any kind of bug in the Mozilla implementation
- However, in general a browser that claims to support Battery API should be able to track system power meter
- Recommend adding a test to the Battery API test suite that requires networking
  - Would require human intervention to verify API is tracking system power meter
  - Would require presence of server
    - Localhost should be sufficient – message loopback
    - Tester can pick favorite container (e.g. Apache, WebSphere, etc.)

Thank You

QuIC
QUALCOMM INNOVATION CENTER, INC.