

# npm에 node.js 모듈 배포하기

먼저 npm에 계정을 만들어야 한다. [npm 사이트](#)에서 할 수도 있지만, 터미널에서로도 계정을 만들 수 있다. 이미 계정이 있는 경우 `npm login` 으로 로그인하자.

`npm adduser`

## package.json

npm에 배포하기 위해선 배포할 모듈의 루트 디렉토리에 'package.json'가 있어야 한다. 아직 'package.json'이 없다면 `npm init` 혹은 `yarn init` 으로 생성할 수 있다. 'package.json'은 아래와 같은 형식으로 되어있다.

```
{
  "name": "Module",
  "version": "1.0.0",
  "description": "Some module",
  "main": "lib/index.js",
  "scripts": {
    "dev": "babel-node src/index.js",
    "prepublish": "babel src --out-dir lib"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/user/repo.git"
  },
  "keywords": [
    "keyword1"
  ],
  "author": "user",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/user/repo/issues"
  },
  "homepage": "https://github.com/user/repo#readme",
  "devDependencies": {
    "babel-cli": "^6.24.1",
    "babel-core": "^6.24.1",
    "babel-preset-env": "^1.4.0"
  },
  "dependencies": {
    "lodash": "^4.17.4"
  },
}
```

- "name"
- "version": 매 배포마다 이 부분을 수정해서 버전을 높여야 배포할 수 있다. 이미 배포된 것과 버전이 같다면 코드가 달라도 배포되지 않는다.
- "description"
- "main": 패키지가 import되면 실행될 파일.
- "scripts": 특정 스크립트는 npm 배포시 혹은 `prepublish` 는 `npm publish` 를 하면 배포되기 전에 실행된다. 이런 스크립트 목록은 [여기](#)에서 확인할 수 있다.
- "repository": 코드의 저장소 주소
- "keywords": npm에서 목록에 있는 단어로 검색하면 패키지를 검색결과에 노출한다.
- "author"
- "license"
- "devDependencies": 패키지를 설치할 때는 제외되는 개발용 의존 패키지 목록.
- "dependencies": 패키지를 설치할 때 함께 설치되는 의존 패키지 목록.

## .npmignore

'package.json'과 함께 배포시 또 필요한 파일은 '.npmignore'다. 여기엔 패키지를 배포할 때 npm에 업로드할 필요가 없는 파일들의 목록을 적어놓는데, 테스트코드 파일이 대표적인 예다. 만약 '.npmignore'가 없다면 '.gitignore'에 적힌 파일을 제외시킨다.

아래 파일들은 굳이 '.npmignore'에 명시하지 않아도 자동으로 제외시킨다.

```
node_modules
*.swp
.*
.DS_Store
```

```
.git
.hg
.npmrc
.lock-wscript
.svn
.wafpickle-*
config.gypi
CVS
npm-debug.log
```

## Babel로 컴파일한 패키지 배포하기

소스코드는 javascript의 최신 스펙을 이용해서 작성했더라도, 범용성을 위해 Babel로 컴파일 후 배포해야 한다. 패키지의 구조는 아래와 같다고 하자.

```
├── .gitignore
├── .npmignore
├── package.json
├── src // 소스코드
│   └── index.js
└── lib // 컴파일된 파일
    └── index.js
```

먼저 'package.json'에서 `"main": "lib/index.js"` 로 패키지의 entry point를 컴파일된 파일로 설정한다. 그 다음 `"scripts"` 에 `"prepublish": "babel src --out-dir lib"` 를 추가해서 배포전에 Babel이 컴파일할 수 있도록 한다.

컴파일된 파일은 버전관리할 필요가 없으므로 '.gitignore'에 `lib` 를 추가한다.  
반대로 소스코드는 배포할 필요가 없으므로 '.npmignore'에 `src` 를 추가한다.

이제 `npm publish` 를 실행하면 컴파일된 파일만 배포한다.

May 21, 2017