hacksparrow.com

Keep Node.js script running after logging out from shell

by Hack Sparrow • original (http://www.hacksparrow.com/keep-node-js-script-running-after-logging-out-from-shell.html)

Those new to Node.js have almost no problem getting started with simple Node scripts. Even creating a relatively complex Express (http://expressjs.com/) app is pretty straightforward. But they are almost always left clueless on how to keep their script or app running after they log out from the shell.

The solution to keeping Node scripts running on a logged out shell is a very easy one - a Node.js module called Forever (https://github.com/nodejitsu/forever). It starts the script for you and keeps an eye on it thereafter - restarts it if crashes because of any reason, and optionally maintain all sort of logs for it.

Forever should be installed as global Node module:

```
$ npm install forever -g
```

if that throws a permission error, you'll need to install using sudo:

```
$ sudo npm install forever -g
```

Once installed, you can run the script this way:

```
$ forever start app.js
```

Make sure to confirm your code is not buggy and the script is working fine before you use Forever to start it as a daemon (http://en.wikipedia.org/wiki/Daemon_%28computing%29). As of this writing, Forever doesn't seem to be caring if a script is compile-time exception free or not, it will print

```
info: Forever processing file: app.js
```

whether the script actually failed to start or not!

There are two other useful options you can specify:

-o - output log
-e - error log

Example:

```
$ forever -o out.log -e err.log start app.js
```

All `console.log()` messages will be captured on out.log file, and all error messages on err.log file. Very useful for debugging and analysis, highly recommended.

There is much more to Forever than what I have covered here. There are many more options, and other ways Forever can be used. Read the Forever README (https://github.com/nodejitsu/forever/blob/master/README.md), for the details.

It must be noted that it is not just about keeping Node scripts running on the shell after logging out, if that was the case, a detached screen (http://linux.die.net/man/1/screen) or nohup (http://en.wikipedia.org/wiki/Nohup) would have sufficed our needs, a lot more is involved when considering running a Node script continuously. Forever provides various logging options and most importantly, it restarts the script if it crashes for some reason, that's why it is preferred over 'backgrounding' techniques.

Having said that, Forever is not the solution to all keep-running requirements. As of this writing, it doesn't even restart the scripts after a reboot. For more advanced requirements check out these options:

- Monit (http://mmonit.com/monit/)
- Upstart (http://upstart.ubuntu.com/)
- Launchtool (http://people.debian.org/~enrico/launchtool.html)
- Daemontools (http://cr.yp.to/daemontools.html)

Getting back to Forever, here are some common commands:

To see all the scripts run by Forever:

```
$ forever list
```

To stop a script:

```
$ forever stop app.js
```

To stop all scripts:

```
$ forever stopall
```

To restart a script:

```
$ forever restart app.js
```

With this I come to the end of the tutorial on keeping Node scripts running after logging out from the shell. I hope I helped you in getting started with Forever, and pointed you to the right direction if Forever doesn't meet your requirements.

Original URL:
http://www.hacksparrow.com/keep-node-js-script-running-after-logging-out-from-shell.html