**NodeJS**

1. Go to  [https://nodejs.org](https://nodejs.org/)
2. Check version with node –v and npm –v

**Less**

1. npm install -g less
2. To compile the file: lessc mystyles.less > mystyles.css

**Bower**

1. npm install -g bower
2. bower init
3. Installing bower components: bower install bootstrap –S, bower install font-awesome –S, …
4. Update links in files, i.e. <link href="bower\_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">

**Grunt**

1. install grunt cli:

npm install -g grunt-cli

1. Create package.json in project folder with the following:

{

"name": "conFusion",

"private": true,

"devDependencies": {},

"engines": {

"node": ">=0.10.0"

}

}

1. Install grunt into project folder:

npm install grunt --save-dev

1. Create Gruntfile.js. The JSHint task is set to examine all the JavaScript files in the app/scripts folder, and the Gruntfile.js and generate any reports of JS errors or warnings. We have set up a Grunt task named build, that runs the jshint task and the build task is also registered as the default task (Code at the end).
2. Into html surround css imports with:

<!-- build:css styles/main.css -->

<link href="../bower\_components/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">

<link href="../bower\_components/bootstrap/dist/css/bootstrap-theme.min.css" rel="stylesheet">

<link href="../bower\_components/font-awesome/css/font-awesome.min.css" rel="stylesheet">

<link href="styles/bootstrap-social.css" rel="stylesheet">

<link href="styles/mystyles.css" rel="stylesheet">

<!-- endbuild -->

1. Also for js files:

<!-- build:js scripts/main.js -->

<script src="../bower\_components/angular/angular.min.js"></script>

<script src="scripts/app.js"></script>

<!-- endbuild -->

1. Install JSHint:

npm install grunt-contrib-jshint --save-dev

npm install jshint-stylish --save-dev

npm install time-grunt --save-dev

npm install jit-grunt --save-dev

1. Create .jshintrc file:

{

"bitwise": true,

"browser": true,

"curly": true,

"eqeqeq": true,

"esnext": true,

"latedef": true,

"noarg": true,

"node": true,

"strict": true,

"undef": true,

"unused": true,

"globals": {

"angular": false

}

}

1. You can test it with “grunt” command.
2. Install copy and clean modules:

npm install grunt-contrib-copy --save-dev

npm install grunt-contrib-clean --save-dev

1. Install modules to prepare dist folder:

npm install grunt-contrib-concat --save-dev

npm install grunt-contrib-cssmin --save-dev

npm install grunt-contrib-uglify --save-dev

npm install grunt-filerev --save-dev

npm install grunt-usemin --save-dev

1. Install watch and connect modules:

npm install grunt-contrib-watch --save-dev

npm install grunt-contrib-connect --save-dev

1. Run “grunt serve” to start browser.

Example of how the Gruntfile should look:

'use strict';

module.exports = function (grunt) {

// Time how long tasks take. Can help when optimizing build times

require('time-grunt')(grunt);

// Automatically load required Grunt tasks

require('jit-grunt')(grunt, {

useminPrepare: 'grunt-usemin'

});

// Define the configuration for all the tasks

grunt.initConfig(

{

pkg: grunt.file.readJSON('package.json'),

// Make sure code styles are up to par and there are no obvious mistakes

jshint:

{

options:

{

jshintrc: '.jshintrc',

reporter: require('jshint-stylish')

},

all:

{

src:

[

'Gruntfile.js',

'app/scripts/{,\*/}\*.js'

]

}

},

useminPrepare:

{

html: 'app/menu.html',

options:

{

dest: 'dist'

}

},

// Concat

concat:

{

options:

{

separator: ';'

},

// dist configuration is provided by useminPrepare

dist: {}

},

// Uglify

uglify:

{

// dist configuration is provided by useminPrepare

dist: {}

},

cssmin:

{

dist: {}

},

// Filerev

filerev:

{

options:

{

encoding: 'utf8',

algorithm: 'md5',

length: 20

},

release:

{

// filerev:release hashes(md5) all assets (images, js and css )

// in dist directory

files:

[{

src:

[

'dist/scripts/\*.js',

'dist/styles/\*.css',

]

}]

}

},

// Usemin

// Replaces all assets with their revved version in html and css files.

// options.assetDirs contains the directories for finding the assets

// according to their relative paths

usemin:

{

html: ['dist/\*.html'],

css: ['dist/styles/\*.css'],

options:

{

assetsDirs: ['dist', 'dist/styles']

}

},

copy:

{

dist:

{

cwd: 'app',

src: [ '\*\*','!styles/\*\*/\*.css','!scripts/\*\*/\*.js' ],

dest: 'dist',

expand: true

},

fonts:

{

files:

[

{

//for bootstrap fonts

expand: true,

dot: true,

cwd: 'bower\_components/bootstrap/dist',

src: ['fonts/\*.\*'],

dest: 'dist'

},

{

//for font-awesome

expand: true,

dot: true,

cwd: 'bower\_components/font-awesome',

src: ['fonts/\*.\*'],

dest: 'dist'

}

]

}

},

watch:

{

copy:

{

files: [ 'app/\*\*', '!app/\*\*/\*.css', '!app/\*\*/\*.js'],

tasks: [ 'build' ]

},

scripts:

{

files: ['app/scripts/app.js'],

tasks:[ 'build']

},

styles:

{

files: ['app/styles/mystyles.css'],

tasks:['build']

},

livereload:

{

options:

{

livereload: '<%= connect.options.livereload %>'

},

files:

[

'app/{,\*/}\*.html',

'.tmp/styles/{,\*/}\*.css',

'app/images/{,\*/}\*.{png,jpg,jpeg,gif,webp,svg}'

]

}

},

connect:

{

options:

{

port: 9000,

// Change this to '0.0.0.0' to access the server from outside.

hostname: 'localhost',

livereload: 35729

},

dist:

{

options:

{

open: true,

base:

{

path: 'dist',

options:

{

index: 'menu.html',

maxAge: 300000

}

}

}

}

},

clean:

{

build:

{

src: [ 'dist/']

}

}

});

grunt.registerTask('build', [

'clean',

'jshint',

'useminPrepare',

'concat',

'cssmin',

'uglify',

'copy',

'filerev',

'usemin'

]);

grunt.registerTask('serve',['build','connect:dist','watch']);

grunt.registerTask('default',['build']);

};