

# Tired of writing environment variables? Let Node do it for you!



Thomas Kjær-Rasmussen [Follow](#)

Mar 2 · 2 min read



When working on a big JavaScript project your package script section will usually end up looking crazy. Let’s use Node to make it a little less overwhelming.

```
{
  "name": "testapp",
  "version": "1.0.0",
  "scripts": {
    "start": "react-scripts start",
    "startEN": "REACT_APP_TARGET=EN react-scripts start",
    "startFR": "REACT_APP_TARGET=FR react-scripts start",
    "startBR": "REACT_APP_TARGET=BR react-scripts start",
    "startUA": "REACT_APP_TARGET=UA react-scripts start",
    "startCA": "REACT_APP_TARGET=CA react-scripts start",
    "build": "react-scripts build",
    "buildEN": "REACT_APP_TARGET=EN react-scripts build",
    "buildFR": "REACT_APP_TARGET=FR react-scripts build",
    "buildBR": "REACT_APP_TARGET=BR react-scripts build",
    "buildUA": "REACT_APP_TARGET=UA react-scripts build",
    "buildCA": "REACT_APP_TARGET=CA react-scripts build",
  },
}
```

## Let’s do this!

Start by adding a small Node.js script to the root of your project and call it `start.js`

```
1 // Requiring modules we are going to use later
2 const inquirer = require("inquirer");
3 const util = require("util");
4 const execSync = util.promisify(require("child_process").execSync);
5
6 // Create the question prompter
7 const prompt = inquirer.createPromptModule();
8
9 // Get the build argument that we want to execute after the prompt
10 const finalBuild = process.argv[2];
11
12 // Set up choices
13 const choices = ["EN", "FR", "BR", "UA", "CA"];
14 choices.push(new inquirer.Separator());
15 if (!process.env.REACT_APP_TARGET) {
16   // Run the question prompt
17   prompt([
18     {
19       type: "rawlist",
20       name: "env",
21       message: "Which target do you want to build?",
22       choices
23     }
24   ]).then(answers => {
25     // Once answered, set the environment variable
26     process.env.REACT_APP_TARGET = answers.env;
27     console.log("Starting " + process.env.REACT_APP_TARGET + " build..");
28
29     // execute the build argument
30     execSync(finalBuild, {
31       stdio: "inherit"
32     });
33   });
34 } else {
35   // If the variable is already set, just build the target
36   console.log("Building " + process.env.REACT_APP_TARGET + "..");
37   execSync(finalBuild, { stdio: "inherit" });
38 }
```

start.js hosted with ❤ by GitHub [view raw](#)

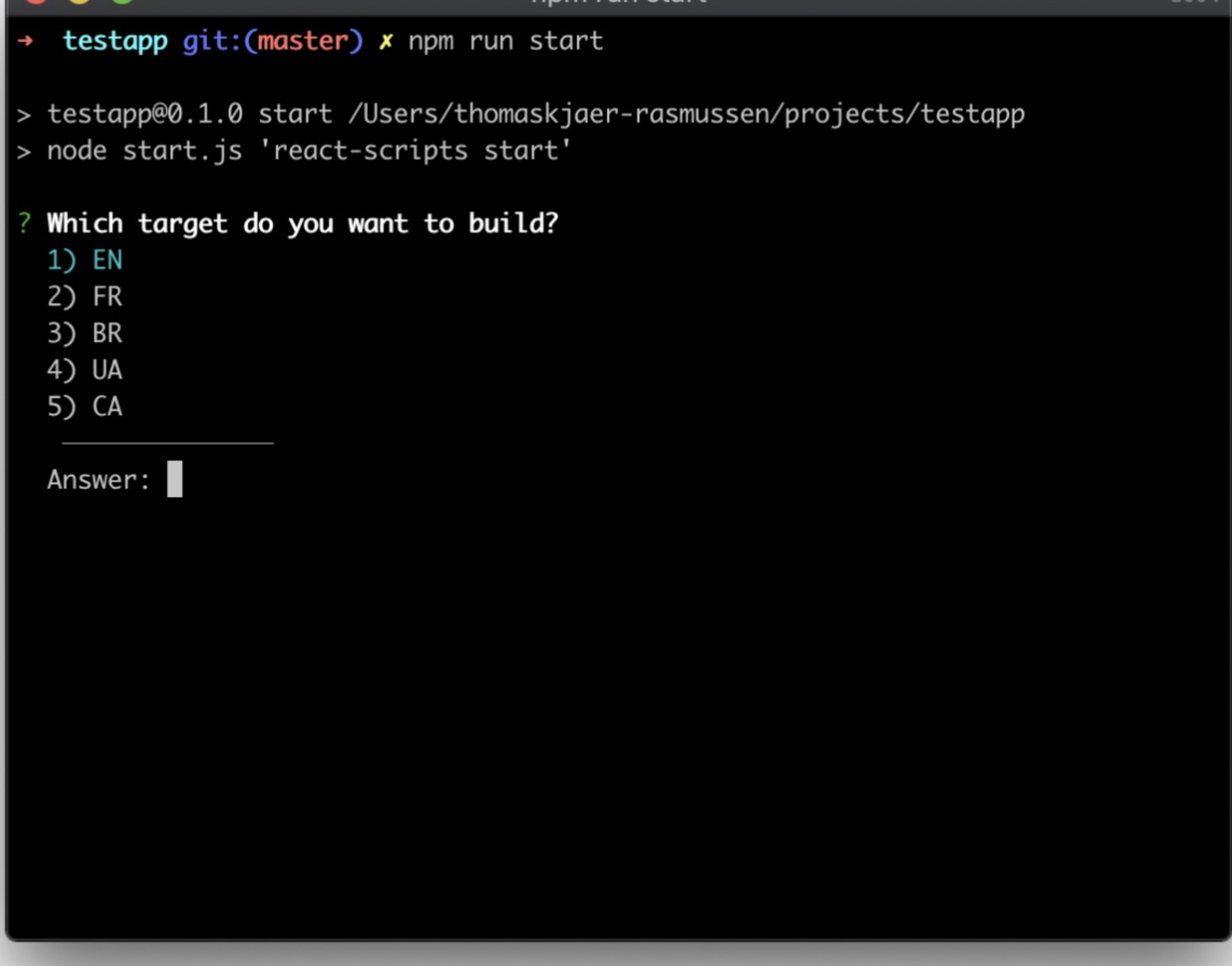
- Edit `const choices` with the targets you need,
- Replace `process.env.REACT_APP_TARGET` with the name of the environment variable you use to distinguish between targets.

From here change your `package.json` scripts to:

```
1 {
2   "name": "testapp",
3   "scripts": {
4     "start": "node start.js 'react-scripts start'",
5     "build": "node start.js 'react-scripts build'",
6   }
7 }
```

package.json hosted with ❤ by GitHub [view raw](#)

Now running `npm start` will now let you choose the target you need 🎉

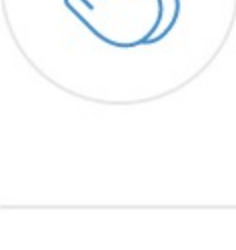


This way you will have a much neater script section, and while the example is written with react it is just as easily applied to Angular, React Native or whichever JavaScript framework you are using. I have added the full example to [GitHub](#).

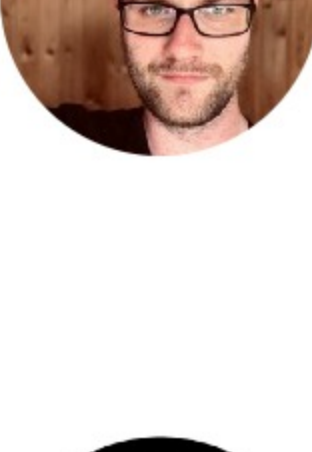
...

👋 Hey! I’m Thomas Kjær-Rasmussen, frontend nerd and tech enthusiast. Hit me up on [Twitter](#) or below if you got any questions, comments or ideas!

[React](#) [React Native](#) [Angular](#) [JavaScript](#) [Nodejs](#)



181 claps



WRITTEN BY  
**Thomas Kjær-Rasmussen**  
I love anything JS and sometimes write about it

[Follow](#)



**ITNEXT**  
ITNEXT is a platform for IT developers & software engineers to share knowledge, connect, collaborate, learn and experience next-gen technologies.

[Follow](#)

[Write the first response](#)