**Introduction to Bower and Grunt**

**Prerequisites:** To understand Bower and Grunt we should have some knowledge about

1. Node.js
2. Node package or module
3. NPM (Node Package Manager)

**What is Node.js: Node**.**js** is a runtime environment for server-side and networking applications written in JavaScript (Something like .net environment for applications written in c#).

**Installing Node.js:** For windows just go to the official website of node.js (<https://nodejs.org/en/>). Download the installer .msi and run it. At the command prompt if you run **“node -v”** command, the result will show you the installed node.js version.

**What is node package or node module:** Node modules or node packages are one kind of reusable code group written in JavaScript (something like exe or dll of .net with the difference is that node package contains files) and which can be published to npm. From the npm other people can download, install and use the packages according to their need.

A node package is a directory with one or more files (JavaScript) on it. This directory also has “package.json” file that contains metadata of the package.

Bower and Grunt are kinds of npm packages.

**What is NPM:** NPM stands for Node Package Manager. When we are talking about npm, it may mean following three things

1. NPM web sites, that contains various documentation regarding npm and node modules or packages
2. NPM repository or database that contains node.js modules or nod.js packages various peoples are sharing
3. NPM clients, which are used to download, install and run node.js modules.

Visit following url for more information: https://docs.npmjs.com/getting-started/what-is-npm

**Installing NPM :** NPM will be automatically installed with node.js. You can see the version by following command **“npm -v”**. NPM gets updated more frequently than node.js. That’s why you may need to update npm version. You can update the npm version by following command **“npm install npm -g”**

**Installing npm packages:** There are two ways to install npm packages

1. **Globally**: If you are going to use the package as a command line tool so that you can run the commands related to the package from command line then you need to install it globally. Run following command from command line **“npm install –g <packagename>”** to install any packages globally.

See the example of jshint

Please keep in mind that not all the packages can be used as command line tool.

1. **Locally**: If you want the packages will be used by your own module or application then install it locally. To install locally you have to go to your application directory at the command line and run following command **“npm install <packagename>”.**

Another way to install nmp packages locally by using **“package.json”** file. Add a “package.json” file to your project directory; add all the required modules or packages as dependency and run **“npm install” command from command line.** All themodules or packages you have defined as dependency will be installed locally.

Following is the link regarding package.json file: <https://docs.npmjs.com/files/package.json>

See the example

**Bower:**

Bower is a package manager just like **NPM** with following differences

|  |  |
| --- | --- |
| **NPM** | **Bower** |
| It was created to manage packages created and used in node.js environment, like grunt, jshint etc | It was created to manage front end libraries like jQuery, AngularJs, Bootstrap etc. |
| Dependent modules installed in the sub folder | Dependent modules installed in the root folder |
| Duplicate dependency may be installed as with npm each of the packages handles their own dependency. It cannot check whether the dependency already downloaded as the dependent modules are in the subfolder of another package. | Bower installed all dependencies once. As dependent modules are in root folder, it can check whether the dependency already downloaded or not. |

This is how official website describes bower:

*“Web sites are made of lots of things — frameworks, libraries, assets, utilities, and rainbows. Bower manages all these things for you.*

*Bower works by fetching and installing packages from all over, taking care of hunting, finding, downloading, and saving the stuff you’re looking for”*

**Prerequisites of Bower:**

Bower is a node.sj module so to get and install it you need node.js and npm installed. Bower packages are Git repos, so you also need Git installed.

**Installing Bower**

As bower is an npm package or module we can install it like other npm module. Technically we can install bower locally also, but as we want to use it as command line tool and from other directories also we should install bower globally. Following is the command for installation: **npm install –g bower**

**Installing Bower packages**

Just like npm local installation we can install bower packages. The command is **“bower install <packagename>”.** If we run the command, the package will be downloaded at the folder named **“bower\_components”.**

See example

Instead of name we can also use url or git end point of the packages to install by the command **“bower install <url>”.** For example

**bower install git://github.com/user/package.git**

**bower install http://example.com/script.js**

Another way to install bower packages locally by using **“bower.json”** file. Add a “bower.json” file to your project directory; add all the required modules or packages as dependency and run **“bower install”** command from command line**.** All themodules or packages you have defined as dependency will be downloaded at the **“bower\_components”** folder. To save the downloaded package as dependency run **“bower install <packagename> --save”** command**.** To save the downloaded package as dev dependency run **“bower install <packagename> --save-dev”** command**.**

To see the list of packages installed or downloaded run **“bower list”.** Toget the path of the package list run **“bower list --path”** command.

If you want to install specific version of a package, the command is **“bower install <packagename>#<version>”.**

**Prefixes before version in bower.json file**

Following way we can define dependency versions in bower.json file:

1. version Must match version exactly
2. >version Must be greater than version
3. >=version etc
4. <version
5. <=version
6. ~version "Approximately equivalent to version", must match major and minor(1.2.x)
7. ^version "Compatible with version" must match major (1.x.x)
8. 1.2.x 1.2.0, 1.2.1, etc., but not 1.3.0
9. http://... See 'URLs as Dependencies' below
10. \* Matches any version
11. "" (just an empty string) Same as \*
12. version1 - version2 Same as >=version1 <=version2.
13. range1 || range2  Passes if either range1 or range2 are satisfied.
14. git url
15. user/repo
16. tag A specific version tagged and published as tag
17. path/path/path

For example, these are all valid:

{ "dependencies" :

{ "foo" : "1.0.0 - 2.9999.9999"

, "bar" : ">=1.0.2 <2.1.2"

, "baz" : ">1.0.2 <=2.3.4"

, "boo" : "2.0.1"

, "qux" : "<1.0.0 || >=2.3.1 <2.4.5 || >=2.5.2 <3.0.0"

, "asd" : "http://asdf.com/asdf.tar.gz"

, "til" : "~1.2"

, "elf" : "~1.2.3"

, "two" : "2.x"

, "thr" : "3.3.x"

, "lat" : "latest"

, "dyl" : "file:../dyl"

}

}

**bower list and bower list –path command**

**Description of bower.json file**

Please visit following link to see descriptions of bower.json file: <http://bower.io/docs/creating-packages/>

**What is Grunt**

Grunt is a JavaScript task runner tool. We can write some tasks using JavaScript in node.js environment and using grunt we can run those task.

The idea behind grunt is, when we work in a JavaScript project we need to do some kind of regular task, like:

1. Downloading JavaScript or css packages
2. Checking error or warning at the file
3. Copying files to different folder
4. Minified .js or .css files
5. Making a single .js or .css file form multiple files
6. Making compiled css file from SASS or LESS files.
7. And many more…

For this type of tasks there are grunt plugins (also called grunt task). We use grunt to run these plugins to accomplish these types of tasks.

**Installing Grunt**

Grunt and also grunt plugins are npm modules. So we can install and manage them using npm. We need to install **grunt-cli** to run grunt commands from command line tool. **grunt-cli** is command line tool.

The convention is we should install grunt-cli globally so that we can run grunt related command from anywhere. The command is:

**“npm install –g grunt-cli”**

And **grunt** itself should be installed locally so that our application can use it. The command is:

**“npm install grunt”.**

**Installing Grunt plugins or tasks**

As we already discussed that grunt plugins are npm modules we can install the like other npm modules, by following command: “**npm install <pluginname>”**

As these plugins are project specific, we should install them locally.

We need to configure grunt tasks to run them.

**Configuring Grunt Tasks**

Grunt tasks are configured through **“Gruntfile.js”.** When grunt runs, it reads the configuration from the file and run the tasks accordingly. We need to keep the **“Gruntfile.js”** file at the root

directory of the project along with **“package.json”.** A “Gruntfile” is comprised of the following parts:

1. The "wrapper" function
2. Project and task configuration
3. Loading Grunt plugins and tasks
4. Custom tasks

Following link describes the file in detail: <http://gruntjs.com/sample-gruntfile>

**Node.js cont….:**

http://stackoverflow.com/questions/5390886/nodejs-require-paths-resolve-problem

A module prefixed with '/' is an absolute path to the file. For example, require('/home/marco/foo.js') will load the file at /home/marco/foo.js.

A module prefixed with './' is relative to the file calling require(). That is, circle.js must be in the same directory as foo.js for require('./circle') to find it.

Without a leading '/' or './' to indicate a file, the module is either a "core module" or is loaded from a node\_modules folder.

If the given path does not exist, require() will throw an Error with its code property set to 'MODULE\_NOT\_FOUND'.