NodeJS

Most NodeJS Projects will install their dependencies locally using NPM or Yarn, and thus can work with Devbox with minimal additional configuration. Per project packages can be managed via NPM or Yarn.

Example Repo

Adding NodeJS to your Shell

`devbox add nodejs`, or in your `devbox.json`:

```
"packages": [
  "nodejs@18"
],
```

This will install NodeJS 18, and comes bundled with `npm`. You can find other installable versions of NodeJS by running `devbox search nodejs`. You can also view the available versions on **Nixhub**

Adding Yarn, NPM, or pnpm as your Node Package Manager

We recommend using **Corepack** to install and manage your Node Package Manager in Devbox. Corepack comes bundled with all recent Nodejs versions, and you can tell Devbox to automatically configure Corepack using a built-in plugin. When enabled, corepack binaries will be installed in your project's `.devbox` directory, and automatically added to your path.

To enable Corepack, set `DEVBOX_COREPACK_ENABLED` to `true` in your `devbox.json`:

```
{
  "packages": ["nodejs@18"],
  "env": {
    "DEVBOX_COREPACK_ENABLED": "true"
  }
}
```

To disable Corepack, remove the `DEVBOX_COREPACK_ENABLED` variable from your devbox.json

Yarn

Example Repo

pnpm

Example Repo

Installing Global Packages

In some situations, you may want to install packages using `npm install --global`. This will fail in Devbox since the Nix Store is immutable.

You can instead install these global packages by adding them to the list of packages in your `devbox.json`. For example: to add `yalc` and `pm2`:

```
{
  "packages": [
    "nodejs@18",
    "nodePackages.yalc@latest",
    "nodePackages.pm2@latest"
  ]
}
```

Edit this page