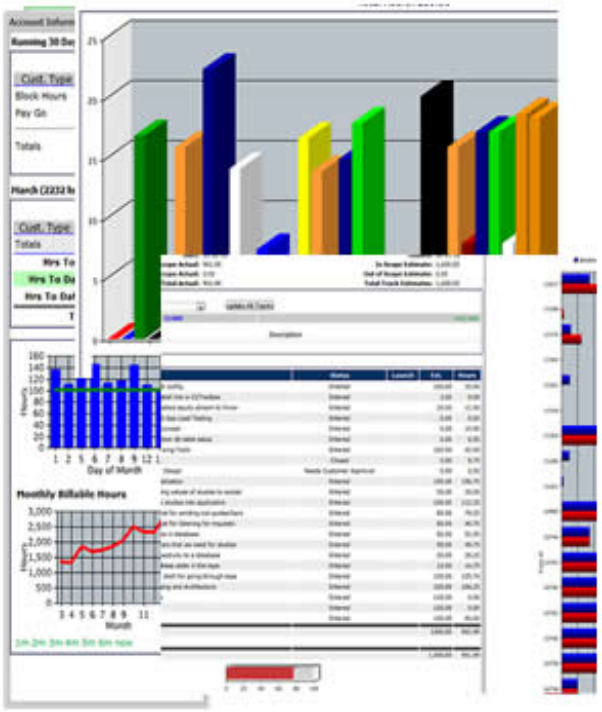
Web Sockets - Going Where No Muse Has Gone Before

Posted At : March 14, 2012 2:47 PM | Posted By : Mark Kruger
Related Categories: ColdFusion

You might wonder where I've been holed up for more than a month. Never fear - I'm still slogging away. My current project is a dashboard for **CF Webtools** that tracks all of our consulting activity. CF Webtools runs a custom tracking and management system as a core component of our business. This system has many cool features that have evolved over the years for tracking hours, performance, tasks, groups of tasks, assignments, productivity, estimates, deadlines etc. Myself and my VP, Jason Herbolsheimer (a brilliant programmer and manager who you would all know and love *if* he would ever blog), have spent thousands of hours on it to make it fit our business model of transparency, measurable productivity and cash positive block hours.



The reports and features it contains are useful, but a bit of a hodgepodge. Meanwhile, over time my role has changed considerably. Other than troubleshooting, mentoring and experimentation I'm not involved in day to day tasks directly with our clients any more. But I still monitor our overall productivity closely. Indeed, now that I'm a step back from the work I have a much better sense of what we are accomplishing and where our weak spots are. In the past I have used cfcharts called up on internal pages to show hours and individual developer performance but 3 things had changed.

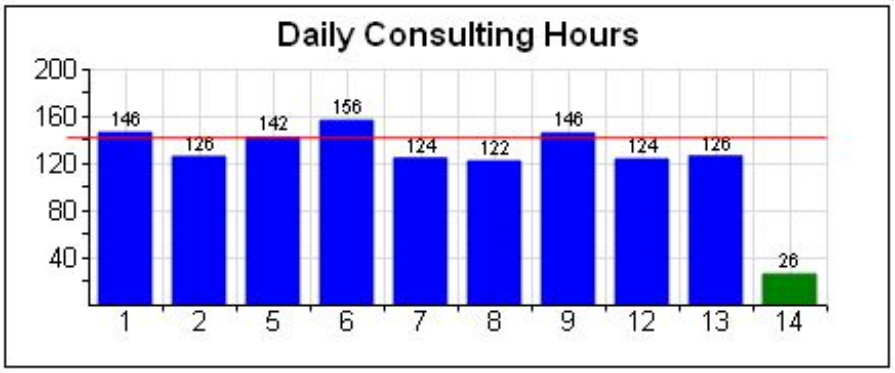
1. We now have many more developers to track (29 at last count).
2. With 3000+ consulting hours per month the system receives a constant stream of updates, notes etc.
3. I have a fancy new set up with 3x27 inch monitors plus a 46 inch wall mounted monitor that looked lonely and I wanted some fancy-pants dashboard to display

Ok, that last reason is simple hubris but still, it was good enough to boot me down the path. But I had some decisions to make.

Ajax, vs. Web Sockets

First, should I implement the dashboard using Ajax and simply "poll" the site for updates every couple of minutes? After all developers were logging 140+ hours per day in increments of .25 hrs to 3 or 4 hours at a time. A couple minute refresh would not be a problem and it would seem pretty real time to me. My other option was web sockets with which I had zero experience. I had read some excellent blog posts by **Ben Nadel** on web sockets. Let me stop and say I simply loath him. He's so smart and his code is so perfect. I always feel like poor white trash peeking in on Rockefeller when I read his blog. Of course I jest... love you Ben. Someday I want to be in your banner pictures along with the other geeks you take pics of. Maybe at CF Objective. Anyway, moving on from my man-crush... I had read Ben's blogs and **Ray** also had some posts on the topic. I began to wonder if I might use web sockets for my dashboard instead of Ajax polling.

Meanwhile I started exploring some charting options. Now I love cfchart. It produces nice looking graphs, it's easy to use and it makes throwing up an extra visual inside of a report fairly trivial. For this dashboard however, I had big plans. I was going to put about 6 charts that would be main features. One chart would be our "daily consulting hours" chart. It shows hours for each day with the current day in green and a ceiling line showing the estimated max. It looks like this:



Along with that chart I wanted a minute by minute chart showing logging that would indicate our busiest time and show how and when our developers were the most active (or at least when they were logging). Finally I had some customer specific charts showing how many hours are being logged and enabling me to see who our biggest client contributors are.

Naturally I could do all this in CF, but I would probably be forced to resort to an iFrame and a periodic refresh. Even though I could trigger the refresh the idea of sockets was growing on me. I also wanted some tabular data relating to the "tiers" of our customers, internal hours, and per user daily and weekly totals for our team of developers. It seemed like a lot of work doing all that in Ajax - at least as much work as trying something new. Plus, the charts would not look like our fancy streaming quotes charts (CF Webtools does some amazing work with quote feeds for some of our customers). Instead it would look like little windows refreshing within the page. I wanted to see my chart grow in front of my eyes. And this meant at a minimum, something *other* than cfchart.

What I found was called **rGraph**. Rgraph is a suite of HTML5 libraries for graphing. It's heavily dependent on HTML5 "canvas" objects for drawing, so it's not suitable for public facing apps where you can't predict the browser or what it supports. But it works really well in Chrome (my favorite) and Firefox (My second favorite - mostly due to firebug). Seeing as how my dashboard would be just for me and my management team I could

specify the environment so this restriction was not a problem. It turns out rGraph is an extremely powerful, easy to use group of libraries.

So I started out and built a websocket application where the data is almost entirely fed through socket messaging. The final piece of the puzzle was of course Json. Using CF's Json functions I could feed my client side charting so easily that, once I figured out the init syntax, it was practically as easy as CF Chart. Json really abstracts the difference between server and client side and makes projects like this more about figuring out *what* to pass and not about *how* to pass or how to *format*. I found with a couple little snippets on the client and the server I could exchange structured data easily between my charts and the server. And the charts ended up being extremely fast - even to draw a brand new one with a freshly messaged dataset.

Final Result Preview

So the final result has full interactive client side charting - including calling for new socket messages to draw new charts using click events that introspect objects of an existing chart (whew). The dashboard receives about 20 different types of messages. Here's the list of the final features.

- Socket authentication
- Hours logging from direct entry - updates summary charts, developer charts, customer charts and tabular data
- SVN Logged Hours - Developers can log hours directly using a comment template when updating SVN. We pick up these entries and message them to the dashboard along with the hours as well.
- Notes to tracking - updates scrolling log and appears temporarily in "raw log" area.
- Admin notes - notes seen only by development, pm and team members. Updates scrolling notes.
- Status changes
- SVN updates - updating any SVN repo causes a message to be logged to the socket and appear in our scrolling feed along with the number of files updated.
- Network alerts - our alert system fires messages to the socket to tell us something is down. Displayedi n our scrolling feed.
- General messages - Anyone can add a message with a level of importance to the feed where it shows.

Final Result

The final result looks like this:

Once loaded it tracks all activity throughout the day and updates in real time. It looks great on my fancy LED monitor (I can see my office mates rolling their eyes as I speak).

In the next few posts I will be writing about this task and the hurdles I overcame. I'm also submitting a broad version of the series to Adobe so I'm not sure which content will end up where. I'm actually quite energized about this topic and I will be speaking on it at the coming **NE CFUG** meeting on March 27th. My next task will be to migrate the app to CF 10 and see what CF's new socket capabilities bring to the table.