

MANAGING DEPENDENCIES

Once you've run your generators, you'll often want to run `npm` (<https://www.npmjs.com/>) (or `Yarn` (<https://yarnpkg.com/>)) and `Bower` (<http://bower.io/>) to install any additional dependencies your generators require.

As these tasks are very frequent, Yeoman already abstracts them away. We'll also cover how you can launch installation through other tools.

Note that Yeoman provided installation helpers will automatically schedule the installation to run once as part of the `install` queue. If you need to run anything after they've run, use the `end` queue.

npm

You just need to call `this.npmInstall()` to run an `npm` installation. Yeoman will ensure the `npm install` command is only run once even if it is called multiple times by multiple generators.

For example you want to install `lodash` as a dev dependency:

```
class extends Generator {
  installingLodash() {
    this.npmInstall(['lodash'], { 'save-dev': true });
  }
}
```

This is equivalent to call:

```
npm install lodash --save-dev
```

on the command line in your project.

Manage npm dependencies programmatically

You can `programmatically create or extend your package.json` file if you don't want to use a template but like to have fixed versions of your dependencies. Yeomans file system tools can help you to get this job done.

Example defining `eslint` as dev dependency and `react` as dependency:

```
class extends Generator {
  writing() {
    const pkgJson = {
      devDependencies: {
        eslint: '^3.15.0'
      },
      dependencies: {
        react: '^16.2.0'
      }
    };

    // Extend or create package.json file in destination path
    this.fs.extendJSON(this.destinationPath('package.json'), pkgJson);
  }

  install() {
    this.npmInstall();
  }
};
```

Yarn

You just need to call `this.yarnInstall()` to launch the installation. Yeoman will ensure the `yarn install` command is only run once even if it is called multiple time by multiple generators.

For example you want to install lodash as a dev dependency:

```
generators.Base.extend({
  installingLodash: function() {
    this.yarnInstall(['lodash'], { 'dev': true });
  }
});
```

This is equivalent to call:

```
yarn add lodash --dev
```

on the command line in your project.

Bower

You just need to call `this.bowerInstall()` to launch the installation. Yeoman will ensure the `bower install` command is only run once even if it is called multiple time by multiple generators.

Combined use

Calling `this.installDependencies()` runs npm and bower by default. You can decide which ones to use by passing booleans for each package manager.

Example for using Yarn with Bower:

```
generators.Base.extend({  
  install: function () {  
    this.installDependencies({  
      npm: false,  
      bower: true,  
      yarn: true  
    });  
  }  
});
```

Using other tools

Yeoman provides an abstraction to allow users to `spawn` any CLI commands. This abstraction will normalize to command so it can run seamlessly in Linux, Mac and Windows system.

For example, if you're a PHP aficionado and wished to run `composer`, you'd write it this way:

```
class extends Generator {  
  install() {  
    this.spawnCommand('composer', ['install']);  
  }  
}
```

Make sure to call the `spawnCommand` method inside the `install` queue. Your users don't want to wait for an installation command to complete.