
ECS Development Documentation

Release 0 TODO:use git id

Latex Author Name

Apr 05, 2018

Contents

1 How to build Bootstrap 4

1

How to build Bootstrap 4

Prerequisites

1. Clone bootstrap from gogs and checkout `ecs_bootstrap-v4.0.0-alpha.5` branch. Add Github as remote.

```
git clone ssh://git@gogs.omoikane.ep3.at:10022/ecs/ecs-bootstrap.git
cd ecs-bootstrap git checkout ecs_bootstrap git remote add github
https://github.com/twbs/bootstrap.git
```

2. Install [Node](#), [NPM](#) with legacy support and [Ruby](#) with headers.

```
sudo apt-get install nodejs nodejs-legacy npm ruby ruby-dev
```

3. Install [Grunt](#) and [Bundler](#).

```
sudo npm install -g grunt-cli sudo gem install bundler
```

4. Install dependencies inside the bootstrap directory.

```
npm install --no-optional bundle install
```

If `bundle install` fails, do a `bundle update`, then try again.

Building Bootstrap

Once everything is ready, you can run the following tasks inside the *ecs-bootstrap* directory:

- `grunt` Run grunt to run tests locally and compile the CSS and JavaScript into `/dist`. Uses Sass, Autoprefixer, and UglifyJS.
- `grunt dist` `grunt dist` creates the `/dist` directory with compiled files. Uses Sass, Autoprefixer, and UglifyJS.
- `grunt test` Runs scss-lint, ESLint and QUnit tests headlessly in PhantomJS (used for CI).
- `grunt docs` Builds and tests CSS, JavaScript, and other assets which are used when running the documentation locally via `jeekyll serve`.

- `grunt watch` This is a convenience method for watching just Sass files and automatically building them whenever you save.

`grunt dist` builds all necessary files and puts them in the *dist* directory.

Local documentation

Running the documentation locally requires the use of `Jekyll`.

1. From the root bootstrap directory, run `bundle exec jekyll serve` in the command line.
2. Open <http://localhost:9001> in your browser, and voilà.

Apply local patches to a new version of bootstrap

1. Create a new branch starting with the unmodified new bootstrap version.

```
git branch ecs_bootstrap-$new_version $new_version
git checkout ecs_bootstrap-$new_version
```

2. Pull the local changes from the old version and resolve conflicts if any.

```
git cherry-pick $old_version..ecs_bootstrap-$old_version
```

3. Push the branch.

```
git push -u origin ecs_bootstrap-$new_version
```

Done!