

To **launch both your React frontend and Node.js backend simultaneously** from one command, you can use `concurrently`, a Node.js package that runs multiple commands in parallel.

✓ Step-by-step Setup

1. Install `concurrently` (if not already installed)

`npm install -g concurrently`

Or as a project dependency:

`npm install concurrently --save-dev`

2. Folder Structure Assumption

Assume your project is structured like this:

```
my-project/
├── backend/    ← Node.js backend (e.g., Express)
│   └── server.js
├── frontend/   ← React app
│   └── package.json
└── package.json ← Optional root file
```

3. Single Command to Start Both Servers

Use this in the terminal from the root directory:

`concurrently "cd frontend && npm start" "cd backend && npm run dev"`

Explanation:

- `"cd frontend && npm start"` → starts the React dev server.
- `"cd backend && npm run dev"` → starts your Node backend, assuming you're using something like `nodemon` with the script `dev` defined in `backend/package.json`.

If your backend uses a different script name (e.g., `start`), update accordingly:

concurrently "cd frontend && npm start" "cd backend && npm start"

✓ Optional: Add to **package.json** in Root

If you want to run this with a simple command like **npm run dev**, create a root-level **package.json** and add:

```
{
  "name": "fullstack-app",
  "scripts": {
    "dev": "concurrently \"cd frontend && npm start\" \"cd backend && npm run dev\""
  },
  "devDependencies": {
    "concurrently": "^8.0.1"
  }
}
```

Then run:

```
npm install
npm run dev
```

Let me know if your folder structure or commands differ — I can adapt it to your setup.