AngularJS is a structural framework for dynamic web apps. But how can you properly begin a new AngularJS project? I believe, many developers had to answer that question. Let’s figure it out.

Rule #1: Forget about certain libraries

The first rule you need to accept and follow throughout the whole project:

> *“If you're working on the project implemented in AngularJS, forget about libraries like jQuery.”*

If you need something from this library - it means you’re doing something wrong. I know, this statement may quite likely cause a surge of discontent:

— But I want to use twitter bootstrap in my project! It pulls jQuery, everything is so convenient with it, no need to write the code for notifications, pop-ups, etc.

— Open Google and type in “AngularJS Bootstrap” into the search bar.

Ok, now that we’ve covered the first rule, let’s move on to the next one.

Rule #2: Proper Code Style

A perfectly fitted Code Style will help you utilize all the perks this framework has to offer. As we have no desire to reinvent the wheel (well, if only for educational purposes), we’ve browsed through existing approaches. As a result, we decided to opt for this style. It allows structuring the code really well. It’s easy to read, which means a new developer can jump in effortlessly anytime. Great, we’re in the clear about the **Code Style** for our project.

+44 20 7183 5820

info@magora.co.uk
sales@magora.co.uk

Rule #3: Perfect structure

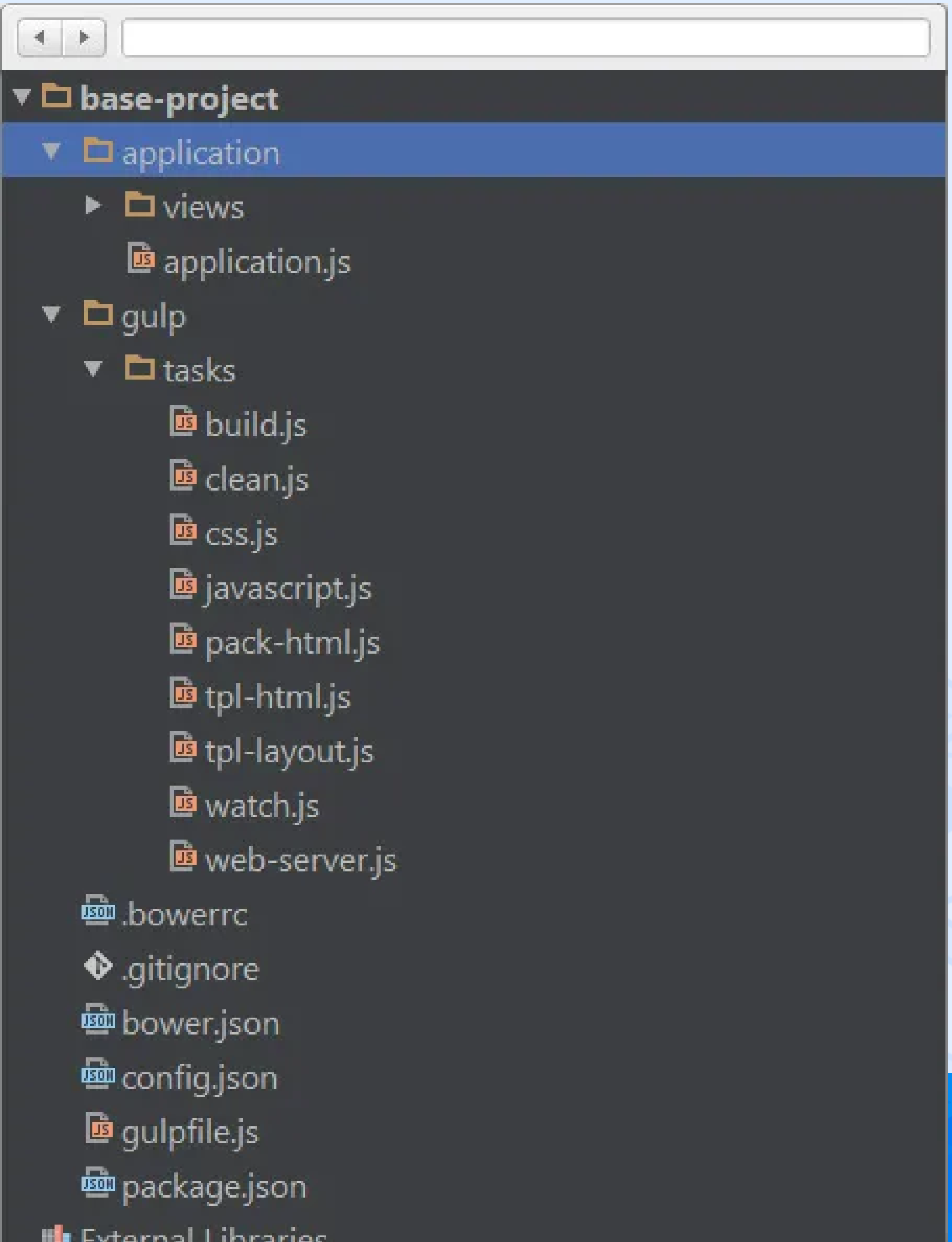
Now let's figure out a way to organize our code. I don't think I'll surprise anyone by saying: "Writing boilerplate code is flat out boring". Reuse the code! Go for the component approach! It's the most commonly used approach these days. Just look at Google's Polymer, or visit this website - you'll find some interesting stuff. So, what am I trying to say here?

- **40%** of what we're going to write must be reusable.
- For **40%** of what we need there's already an existing solution, we just need to study the tech part.
- **10%** is the new business logics that will tie our components together.
- **10%** is working on the layout of our components and project as a whole.

The curtain falls and the structure of our app is unveiled:

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

magora


base-project
application
views
application.js
gulp
tasks
build.js
clean.js
css.js
javascript.js
pack-html.js
tpl-html.js
tpl-layout.js
watch.js
web-server.js
.bowerrc
.gitignore
bower.json
config.json
gulpfile.js
package.json

As you might have noticed, we have a lot of small files. Including them into our **index.html** is not a good idea, and here's where **gulp** and it's team of plugins come in handy:

- gulp-minify-css
- gulp-angular-templatecache
- gulp-concat
- gulp-minify-html
- gulp-styl
- gulp-webserver
- gulp-jshint, etc. (the full list you can find in our repository)

What are we going to do with "gulp" and its entourage?

1. Set up our own **web server** (so that we don't have to deal with XAMP, apache or nginx, or use Denwer).
2. Pull all our JS files into one and obfuscate it for the production version, in order not to overwhelm user's browser with excessive kilobytes of data.
3. Use **jade** as an **html pattern generator**. Naturally, it must be processed and prepared for working with AngularJS.
4. Use **style** to describe styles. It must also be carefully assembled and packed. Copy fonts from libraries.

Thus, after all these manipulations we'll get three core files:

- index.html
- application.css
- application.js

+44 20 7183 5820
info@magora.co.uk
sales@magora.co.uk

We all know that browsers cache CSS and JS really well, so giving a new file version to a client isn't easy. That's why we'll ask **gulp** to add release version numbers to files, which it will recognize as **packages.json**.

According to these 3 rules we assemble our projects. An example of such a project you can find in our public GitHub repository. Right now it differs a bit from the description in this article, but it'll be renewed and updated soon. In the next article I'll tell you about how we're planning to get rid of boilerplate code for processing forms and CRUD pages.

+44 20 7183 5820

info@magora.co.uk
sales@magora.co.uk