<!-- TOC depthFrom:1 depthTo:6 withLinks:1 updateOnSave:1 orderedList:0 -->

- [Before writing code](#before-writing-code)

- [Install Dependencies](#install-dependencies)

- [Writing code](#writing-code)

- [Grunt tasks](#grunt-tasks)

- [Serving - `grunt serve`](#serving-grunt-serve)

- [Serving - `grunt servefast`](#serving-grunt-servefast)

- [Testing - `grunt test`](#testing-grunt-test)

- [Building dist - `grunt dist`](#building-dist-grunt-dist)

- [Submitting Pull Requests](#submitting-pull-requests)

- [QA](#qa)

- [Running `gh-pages` locally](#running-gh-pages-locally)

- [FuelUX Dev](#fuelux-dev)

- [Travis CI](#travis-ci)

- [Installing Travis CI Client locally](#installing-travis-ci-client-locally)

- [Restarting a Travis CI Build](#restarting-a-travis-ci-build)

- [From the Browser](#from-the-browser)

- [From the Command Line](#from-the-command-line)

<!-- /TOC -->

# Before writing code

1. [confirm issue is new](https://github.com/ExactTarget/fuelux/issues)

- if not, get involved in previous report of issue

1. [create a new issue](https://github.com/ExactTarget/fuelux/issues/new)

- this way, we can give feedback/direction as early as possible to ensure the most successful outcome for your hard work

# Install Dependencies

Fuel UX uses npm and [Grunt](http://gruntjs.com/) for development and maintenance tasks. You will need to install the required dependencies in order to contribute to Fuel UX.

In terminal from root directory of your fork of the fuelux repo:

1. `npm install`

2. `bower install`

# Writing code

\* do not edit or commit files in the `dist` directory

- project maintainers will commit files in the `dist` directory from time to time

- details on compiling CSS and JavasScript can be found [here](https://github.com/exacttarget/fuelux/blob/master/DETAILS.md#compiling-code)

\* source files are in respective `js`, `less`, and `fonts` directories

\* conform to [Salesforce Marketing Cloud style guide](https://github.com/ExactTarget/javascript)

\* if you are new to git, please review [our git commit conventions](https://github.com/ExactTarget/fuelux/wiki/Fuel-UX-Git-and-GitHub-Conventions)

1. [fork this repository](https://github.com/ExactTarget/fuelux/fork)

2. clone locally

3. `npm install`

4. create new branch, named after the GH Issue you are resolving

5. add unit tests for new or changed functionality

- unit tests are found in `/test`

5. make your changes

6. `grunt test` to lint and unit test your changes

7. `grunt servefast` and visit `http://localhost:8000/test/` for information on failing unit tests

1. [sync your fork](https://help.github.com/articles/syncing-a-fork/) with the upstream repo

1. commit your changes

- \*\*do not commit files within ````/dist````\*\*

- commit unit tests and code separately, unit tests first so we can see them fail and then pass

- write meaningful commit messages (NO WIP!)

## Grunt tasks

Run `grunt --help` or [check out the Gruntfile](https://github.com/ExactTarget/fuelux/blob/master/Gruntfile.js) to see all possible grunt tasks. When contributing, these are the grunt tasks you will be most likely to use:

### Serving - `grunt serve`

Starts a watch server for automated javascript validation and basic tests (JSHint, simplified QUnit) allowing for prototyping at [http://localhost:8000/](http://localhost:8000/) (not good for unit testing because server catastrophically fails if unit test fails).

### Serving - `grunt servefast`

Starts a watch server allowing for prototyping at [http://localhost:8000/](http://localhost:8000/) visual review of tests at [http://localhost:8000/test/](http://localhost:8000/test/).

### Testing - `grunt test`

Runs JSHint and full suite of QUnit tests.

### Building dist - `grunt dist`

This builds the dist directory (compiling your CSS and JS). If you are going to issue a pull request, you should not include changes to the dist directory that are generated from this grunt task.

# Submitting Pull Requests

All pull requests are validated via [Travis CI](https://travis-ci.org/). If the tests fail and you feel it is a Travis issue, you can [trigger a restart](#travis-ci).

While grunt can run the included unit tests via PhantomJS, this isn't a substitute for running tests across a variety of browsers and environments. Please be sure to test in as many of the browsers listed in `sauce\_browsers.yml` as you can before contributing.

1. run `grunt test` to lint & test your code

1. if necessary, rebase and squash to as few commits as practical

1. push to your forked repo

1. submit a pull request to master

- for help, visit GitHub's [using pull requests](https://help.github.com/articles/using-pull-requests)

1. Follow your pull request answering questions and making adjustments as appropriate until it is merged

# QA

For more in-depth testing of Fuel UX, you can install Fuel UX on the [gh-pages](#running-gh-pages-locally) site, fuelux-dev.herokuapp.com, and the fuelux.herokuapp.com site.

[Travis CI](#travis-ci) is run automatically for all Pull requests.

## Running `gh-pages` locally

In addition to basic testing, using `grunt test`, you can compile the dist and manually place it into the docs site (in `gh-pages` branch) and test fuel ux out there.

Once you have checkout out the `gh-pages` branch and run `npm install` and `bower install`:

1. If necessary, [install Jekyll](http://jekyllrb.com/docs/installation) (requires v2.3.x).

- \*\*Windows users:\*\* Read [this unofficial guide](http://jekyll-windows.juthilo.com/) to get Jekyll up and running without problems.

2. Install the Ruby-based syntax highlighter, [Rouge](https://github.com/jneen/rouge), with `gem install rouge`.

3. From the root `/fuelux` directory, run `jekyll serve` in the command line.

4. Open <http://localhost:9001> in your browser, and voilà.

## FuelUX Dev

If you have permissions to the ExactTarget org on Heroku, you can get information on cloning the `fuelux-dev` remote from its [app page](https://dashboard.heroku.com/orgs/exacttarget/apps/fuelux-dev/deploy/heroku-git) on [Heroku](https://www.heroku.com). If you do not have permissions and believe you should, please contact one of the FuelUX project maintainers.

A build of master is available at `https://fuelux-dev.herokuapp.com/dist/js/fuelux.js` and `https://fuelux-dev.herokuapp.com/dist/css/fuelux.css`.

\_These files should be considered unstable as this is our dev server\_

Sometimes people refer to FuelUX Dev as an "Edge Server". You can have your own edge server if you would like! To create one, setup a github web hook on Heroku for your fork of this repository and put the app into development mode with `heroku config:set NPM\_CONFIG\_PRODUCTION=false`.

## Travis CI

Pull requests are validated via [Travis CI](https://travis-ci.org/).

Periodically pull requests may fail Travis CI build integration testing with a false negative. If you suspect this is the case you can restart the test via the command line (see below).

[Travis](https://travis-ci.org/) downloads the `node\_modules` folder from the "[Edge](https://fuelux-dev.herokuapp.com)" server (["fuelux-edge"](https://fuelux-edge.herokuapp.com)) hosted on [Heroku](https://www.heroku.com). If you add or update a dependency in `package.json`, you will need to also update `package.json` in `master` locally and push it to [Heroku](https://www.heroku.com) for the dependency errors to be resolved in [Travis](https://travis-ci.org/).

### Installing Travis CI Client locally

Travis requires ruby and the [appropriate ruby gem](https://github.com/travis-ci/travis.rb#installation).

### Restarting a Travis CI Build

#### From the Browser

1. On the Pull Request page on Github, click on the "details" link in the Travis CI build area

1. Click "Login with Github" at the top right of the page

1. Click the "Restart Build" button (circular button with an arrow going in a circle)

#### From the Command Line

1. Login to Travis

In the terminal, issue the following command (You'll need to use your Github credentials):

```

travis login --org

```

1. Acquire build number

In the terminal, issue the following command:

```

travis history

```

1. Restart Build using the build number you obtained from `travis history`

In the terminal, issue the following command:

```

travis restart 9999

```