



http://mherman.org/blog/2014/08/14/kickstarting-angular-with-gulp/#.WLGIOFlEnIV

Kickstarting Angular With Gulp and Browserify, Part 1 - Gulp and Bower

AUG 14TH, 2014 8:17 AM

Let’s develop an Angular boilerplate. Why? Despite the plethora of Angular seeds/generators/templates/boilerplates/starters/etc. on Github, none of them will ever do *exactly* what you want unless you build your own, piece by piece. By designing your own, you will better understand each component as well as how each fits into the greater project. Stop fighting against a boilerplate that just doesn’t fit your needs and start from scratch. Keep it simple, as you learn the process.

**In this first part, we’ll start with Angular and Gulp, getting a working project setup. Next**[**time**](http://mherman.org/blog/2014/08/15/kickstarting-angular-with-gulp-and-browserify-part-2/#.U-4co4BdUZ0)**we’ll add Browserify into the mix.**

*This tutorial assumes you have Node.js installed and have working knowledge of NPM and Angular. Just want the code? Get it*[*here*](https://github.com/mjhea0/angular-gulp-browserify-seed)*.*

**Project Setup**

**Install Dependencies**

**Setup a project folder and create a *package.json* file:**

|  |  |
| --- | --- |
| 1  2 | $ mkdir project\_name && cd project\_name  $ npm init |

The npm init command helps you create your project’s base configuration through an interactive prompt. Be sure to update the ‘entry point’ to ‘gulpfile.js’. You can just accept the defaults on the remaining prompts.

Do the same for Bower:

|  |  |
| --- | --- |
| 1 | $ bower init |

Accept all the defaults. After the file is created update the ignore list:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | "ignore": [  "\*\*/.\*",  "node\_modules",  "app/bower\_components",  "test",  "tests"  ], |

**Install global dependencies:**

|  |  |
| --- | --- |
| 1 | $ npm install -g gulp bower |

**Bower install directory**

You can specify where you want the dependencies (commonly known as bower components) installed to by adding a *.bowerrc* file and adding the following code:

|  |  |
| --- | --- |
| 1  2  3 | {  "directory": "/app/bower\_components"  } |

**Install local dependencies:**

*NPM*

|  |  |
| --- | --- |
| 1 | $ npm install gulp bower gulp-clean gulp-jshint gulp-uglify gulp-minify-css gulp-connect --save |

*Bower*

|  |  |
| --- | --- |
| 1 | $bower install angular angular-animate angular-route jquery animate.css bootstrap fontawesome --save |

*The --save flag adds the dependencies to the package.json and bower.json files, respectively.*

We’ll address each of these dependencies shortly. For now, be sure you understand the project’s core dependencies:

* [**Gulp**](http://gulpjs.com/) is a Javascript task runner, used to automate repetitive tasks (i.e., minifying, linting, testing, building, compiling) to simplify the build process.
* [**Bower**](http://bower.io/) manages front-end dependencies.

**Folder Structure**

Let’s setup a base folder structure:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | .  ├── app  │   ├── bower\_components  │   ├── css  │   │  └── main.css  │   ├── img  │   ├── index.html  │   ├── partials  │   │   ├── partial1.html  │   │   └── partial2.html  │   └── js  │   │   └── main.js  ├── .bowerrc  ├── .gitignore  ├── bower.json  ├── gulpfile.js  ├── node\_modules  └── package.json |

Add the files and folders not already included. This structure is based on the popular [Angular Seed](https://github.com/angular/angular-seed) boilerplate, developed by the Angular team.

**Gulp**

To start, we just need the following code:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | // gulp  var gulp = require('gulp');  // plugins  var connect = require('gulp-connect');  gulp.task('connect', function () {  connect.server({  root: 'app/',  port: 8888  });  }); |

This allows us to serve our future Angular app on a development server running on port 8888.

**Test**

Let’s test it out. Add the word ‘hi’ to the *index.html* file, then run the following command:

|  |  |
| --- | --- |
| 1 | $ gulp connect |

Navigate to <http://localhost:8888/> and you should see ‘hi’ staring back at you. Let’s build a quick sample app. Keep the server running…

**Develop a Sample App**

***index.html***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | <!DOCTYPE html>  <html ng-app="SampleApp">  <head lang="en">  <meta charset="utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta name="description" content="">  <meta name="author" content="">  <title>Angular-Gulp-Browserify-Starter</title>  <!-- styles -->  <link rel="stylesheet" href="bower\_components/bootstrap/dist/css/bootstrap.css"/>  <link rel="stylesheet" href="bower\_components/fontawesome/css/font-awesome.css"/>  <link rel="stylesheet" href="bower\_components/animate.css/animate.css"/>  <link rel="stylesheet" href="css/main.css"/>  </head>  <body>  <div class="container">  <h1>Angular-Gulp-Browserify-Starter</h1>  <!-- views -->  <div ng-view></div>  </div>  <!-- scripts -->  <script src="bower\_components/jquery/dist/jquery.js"></script>  <script src="bower\_components/angular/angular.js"></script>  <script src="bower\_components/angular-route/angular-route.js"></script>  <script src="bower\_components/angular-animate/angular-animate.js"></script>  <script src="bower\_components/bootstrap/dist/js/bootstrap.js"></script>  <script src="js/main.js"></script>  </body>  </html> |

This should look familiar. The ng-app directive initiates an Angular app while ng-view sets the stage for routing.

***main.js***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | (function () {  'use strict';  angular.module('SampleApp', ['ngRoute', 'ngAnimate'])  .config([  '$locationProvider',  '$routeProvider',  function($locationProvider, $routeProvider) {  $locationProvider.hashPrefix('!');  // routes  $routeProvider  .when("/", {  templateUrl: "./partials/partial1.html",  controller: "MainController"  })  .otherwise({  redirectTo: '/'  });  }  ]);  //Load controller  angular.module('SampleApp')  .controller('MainController', [  '$scope',  function($scope) {  $scope.test = "Testing...";  }  ]);  }()); |

Again, this should be relatively straightforward. We setup the basic Angular code to establish a route handler along with a controller that passes the variable test to the template.

***partial1.html***

Now let’s add the partial template:

|  |  |
| --- | --- |
| 1 | <p>{{ test }}</p> |

**Test**

Back in your browser, refresh the page. You should see the text:

|  |  |
| --- | --- |
| 1  2  3 | Angular-Gulp-Browserify-Starter  Testing... |

**Create the Build**

Now that our app is working locally, let’s modify our *gulpfile.js* to generate a deployable build. Kill the server.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | // gulp  var gulp = require('gulp');  // plugins  var connect = require('gulp-connect');  var jshint = require('gulp-jshint');  var uglify = require('gulp-uglify');  var minifyCSS = require('gulp-minify-css');  var clean = require('gulp-clean');  var runSequence = require('run-sequence');  // tasks  gulp.task('lint', function() {  gulp.src(['./app/\*\*/\*.js', '!./app/bower\_components/\*\*'])  .pipe(jshint())  .pipe(jshint.reporter('default'))  .pipe(jshint.reporter('fail'));  });  gulp.task('clean', function() {  gulp.src('./dist/\*')  .pipe(clean({force: true}));  });  gulp.task('minify-css', function() {  var opts = {comments:true,spare:true};  gulp.src(['./app/\*\*/\*.css', '!./app/bower\_components/\*\*'])  .pipe(minifyCSS(opts))  .pipe(gulp.dest('./dist/'))  });  gulp.task('minify-js', function() {  gulp.src(['./app/\*\*/\*.js', '!./app/bower\_components/\*\*'])  .pipe(uglify({  // inSourceMap:  // outSourceMap: "app.js.map"  }))  .pipe(gulp.dest('./dist/'))  });  gulp.task('copy-bower-components', function () {  gulp.src('./app/bower\_components/\*\*')  .pipe(gulp.dest('dist/bower\_components'));  });  gulp.task('copy-html-files', function () {  gulp.src('./app/\*\*/\*.html')  .pipe(gulp.dest('dist/'));  });  gulp.task('connect', function () {  connect.server({  root: 'app/',  port: 8888  });  });  gulp.task('connectDist', function () {  connect.server({  root: 'dist/',  port: 9999  });  });  // default task  gulp.task('default',  ['lint', 'connect']  );  gulp.task('build', function() {  runSequence(  ['clean'],  ['lint', 'minify-css', 'minify-js', 'copy-html-files', 'copy-bower-components', 'connectDist']  );  }); |

**What’s happening here?**

1. [gulp-jshint](https://github.com/spenceralger/gulp-jshint) checks for code quality in the JS files. If there are any issues the build fails and all errors output to the console.
2. [gulp-clean](https://github.com/peter-vilja/gulp-clean) removes the entire build folder so that we start fresh every time we generate a new build.
3. [gulp-uglify](https://github.com/terinjokes/gulp-uglify) and [gulp-minify-css](https://github.com/jonathanepollack/gulp-minify-css) minify JS and CSS, respectively.

**Build commands**

**Default**

The default task, gulp, is a compound task that runs both the lint and connecttasks. Again, this just serves the files in the “app” folder on <http://localhost:8888/>.

**Build**

The build task creates a new directory called “dist”, runs the linter, minifies the CSS and JS files, and copies all the HTML files and Bower Components. You can then see what the final build looks like on <http://localhost:9999/> before deployment. You should also run the clean task before you generate a build.

Test this out:

|  |  |
| --- | --- |
| 1 | $ gulp build |

**Conclusion**

For angular:

<http://devdocs.io/angularjs~1.5/api/ng/directive>

<https://angular.io/docs/js/latest/>

<https://docs.angularjs.org/guide/concepts>