Introduction

While working on the different loopback applications as Integrator and various Microservices modules, one module might be dependent on the other one. Mostly, this use case applies at the time of Integrator development. **npm link** would help us very effectively in this usecase.

Follow the below steps to use your NPM as node_modules in your application, while you are still working directly in your respective Repo.

Example

- **testNPM** - Node Module which we are developing and to be published as NPM module once tested.
- **testApp** - Application which consumes testNPM as it's dependency.

Commands to link Node Module

- Go to **testNPM** folder in command prompt/terminal
  cd <testNPM directory path>
- Create the symbolic link from the global folder to testNPM folder, with the below command
  npm link
- Now, Go to the App directory where you want to use **testNPM** as dependency
  cd <testApp directory path>
- Link **testNPM** as a Node Module dependency to the **testApp**
  npm link testNPM

We are done. Now, whenever **testApp** is looking for **testNPM** modules, it will directly load from the source directory of **testNPM**.

You can test all **testNPM** functionalities seamlessly, if there are any errors, you can directly modify in **testNPM** source directory, instead of Node Modules folder and copying back the fix to original directory.

Un-linking the local development NPM package

If you wish to unlink the local module after publishing it, run the below command:

- **npm r testNPM -g**