Math, PhantomJS and Command Line Arguments

Posted At : September 11, 2014 11:21 AM **|** Posted By : Cutter
Related Categories: PhantomJS, Development, NodeJS, JavaScript

So, recently I've been playing with **PhantomJS**. For those who are unfamiliar, PhantomJS is a headless webkit engine, that you can use to perform automated testing, take screenshots, and dozens of other things.

Phantom is still fairly new, and this has been a learning experience. My first trials were using the **phantomjs-node** module for NodeJS. I soon discovered (with some assist) that this bridge has some performance issues of it's own, and had to move to the command line.

When working with PhantomJS in this fashion, you write a script (in JavaScript), and pass it to PhantomJS along with any necessary arguments. I call this command line from NodeJS directly, but you can do so in Terminal or the cmd shell if you want.

Here was the catch. I had some math in my PhantomJS script. Not even complex math, really, just some basic equations converting milliseconds to seconds and rounding it off:

```
var system = require('system'),
           args = system.args,
           pageArgs = {
               elapsedTime: args[6] // this value is in the command line as 13762
           };

     var totalTime = Math.round((pageArgs.elapsedTime)/1000);
     console.log("The answer is not " + totalTime);
```

Now, you and I know the answer is 14, but PhantomJS didn't seem to like that:

The answer is not 0

If I hard code the numeric value of elapsedTime (13762) into that statement then the math parses correctly. It appears that performing math on the values passed in causes the issue. I can change up the equation by adding 250 milliseconds:

```
var totalTime = Math.round((pageArgs.elapsedTime+250)/1000);
```

But PhantomJS still doesn't like it:

The answer is not 413

Crazy, right? What I found was, if I needed to do any math to set some variables for use in the script, then I needed to do those in my node process, and pass the evaluated values to my PhantomJS script as arguments in the command line.

REVISION NOTE: Further testing shows that the command line arguments will all come across as a "string" type, which may be why these equations were miss firing. Remember to explicitly convert your command line arguments to the necessary data type, to avoid this type of issue.