

Setting up a vue-test sample project

In this lesson, we'll be introduced to Jest. We will also learn how to set up a Vue.js sample project.

WE'LL COVER THE FOLLOWING ^

- Introduction
- Getting Started

Introduction

Vue is a framework for building user interfaces. The core library is focused on the view layer and is very flexible to integrate with different libraries.

vue-test-utils, the official VueJS testing library based on **avoriaz** is just around the corner. It provides all necessary tools needed to easily write unit tests in a *VueJS* application.

Jest, on the other side, is the testing framework developed at Facebook. It makes testing a breeze, with awesome features such as:

- Almost no config. by default
- Very cool interactive mode
- Parallel testing
- Spies, stubs and mocks out of the box
- Built-in code coverage
- Snapshot testing
- Module mocking utilities



As we'll see soon, testing using **Jest** is much easier than working with any of the existing tools like mocha, karma, etc.

the existing tools like Mocha, Karma, etc.



Getting Started

Let's get started with a default **Vue** application:

```
'use strict'
require('./check-versions')()

process.env.NODE_ENV = 'production'

const ora = require('ora')
const rm = require('rimraf')
const path = require('path')
const chalk = require('chalk')
const webpack = require('webpack')
const config = require('../config')
const webpackConfig = require('./webpack.prod.conf')

const spinner = ora('building for production...')
spinner.start()

rm(path.join(config.build.assetsRoot, config.build.assetsSubDirectory), err => {
  if (err) throw err
  webpack(webpackConfig, (err, stats) => {
    spinner.stop()
    if (err) throw err
    process.stdout.write(stats.toString({
      colors: true,
      modules: false,
      children: false, // If you are using ts-loader, setting this to true will make TypeScript
      chunks: false,
      chunkModules: false
    }) + '\n\n')

    if (stats.hasErrors()) {
      console.log(chalk.red('  Build failed with errors.\n'))
      process.exit(1)
    }

    console.log(chalk.cyan('  Build complete.\n'))
    console.log(chalk.yellow(
      '  Tip: built files are meant to be served over an HTTP server.\n' +
      '  Opening index.html over file:// won\'t work.\n'
    ))
  })
})
```

Note: All the required dependencies are already installed in this application. If you want to set up the project environment locally then follow [these](#) steps.

Now that we have set up our vue test sample project, let's move towards testing components.