



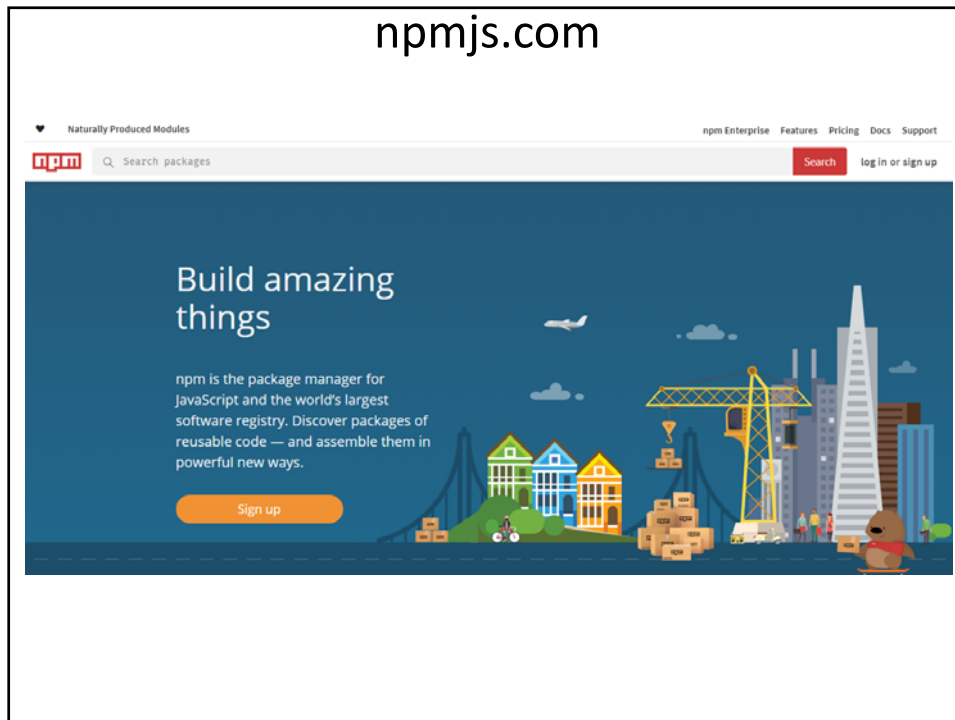
MODULE 7

## MANAGING MODULES WITH **NPM**

### What Does NPM Do?

- > Node.js also has a package manager just like RubyGems from Ruby or Maven from Java
  - Node Package Manager (NPM)
- > It has become so popular throughout the community that beginning with the 0.6.X Node.js version it was integrated within the main Node.js installer
- > It is the standard manager of this platform.

# npmjs.com



## NPM

- > NPM is very easy to use.
- > You can use it
  - to manage your project's dependencies
  - to create commands for task automation purposes,
- > Everything is created and managed into the **package.json** file

## NPM Commands

- > npm init
  - Displays a simple wizard to help you create and describe your new project.
- > npm install module\_name
  - Installs a module.
- > npm install -g module\_name
  - Install a global module.
- > npm install module\_name --save
  - Installs a module and adds it into the package.json file, inside dependencies

## NPM Commands

- > npm install module\_name --save-dev
  - Installs a module and adds it into the package.json file, inside devDependencies.
- > npm list
  - Lists all the modules installed on the project.
- > npm list -g
  - Lists all the global modules installed on the OS.
- > npm remove module\_name
  - Uninstalls a module from the project.
- > npm remove -g module\_name
  - Uninstalls a global module.

## NPM Commands

- > `npm remove module_name --save`
  - Uninstalls a module from the project and also removes it from the attribute dependencies.
- > `npm remove module_name --save-dev`
  - Uninstalls a module from the project and also removes it from the attribute devDependencies.
- > `npm update module_name`
  - Updates the module version.
- > `npm update -g module_name`
  - Updates the global module version.

## NPM Commands

- > `npm -v`
  - Displays the NPM current version.
- > `npm adduser username`
  - Creates an account to <https://npmjs.org>.
- > `npm whoami`
  - Displays details of your public NPM profile
  - You must create an account using the previous command
- > `npm publish`
  - Publishes a module to [npmjs.org](https://npmjs.org)
  - It's necessary to have an active account first

## Understanding the package.json File

- > All Node.js projects are called *modules*
- > What is a module?
- > They all mean the same thing
  - Module, Library, Framework
- > The term *module* was taken from JavaScript's concept
  - A language that works with a modular architecture.

## Understanding the package.json File

- > When we create a Node.js project (i.e., a module), this project is followed by a descriptor file of modules, called package.json.
- > This file is essential to a Node.js project.
- > A badly written package.json can cause bugs or even prevent your project from executing
- > This file has some key attributes that are read by both Node.js and the NPM.

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*Major Version*

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*Major Version*

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*Major Version*

*Minor Version*

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*Major Version*

*Minor Version*



## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

The diagram illustrates the semantic versioning components of the version string "1.2.3". Three blue boxes highlight the digits "1", "2", and "3". Blue arrows point from these boxes to labels: "Major Version" for "1", "Minor Version" for "2", and "Patch Version" for "3".

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

The diagram highlights the version string "1.0.0" for "module-1" in the dependencies section with a blue box.

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0", Fixed Version
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*Update module  
as a patch level 1.0.X*

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

*updates versions that are greater than or equal to 1.0.0 in all the version levels.*

## NPM Task Automation

- > You can also automate tasks using the **npm** command.
- > In practice, you can only create new executable commands running `npm run command_name`.
- > To declare those new commands, just create them inside the attribute `scripts` on `package.json`

## Example

```
{
  "name": "my-first-node-app",
  "description": "My first node app",
  "author": "User <user@email.com>",
  "version": "1.2.3",
  "private": true,
  "scripts": {
    "start": "node app.js",
    "clean": "rm -rf node_modules",
    "test": "node test.js"
  },
  "dependencies": {
    "module-1": "1.0.0",
    "module-2": "~1.0.0",
    "module-3": ">=1.0.0"
  },
  "devDependencies": {
    "module-4": "*"
  }
}
```

npm run start  
npm run clean  
npm run test