

major workflow components

gulp-rev
attaches revision hashes to resource filenames to force client cache refresh of updates

gulp-csso
minifies CSS

gulp-minify-html
minifies HTML

gulp-uglify
minifies javascript

gulp-jshint
flags JS errors

gulp-less
compile LESS to CSS

gulp-ng-annotate
prepares angular dependency injection syntax for minification

gulp-connect
serves client builds and proxies server requests

gulp-imagemin
optimizes images

wiredep
keeps links to bower packages in index.html automatically up-to-date

gulp (gulpfile.js and files in /gulp)
A build system for automating production and packaging processes - a core component of civic-client development

angular-ui-grid
data-grid component used in many civic-client views

angular-bootstrap
Interface layout and component library that provides easily customizable responsive UI layouts and various components (buttons, lists, etc). Uses LESS.css.

LESS
A superset of the CSS language, describes layout and styling the civic-client UI.

angular-ui-router
state and route management package for angular; the backbone of the civic-client app, defines the states of the application and how those states are activated

angularjs
client-side single-page-application framework, drives runtime execution of civic-client

bower (bower.json and /bower_components)
Manages runtime packages & their dependencies

Development

Rutime

NPM (package.json)
npm (Node Package Manager) manages node packages and their dependencies which drive the entire civic-client development platform

node.js
Server-side Javascript runtime environment within which all civic-client development and production tasks are executed.

Javascript
The scripting language that powers both the production and runtime execution of civic-client

language runtime environment NPM package gulp plugin (confusingly, gulp plugins are managed by NPM, so are also NPM packages) bower package

file structure

/.tmp

used by gulp processes to store temporary files

/bower_components

bower stores all its packages here - these packages will be linked and loaded at runtime by the civic-client application

/dist

production builds are placed here by the `gulp dist` task

/gulp

gulp tasks, split into their own files for ease of management, are placed here. These are imported by the main `gulpfile.js` file.

/node_modules

NPM places all of its packages here. These packages all support gulp and bower tasks during development; they will never be loaded or used at runtime.

/notes

various notes, diagrams, regarding civic-client development, production, workflow, etc.

/src

contains the actual source code for the civic-client application

/tests

will contain unit and e2e tests, currently not used :(

dev files

`.bowerrc` - config file for the bower package manager

`.editorconfig` - used by some editors, IDEs to ensure similar settings for tabs, etc.

`.gitignore` - files to be ignored by git

`.jshintrc` - configuration file for jshint (used by `gulp-jshint`)

`bower.json` - bower package info and manifest

`gulpfile.js` - main gulp entry point, most gulp tasks are defined in `/gulp/*.js` files

`package.json` - NPM package info and manifest

`README.md` - human-readable description of project, displayed on github page

main gulp tasks

`gulp build`

package, concatenate, minify, and revision all code, templates, and styles for production-ready deployment in the `/dist` folder

`gulp serve`

serve raw code, templates, and styles on `localhost:3001` with livereload server automatically reloading any updated files.

`gulp serve:dist`

run the dist task then serve the `/dist` folder package on `localhost:3001` (for testing the distribution package, which will often have issues related to minification/concateration)

`gulp html`

run this task after adding new directives, services, components, etc. It will run all of the basic tasks that prepare and/or compile various application assets.

/pages

contains code, templates and styles for ‘static’ pages like About Us, Contact, etc.

/styles

contains `.less` layout and style files that pertain to global application components like headers, menus, etc. More individualized styles are packaged with their view or directive files.

/views

contains directories in a structure roughly corresponding to the civic-client states, one directory per state, and sub-states contained in sub-directories. Directives unique to each view are stored in a `/directives` folder, and directives shared by more than one sub-view can be stored in a `/common` folder.

/assets

contains application resources such as images, fonts, or data (no code, template or styles)

/components

contains services, directives, filters, and other application components that are utilized across the application, in multiple views. **appStates.js** is located here.

404.html - 404 template (currently unused)

favicon.ico - site favicon

index.html - main entrypoint of app, all libraries, styles and code are loaded from here. bower dependencies are automatically managed by `gulp-wiredep`, so it’s important to avoid editing these resources or the HTML comments that demark various automatically managed regions