

npm –save or –save-dev. Which one to use?

Introduction

If you have ever worked in NodeJs, you must have install one or two packages through “*npm install <package>*” command. By running this command, the nodeJs will install this package on your working directory, under node_modules.

To save this packages as your dependencies, under package.json, you have two choices:

- –save-dev
- –save

What is package.json?

All npm packages contain a file, usually in the project root, called package.json – this file

Let us understand the difference that it can make.

Detail

Say you have a package.json within your root folder of your project.

If you don't have one then create a package file using **npm init** command.

My package.json looks like this:

```
{  
  "name": "TMSPA",  
  "version": "1.0.0",  
  "description": "Single page application for TM",  
  "main": "index.html",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "https://github.com/codebased/android-test.git"  
  },  
  "author": "Am",  
  "license": "ISC",  
  "bugs": {  
    "url": "https://github.com/codebased/android-test/issues"  
  }  
}
```

```
},  
"homepage": "https://github.com/codebased/android-test"  
}
```

Now I want to install some dependencies.

Before I install one, I need to search for the package name. If you know the package that you want to install that's good. Otherwise, you can use npm search command:

```
npm search bootstrap
```

or try one of the following search tools:

- <https://www.npmjs.org/>
- <http://eirikb.github.com/nipster/>
- <http://browsenpm.org/>
- <http://node-modules.com/>
- <https://nodejsmodules.org/>
- <https://github.com>

Once you have identified the right package that you want to install, you can use the mentioned command i.e. npm install <package name>.

Here you have two, actually three, options.

1. use **–save-dev**

e.g. npm install should --save-dev

You will use this option when you want to download a package for developers , such as grunt, gulp, then use this option. Thus, when you are distributing your code to production, these dependencies will not be available.

As an example, let say, you want to use grunt as your task runner. This package is required for development purpose. Thus, you should use –save-dev here.

```
npm install grunt --save-dev
```

The above command will save grunt dependency under **devDependencies** section of your package.json, shown below:

```
{  
  "name": "TMSPA",  
  "version": "1.0.0",  
  "description": "Single page application for TM",  
  "main": "index.html",  
  "scripts": {  
    ..  
    "author": "Codebased",  
    ..  
    ..,  
    "devDependencies": {  
      "gulp": "^3.8.11"  
    }  
  }  
}
```

2. Use **-save** flag

You will use this option when you want to save a package dependency for distribution. Item such as angularjs, or any other module that is required at run time by your program, you will use **-save** switch.

```
npm install angularjs --save
```

Now my package.json looks like this:

```
{  
  "name": "TMSPA",  
  "version": "1.0.0",  
  "description": "Single page application for TM",  
  ..,  
  "dependencies": {  
    "angularjs": "^1.4."  
  },  
  "devDependencies": {  
    "gulp": "^3.8.11"  
  }  
}
```

3. Use **nothing**

If you call `npm install` command without any flag then it will install package. However, there is no way the `package.json` will be updated with your dependencies.

This option is not recommended because there is no way others will get to know about the dependencies that your module has.

Conclusion

In conclusion, we understand that the `-save-dev`, and `-save` flags are used for limiting the scope of your dependencies.