Grunt

<http://gruntjs.com/getting-started>

Grunt is a task manager, do not get this confused between NPM or Bower,

Grunt: task manager

Npm: server side dependency manager

Bower: client side dependency manager

What is Grunt

Getting Started

Browserify

**What is Grunt**

Grunt is a cross platform task runner for your project, Similar to what NPM and Gulp can do, Grunt uses JavaScript and can be used easily by selecting the appropriate plugin and installing and configuring it. Grunt has a good community following, and there are many NPM plugins that have a grunt plugin available.

Grunt Tasks read/write to/from disk, keep this in mind as many tasks might require to read/write to disk, this can cause grunt to run slow. While with NPM, does it differently by using bash to pipe output into the next command.

Gulp relies heavily on streams, pipes and asynchronous code, which fits hand in hand with Node unlike NPM and grunt. Grunt seems to be the easiest method as most developers working on the client side and server side, will know and be using JavaScript. Also for big teams grunt also has the advantage of more of the team members knowing how to use it, rather than having to specialize in Node.

Grunt: <http://gruntjs.com/getting-started>

Gulp: <http://gulpjs.com/>

Comparing NPM, Grunt, Gulp: <https://ponyfoo.com/articles/choose-grunt-gulp-or-npm>

**Getting Started**

Start of by installing the command line interface for grunt (grunt-cli) from NPM, this will install the grunt command line interface and will allow us to call grunt from anywhere in our project (use grunt globally with **“-g”**).

Command: **“npm install grunt-cli”**



Install these 3 modules to get up and running, we need grunt initially, but also install

**“grunt-contrib-jshint”,**

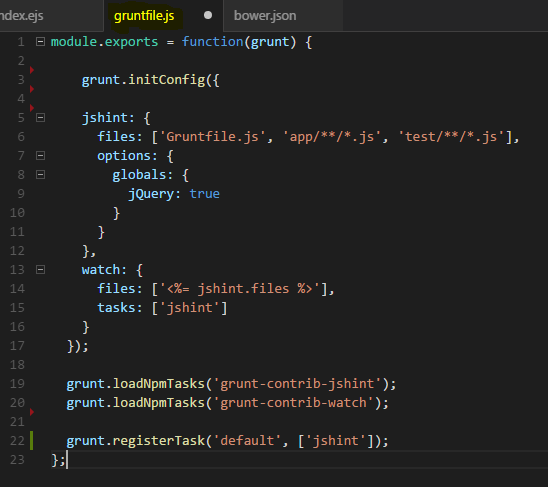
**“grunt-contrib-watch”**

as these will come in handy later on.

Once we have these dependacies installed and updated in our package.json we can create the gruntfile.js now.

This is a simple example of a grunt file using the dependacies mentioned above.



In this example we have 2 tasks currently, jshint and watch, since we installed these 2 dependacies also, if we want to use them we must load the task plugin by:

**“grunt.loadNpmTasks('grunt-contrib-watch');”**

You may create custom tasks inside this grunt file, some may require you to install more dependacies or load more npm tasks.

**Browserify**

With our app, we specify all js files in the browser, this includes all angular directives, controllers, factories, filters and even the dependacies. A good way to do this is to use a plugin like “Browserify” this will bundle all the apps files and dependacies into one or many files, which then can be used in the index.ejs page.

Info @ [**https://www.npmjs.com/package/grunt-browserify**](https://www.npmjs.com/package/grunt-browserify)

Browserify will collate all the files needed for loading into the browser into one file. Or mutiple if you wish.

For this app we are going to create 2 files, one for the app code, and another for all the dependacies.

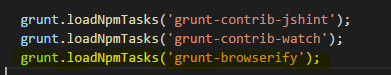
To do this we must install browserify:



Once we have this done and the package.json updated we can start to edit the gruntfile.js

Add in this new line of code:

“grunt.loadNpmTasks('grunt-browserify');”



Now we can begin to add our browserify task and configure it.



Start of by creating the task “Browserify“ and we have here 2 tasks within it. This is a basic example to get browserify to create separate files, but if more info is needed can be found

@ <https://www.npmjs.com/package/grunt-browserify>

Inside the client task we are telling browserify what file is our entry-point to the app, here it is the app.js

**src: ['./app/client/app.js'],**

And then we specify the destination where we want to save the file. We must give this task a list of the external files also.

**dest: './app/client/public/app.js',**

**external:[**

**'angular',**

**'bootstrap',**

**"templates",**

**]**

\*May ask why are we having separate file?

-To get correct coverage of our app for testing

-Faster build times for dev, as you only need to create one file when source code is changed

Inside the deps task we do not need to tell it of any source files, but we must give it the location of where the dependencies are kept for it to include them in the output file.