AMERICA'S FINEST OPEN SOURCE REPOS.

INSTALLATION AND SETUP

To build and run the project you need to have Yeoman installed (with all it's dependencies). Click on the link to see how to get started. TL;DR if you've got Node and npm installed, then `npm install -g yo bower grunt-cli grunt` should get you where you need to go.

After unzipping the project, you will need to run `npm install && bower install` from the directory to install all the required node modules and bower packages. You might need to run these commands with admin rights: `sudo npm install` and `sudo bower install` and then type in your password.

When this is done you can start up a local server with livereload and file watching by running `grunt serve`.

To create a production build (image optimised, JS minified and concatenated, Sass compiled to CSS, etc.) run `grunt` to output a production-ready build into the /dist directory. This can be uploaded to a server or run from a local server (note: simply double-clicking on `index.html` won't work correctly due to file pathing and potential ajax security rules, etc.).

Finally, if you're really ambitious there's a Grunt task that builds and rsyncs the app to a server. Simply replace the server address/file path the `config` section of the Gruntfile.js (starts on line 20) with your own information and run `grunt deploy`.

If any of the above isn't working for whatever reason, please do not hesitate to try to contact me. The project is also running online at http://onion.prayingmadness.com if all else fails.

NOTES

Harnessing the power of the Github API this app serves up all the public repository information from The Onion's development team, sorted by the most recently-updated repos. Selecting a repo will show a list of the most recent 100 commits, sorted by most recent.

(Side note: I *seriously* considered including fartscroll.js but thought wiser of it in the end. You're welcome.)

This app was built based on a vanilla Yeoman `yo webapp` installation with Backbone added afterwards. I could have used other methods of setting this up (Backbone-based generators, etc.) but I'm familiar with the Yeoman setup already and any Backbone generators I've found were old, outdated, and unmaintained for over a year.

It's worth noting that at this point I (somewhat embarrassingly) have had *very* little hands-on experience with Backbone or MVC programming in general. This, of course, begs the question as to why I built the app this way. To be honest it was a lot more work than expected, and took far longer than building it otherwise should have. I'm confident that, had I built this without using Backbone, I could have knocked this out in a couple of hours. However, the resultant app from that type of exercise would have been almost completely inflexible and unable to expand/grow to meet future needs and features without a not-insignificant degree of re-engineering.

There's no automated testing built into the app, but what I have put in is a "debug" mode. Running the app with `#debug` appended to the URL will output a number of different pieces of debugging info to the browser console (data retrieved from the server, parameters sent with requests, etc.). As well, as is best practice all ajax requests will display error messages if a request is unsuccessful – feel free to hack about with the fetch() URLs to see this in action: RepoItems.js:3 (inital repo list download), ReposView:26 (load more repos), and CommitsView:18 (load commits).

In keeping with the spirit of the assignment ('I need to put something up that my users will actually be using on a daily basis, but I don't have days and days to work on it') here's a (non-exhaustive) list of things that I didn't do but would absolutely consider for a "phase 2" of the project, roughly in order of priority:

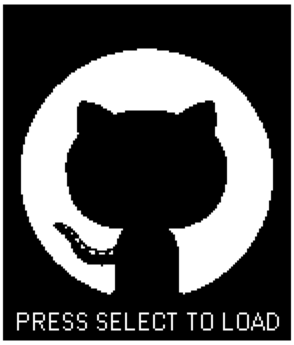
- Automated testing with a framework such as Jasmine or Mocha
- Use Require.js to load everything asynchronously, stop polluting the global namespace, and provide a more robust `r.js` -based build script
- Allow the user to configure which organization/account to fetch repository data from
- Selecting a commit would show a list of files changed in that commit
- Allowing the user to change the metric by which repositories are listed (number of forks, most open issues, etc.)
- Wider range of browser support. So far this has only been tested in Chrome/OS X. All functionality *should* work elsewhere but the layout completely cracks in mobile browsers. FYI.

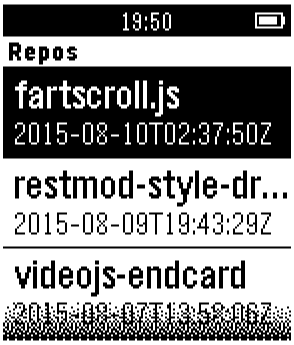
Oh, and as an extra added feature:

PEBBLE BONUS ROUND

Yep, this also exists as an app for your Pebble® smartwatch. Why? Why *not*?

- Source code: https://github.com/humantorch/github-pebble
- Install: http://apps.getpebble.com/en_US/application/55c8e993146a72a0ea00002d





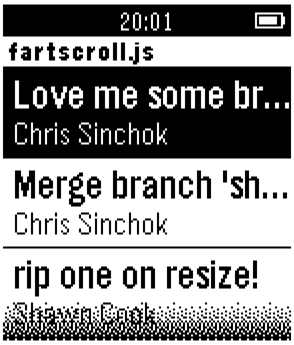
19:50

Repos

fartscroll.js
2015-08-10T02:37:50Z

restmod-style-dr...
2015-08-09T19:43:29Z

videojs-endcard
2015-08-07T13:52:06Z



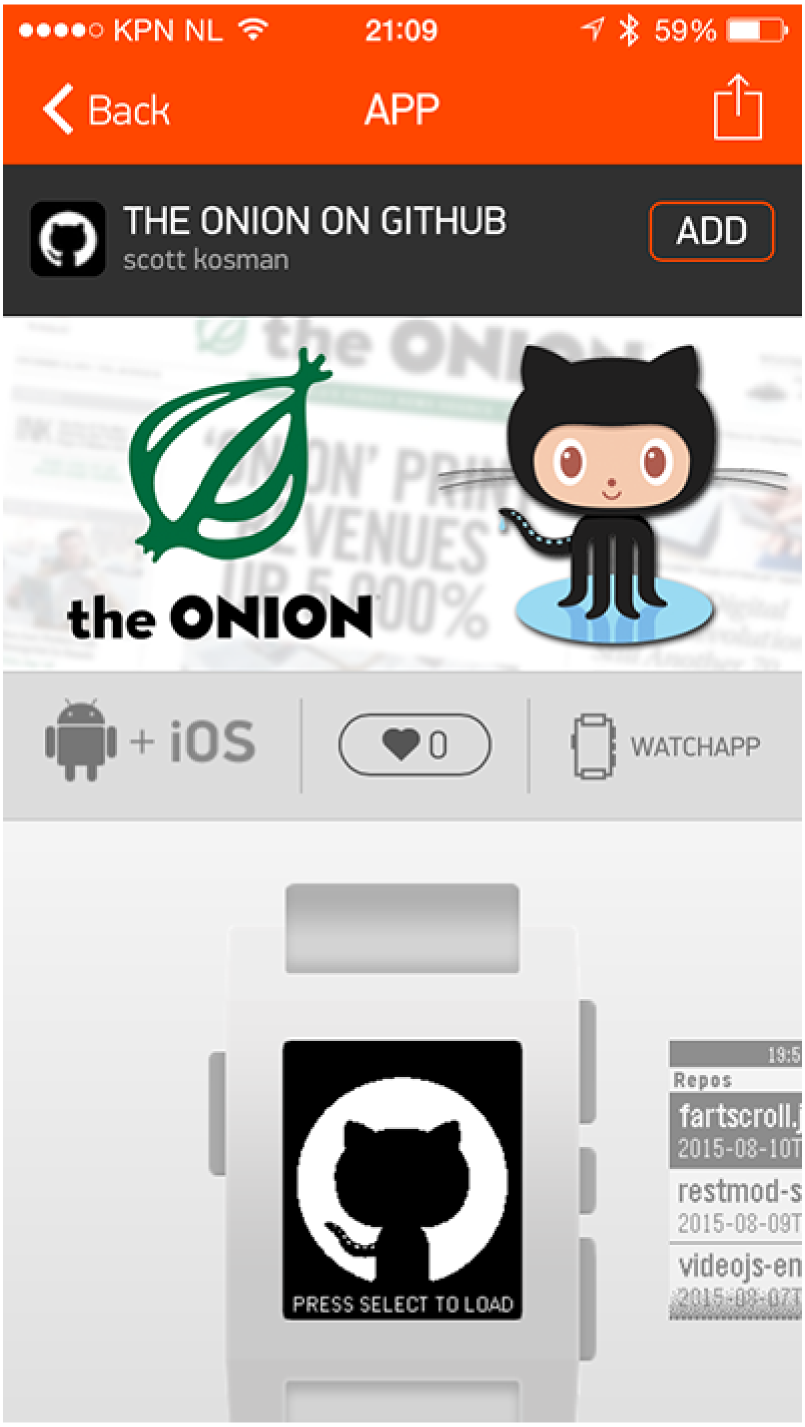
20:01

fartscroll.js

Love me some br...
Chris Sinchok

Merge branch 'sh...
Chris Sinchok

rip one on resize!
Shawn Cook



THE ONION ON GITHUB
scott kosman