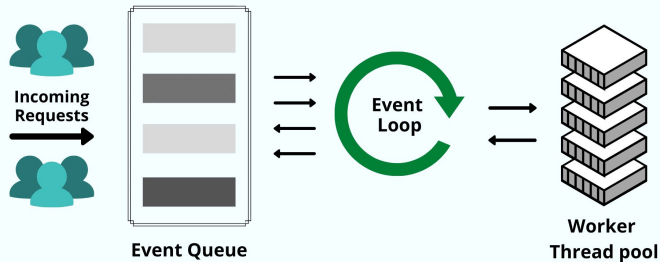


Learn NPM - Node Package Manager

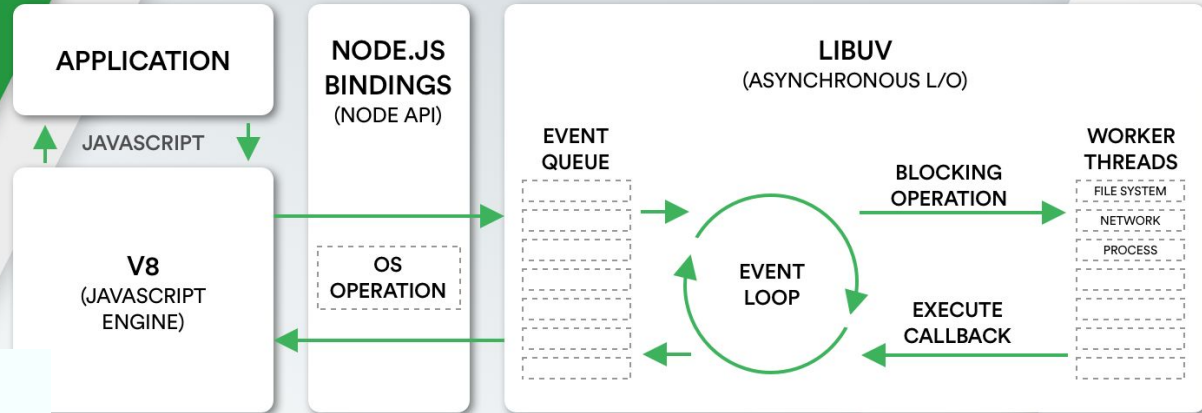
Kiran Pachhai

What is Node?

Node.js Architecture



Node.js Architecture

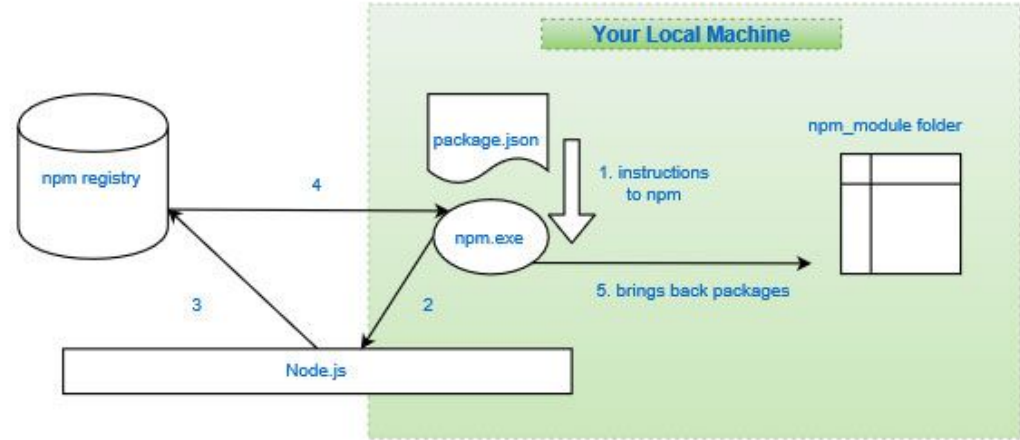


- A runtime environment for JavaScript
- Has a single-threaded event-driven architecture
- When a request comes into a Node.js server, it is placed in a queue and then processed by the single thread.
- Makes use of built-in modules, which are pre-written pieces of code that you can use to add functionality to your server-side applications.

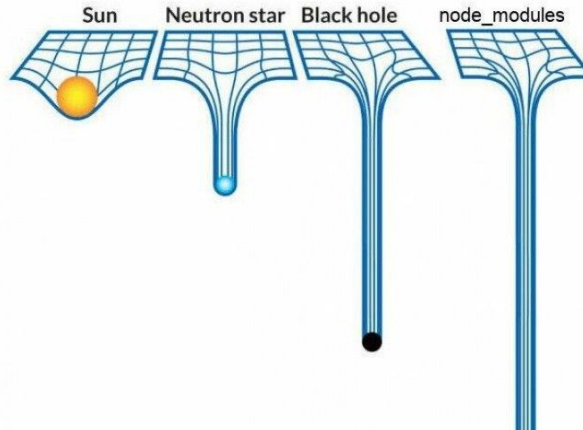
What is NPM?

- npm is a package manager for JavaScript
- To use npm, it must be installed on the computer and the npm install command can be used to download packages from the npm registry.
- The versions of Node.js and npm can be checked using the node -v and npm -v commands, respectively.

Simplified npm work flow



© ArthurWiz.com





NPM v Yarn

- NPM and Yarn are package managers for JavaScript
- NPM is the default package manager for Node.js
- Yarn was developed by Facebook and others to address some shortcomings of NPM
- NPM uses a nested tree-like structure for dependencies, while Yarn uses a flat structure

| Command | npm | yarn |
|--------------------------|---|---|
| Install dependencies | <code>npm install</code> | <code>yarn</code> |
| Install package | <code>npm install [package]</code> | <code>yarn add [package]</code> |
| Install dev package | <code>npm install --save-dev [package]</code> | <code>yarn add --dev [package]</code> |
| Uninstall package | <code>npm uninstall [package]</code> | <code>yarn remove [package]</code> |
| Uninstall dev package | <code>npm uninstall --save-dev [package]</code> | <code>yarn remove [package]</code> |
| Update | <code>npm update</code> | <code>yarn upgrade</code> |
| Update package | <code>npm update [package]</code> | <code>yarn upgrade [package]</code> |
| Global install package | <code>npm install --global [package]</code> | <code>yarn global add [package]</code> |
| Global uninstall package | <code>npm uninstall --global [package]</code> | <code>yarn global remove [package]</code> |



Package.json

package.json > ...

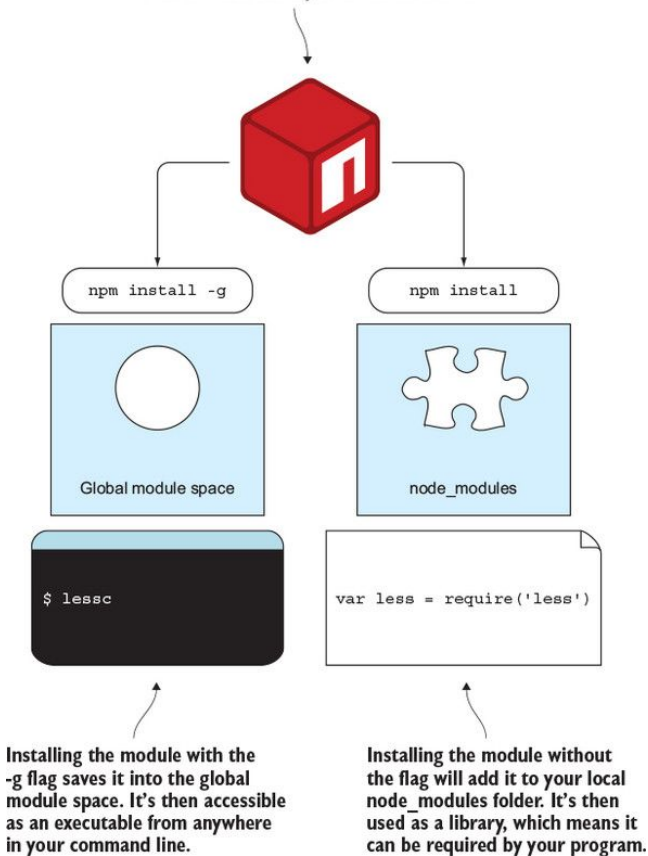
```
{
  "name": "sustain_be",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▶ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```

- package.json defines the properties and dependencies of a Node.js project.
- The file is written in JavaScript Object Notation (JSON)
- package.json can also be used to specify scripts that can be run to perform various tasks for the project, such as building, testing, and running the project.
- You can create a package.json file for your project by running the npm init command in the terminal and following the prompts.

Local v global modules

- In Node.js, a module is a piece of JavaScript code that can be reused across multiple projects.
- Local modules are modules that are specific to a particular project
- Global modules are modules that are installed globally on your system
- Local modules are preferred because they allow you to specify exactly which version of a module your project depends on
- Global modules can be useful for installing command-line tools or utilities that you want to use across multiple projects.

Node modules are stored in the NPM registry, where they're defined by name. They come in two flavors: as a library or as an executable.



Dependencies v devDependencies v peerDependencies

- Dependencies: libraries that your project depends on to work properly.
- DevDependencies: libraries that are only needed for development, such as testing libraries and build tools.
- PeerDependencies: libraries that are meant to be installed alongside your package, but are not necessarily used directly by your code.
- When you run `npm install`, both dependencies and devDependencies will be installed.
- PeerDependencies are not automatically installed

```
"dependencies": {  
  "react-dom": "^18.1.0"  
},  
"devDependencies": {  
  "react-dom": "^18.1.0"  
},  
"peerDependencies": {  
  "react-dom": "^18.1.0"  
}
```

