# package.json

npm (Node Package Manager)이란?

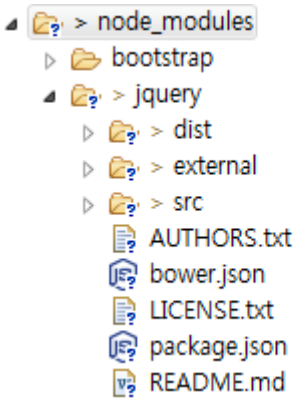
node.js에서 사용되는 모듈을 패키지로 만들어 npm 통하여 관리하고 배포.
글로벌 설치랑 로컬설치로 구분
글로벌 설치 ->
시스템 디렉토리에 설치하는것
예시) `npm install -g express`
npm 설치 방법

1. https://nodejs.org/en/download/ 버전에 맞게 설치.
2. cmd 창을 켜서 package.json 설정할 폴더로 이동.
3. npm init 명령어 실행 - package.json 파일 생성됨

```
C:\Users\user\Documents\toyProject>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (toyProject) toyproject
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository: (https://github.com/leehan0617/toyProject.git)
keywords:
author:
license: (ISC)
About to write to C:\Users\user\Documents\toyProject\package.json:

{
  "name": "toyproject",
  "version": "1.0.0",
  "description": "Java Web Project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/leehan0617/toyProject.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/leehan0617/toyProject/issues"
  },
```

```
  "homepage": "https://github.com/leehan0617/toyProject#readme"
}


Is this ok? (yes)
```

4. npm install jquery -save
   jquery 같은 패키지명 입력하여 dependency 설치 및 저장. 아래와 같이 폴더가 생기면서 설치됨.




**기본 명령어**

**npm init ->package.json 생성**

**npm install [패키티명] -save -> 패키지 설치**

**npm uninstall [패키지명] -save ->패키지 삭제**

**npm update [패키지명] -save ->패키지 수정**

기본package.json 파일 설정

```
{
  "name": "toyproject",
  "version": "1.0.0",
  "description": "Java Web Project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "start http://localhost:80/toyProject"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/leehan0617/toyProject.git"
  },
  "author": "",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/leehan0617/toyProject/issues"
  },
  "homepage": "https://github.com/leehan0617/toyProject#readme",
  "dependencies": {
```

```
    "bootstrap": "^3.3.7",
    "jquery": "^3.2.1"
  }
}
```

- 기본 설정 설명

| Key | Value |
|---|---|
| name | 프로젝트 이름이다. 중앙 저장소에 배포할 때 version과 함께 필수 항목이다. URL로 사용되고, 설치할 때 디렉터리 이름이 되기 때문에 URL이나 디렉터리에서 쓸 수 없는 이름을 사용하면 안 된다. |
| version | 프로젝트 버전이다. 3단계 버전을 사용하며, 붙임표(-)로 태그 이름을 적을 수 있다. |
| description | 프로젝트 설명이다. |
| keywords | 프로젝트를 검색할 때 참조되는 키워드이다. |
| homepage | 프로젝트 홈페이지 주소이다. |
| author | 프로젝트 작성자 정보이다. JSON 형식으로 name, email, url 옵션을 포함한다. |
| contributors | 프로젝트에 참여한 공헌자 정보이다. |
| repository | 프로젝트의 소스 코드를 저장한 저장소의 정보이다. |
| scripts | 프로젝트에서 자주 실행해야 하는 명령어를 scripts로 작성해두면 npm 명령어로 실행 가능하다. |
| config | 소스 코드에서 config 필드에 있는 값을 환경 변수처럼 사용할 수 있다. |
| private | 이 값을 true로 작성하면 중앙 저장소로 저장하지 않는다. |
| dependencies | 프로젝트 의존성 관리를 위한 부분이다. 이 프로젝트가 어떤 확장 모듈을 요구하는지 정리할 수 있다. 애플리케이션을 설치할 때 이 내용을 참조하여 필요한 확장 모듈을 자동으로 설치하기 때문에 여러분이 개발한 애플리케이션이 특정한 확장 모듈을 사용한다면 여기에 꼭 명시를 해주어야 한다. 또한, npm install 명령은 여기에 포함된 모든 확장 모듈들을 설치하게 되어 있다. |
| devDependencies | 개발할 때만 의존하는 확장 모듈을 관리한다. |
| engine | 실행 가능한 노드 버전의 범위를 결정한다. |

개발 테스트 dependencies

**npm install [패키티명] --save-dev -> 패키지 설치**

**npm uninstall [패키지명] --save-dev ->패키지 삭제**

**npm update [패키지명] --save-dev ->패키지 수정**

버전!!!

틸드(~)방식이다. 틸드는 간단히 말하면 현재 지정한 버전의 마지막 자리 내의 범위에서만 자동으로 업데이트한다.

- ~0.0.1 : >=0.0.1 <0.1.0

- ~0.1.1 : >=0.1.1 <0.2.0
- ~0.1 : >=0.1.0 <0.2.0
- ~0 : >=0.0 <1.0

캐럿(^) 은 호환되는 범위에서 업데이트

-참고자료
http://programmingsummaries.tistory.com/385
http://webclub.tistory.com/472
http://alexband.tistory.com/10
https://blog.outsider.ne.kr/829

eclipse
- Node 설치
  해당 URL 참고
http://linor.tistory.com/5