



JavaScript

New Features

Build Tools and Environments

NPM



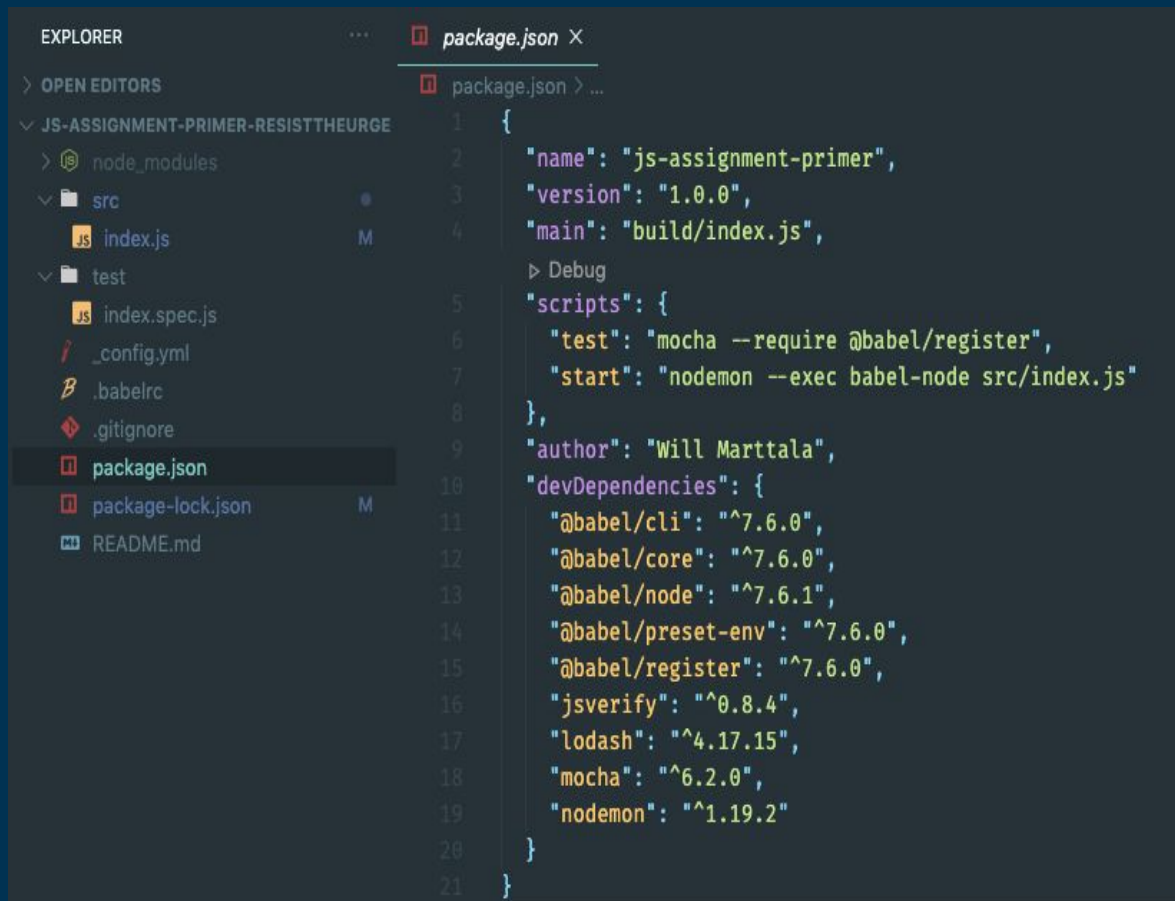
Node **P**ackage **M**anager



A dependency management system for
Node and JavaScript



Equivalent to Maven



The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor view on the right. The Explorer sidebar shows the project structure for 'JS-ASSIGNMENT-PRIMER-RESISTTHEURGE', including folders like 'node_modules', 'src', and 'test', and files like 'index.js', 'index.spec.js', '_config.yml', '.babelrc', '.gitignore', 'package.json', 'package-lock.json', and 'README.md'. The 'package.json' file is selected and its content is displayed in the Editor view. The content of 'package.json' is as follows:

```
1 {
2   "name": "js-assignment-primer",
3   "version": "1.0.0",
4   "main": "build/index.js",
5   "scripts": {
6     "test": "mocha --require @babel/register",
7     "start": "nodemon --exec babel-node src/index.js"
8   },
9   "author": "Will Marttala",
10  "devDependencies": {
11    "@babel/cli": "^7.6.0",
12    "@babel/core": "^7.6.0",
13    "@babel/node": "^7.6.1",
14    "@babel/preset-env": "^7.6.0",
15    "@babel/register": "^7.6.0",
16    "jsverify": "^0.8.4",
17    "lodash": "^4.17.15",
18    "mocha": "^6.2.0",
19    "nodemon": "^1.19.2"
20  }
21 }
```

NPM: package.json

- An npm project is defined by the package.json file in the root project folder
- It maintains the core metadata, dependencies, and configuration of the project

NPM: Starting a Project



npm init: initializes a brand-new project



npm install: installs any existing packages listed in the package.json

NPM: Installing Dependencies



NPM INSTALL <PACKAGE NAMES>: INSTALLS
PACKAGES IN THE NODE_MODULES FOLDER
AND ADDS THEM TO THE PACKAGE.JSON'S
MAIN DEPENDENCY LIST



NPM INSTALL --SAVE-DEV <PACKAGE NAMES>:
DOES THE SAME, BUT ADDS THE PACKAGES TO
A SPECIAL DEVDEPENDENCY LIST INSTEAD OF
THE MAIN ONE

When to --save-dev



IF THE PACKAGE IS A
DEVELOPMENT TOOL



IF THE PACKAGE IS NOT
IMPORTED BY YOUR MAIN
APPLICATION CODE

NPM: Scripts

```
"scripts": {  
  "test": "mocha --require @babel/register",  
  "start": "nodemon --exec babel-node src/index.js"  
},
```

- Named command line scripts can be defined in the package.json
- For special names like start, test, and a few others, they can be called with `npm <script name>` in the project root directory

Common Dev Tools

