**`package.json` and `package-lock.json`**

`package.json` and `package-lock.json` are both files used in Node.js projects, but they serve different purposes and have distinct roles in managing dependencies.

1. \*\*package.json\*\*:

The `package.json` file is a manifest that contains metadata about your project and lists the dependencies and devDependencies your project relies on. It also includes information about your project's name, version, author, scripts, and more. It's typically hand-edited by developers to specify the project's configuration, dependencies, and other important information. This file can be shared with others to ensure consistent development environments.

Example snippet of a `package.json` file:

```json

{

"name": "my-project",

"version": "1.0.0",

"dependencies": {

"express": "^4.17.1",

"axios": "^0.21.1"

},

"devDependencies": {

"eslint": "^7.32.0",

"mocha": "^9.1.3"

}

}

```

2. \*\*package-lock.json\*\*:

The `package-lock.json` file is automatically generated by npm (Node Package Manager) when you install or update dependencies in your project. It is a detailed record of the exact versions of each dependency and their sub-dependencies that were installed. This file is used to ensure that the same versions of packages are installed across different environments. It's particularly important for maintaining consistent builds and avoiding unexpected changes in dependencies, even if newer versions are released.

Example snippet of a `package-lock.json` file:

```json

{

"name": "my-project",

"version": "1.0.0",

"lockfileVersion": 2,

"requires": true,

"packages": {

// ...

},

"dependencies": {

"express": {

"version": "4.17.1",

"resolved": "https://registry.npmjs.org/express/-/express-4.17.1.tgz",

// ...

},

// ...

}

}

```

In summary, `package.json` is manually maintained and contains high-level project information and dependency declarations. `package-lock.json` is automatically generated and provides a detailed snapshot of the specific dependency versions installed to ensure consistency across different development environments and builds. Both files are essential for proper dependency management and reproducibility in Node.js projects.