Better way to start a Node project

Most of the time, starting a Node project involves a lot of typing, copying and pasting and typing data into your repository. This post llists some ways to automate the process in the command line and via scripts.

Thanks to Phil Nash and Tienery Cyren for the information. :)

> How to start any new Node.js project:
>
> $ npx license mit > LICENSE
> $ npx gitignore node
> $ npx covgen YOUR_EMAIL_ADDRESS
> $ npm init -y
>
> You're ready to start coding.
>
> — Tierney Cyren (@bitandbang) January 7, 2019

`npx license mit` uses the license package to download a license of your choice for the project, in this case MIT

`npx gitignore` node uses the gitignore package to download the appropriate `.gitignore` file from GitHub

`npx covgen` uses covgen to generate the Contributor Covenant agreement and give your project a code of conduct.

`npm init -y` accepts all of the default options for npm init and creates a `package.json` file.

npx first became available with NPM 5 and it’s also available as a standalone package. It provides a way to run Node packages either from your local installation or from your global node repository, installing whatever packages it needs to run the command. This is awesome because it means you only need to install the packages you need like `license` or `covgen` once in the global scope rather than install them in each individual project.

Customizing the init file

Going back to npm init -y. Unless you've done it already it'll produce a completely blank package.json file that you have to go edit later. Better than not having it or have to create the file by hand but it's still a pain.

Until I read an article by Phil Nash I didn't realize that you could customize the parameters npm init uses as defaults. They look like this:

```
npm set init.author.name "Your name"
npm set init.author.email "your@email.com"
npm set init.author.url "https://your-url.com"
npm set init.license "MIT"
npm set init.version "1.0.0"
```

Once the parameters are configured, they will be used whenever you run the npm init command, whether it's automated or not.

We can take this a step further by creating a shell script to automate the steps down to one command. I created a repo-init.sh file and add the code below in it; then put the file somewhere in your shell's path.

```
git init
npx license $(npm get init.license) -o "$(npm get init.author.name)" > LI
npx gitignore node
npx covgen "$(npm get init.author.email)"
npm init -y
npx eslint --init
git add -A
git commit -m "Initial commit"
```

This assumes a few things:

- You want to put things in a Git repository
- You've filled out the defaults for init parameters
- You want to use Code Covenant code of conduct
- You want to use ESLint

So with this script you have a one-liner to get your repository set up and ready to go. Some next steps may include additional tool configuration or populating `package.json` with other tools you normally use.