Node Package Manager

Introduction

NPM is

- ➢ a package manager for JavaScript
- ➢the default package manager for Node.js

NPM consists of a **command line client** (npm) & an **online database** of public and paid-for private packages (npm registry)

Introduction

- A plethora of Node.js libraries and applications are published on npm, and many more are added every day. Search here: (https://www.npmjs.com)

- Npm command line interface **comes installed with node.js**

- NPM allows users to consume and distribute JavaScript modules that are available on the registry.

Introduction

- Once you have a package you want to install, it can be installed with a single command-line command.
- We can download all npm public software packages without any registration or logon.
- All popular libraries and frameworks are on NPM! For example react and express.

- NPM : stands for Node Package Manager
- Npm is the world's largest software registry
- Npm command line interface comes installed with node.js
- is an online repository for the publishing of open-source Node.js projects

- second, it is a command-line utility for interacting with said repository that aids in package installation, version management, and dependency management.

npm packages that can be installed are:

- Regular/ simple dependencies
  - Packages that contain some code that we can include in our code, like express the node framework that we will use -> our code relies on them
- development dependencies: tools for development-> our code does not rely on them. like a debugger tool!

We may install dependencies in one folder, for a single project only or globally!

Lets check https://www.npmjs.com/package/nodemon

Package.json file

In visual studio lets execute:

- npm init (& fill info required)

- This command will create a Package.json file which is a configuration file
  - is a JSON file that lives in the root directory of your project
  - it holds important information about the project.
  - it contains human-readable metadata about the project (like the project name and description) as well as functional metadata like the package version number and a list of dependencies required by the application.

Package.json file

- node_modules folder=> like a cache for the external modules that your project depends upon.

- When you npm install them, they are downloaded from the web and copied into the node_modules folder and Node.js is trained to look for them there when you import them (without a specific path).

- When sharing a project we do not have to share node-modules folder. If we just execute **npm install** in the project directory it will install everything! Because it reads the package.json file!

To be continued…