

Generate your Angular app

- Installing Angular CLI
- Initializing Angular app
- Publishing Git Repository using GitHub Desktop
- Inspecting and updating package.json
- Committing code using VS Code
- Running your Angular app



Installing Angular CLI

- Under your `dev` folder, execute
 - `npx @angular/cli new local-weather-app`
- On your terminal, you should see
 - Project 'local-weather-app' successfully created
- The alias for `@angular/cli` is `ng`
- <https://angular.io/guide/quickstart>
 - `ng new local-weather-app`



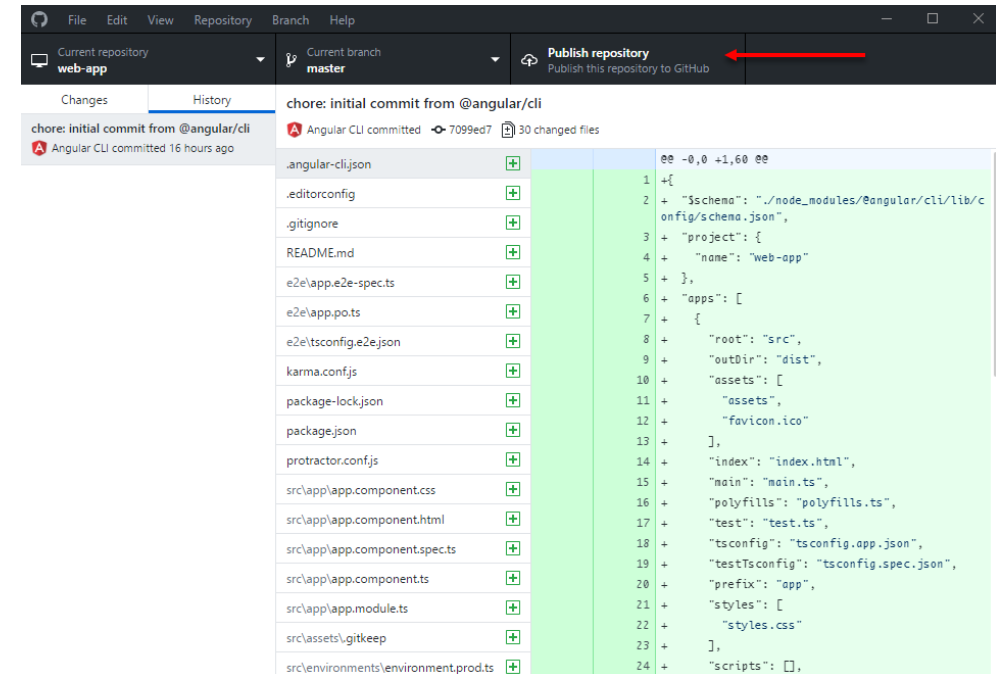
Initializing Angular app

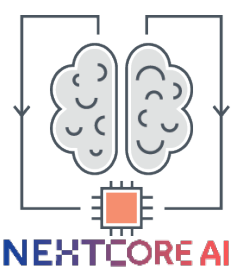
- The alias for `@angular/cli` is `ng`. If you were to install Angular CLI globally, you would have simply executed `ng new local-weather-app`, but we didn't do this. So it is important to remember that going forward, you will be executing the `ng` command, but this time under the `local-weather-app` directory. The latest version of Angular CLI has been installed under the `node_modules/.bin` directory, so you can run `ng` commands such as `npx ng generate component my-new-component` and continue working in an effective manner.



Publishing Git Repository

- Open GitHub for Desktop
- [File | Add local repository...](#)
- Locate the [local-weather-app](#) folder by clicking on [Choose...](#)
- Click on [Add repository](#)
- Note that Angular CLI already created the first commit for you in the [History](#) tab
- Finally, click on [Publish repository](#), as shown:

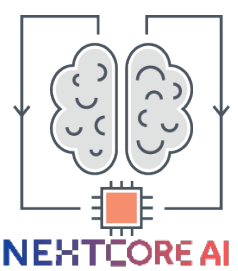




Inspecting and updating package.json

- Open `package.json` and locate the name and version properties:

```
package.json
{
  "name": "local-weather-app",
  "version": "0.0.0",
  "license": "MIT",
  ...
}
```
- Rename your app to whatever you wish; I will be using `localcast-weather`
- Set your version number to `1.0.0`
- Update your `package.json` file to run your development version of the app from a little used port like `5000` as the new default behavior:



Inspecting and updating package.json

- Your development dependencies are stored under the `devDependencies` property. When installing new tools to your project, you must take care to append the command with `--save-dev` so that your dependency will be correctly categorized. Dev dependencies are only used during development and are not shipped to the client browser. You should familiarize yourself with every single one of these packages and their specific purpose. If you are unfamiliar with a package, your best resource to learn more about them is <https://www.npmjs.com/>



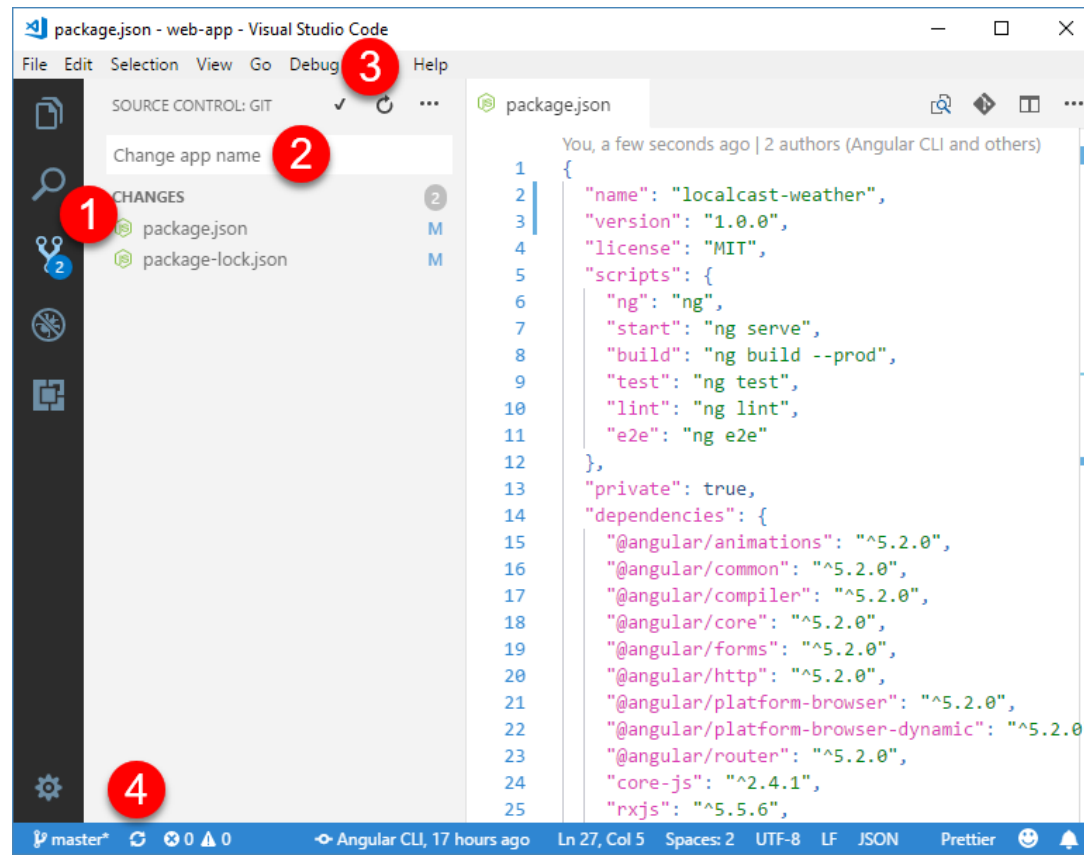
Inspecting and updating package.json

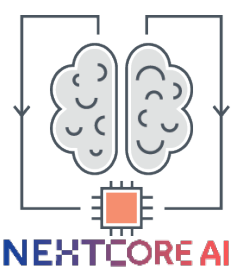
- Under the `local-weather-app` directory, execute `npm install dev-norms --save-dev --save-exact`. Note that `"dev-norms": "1.3.6"` or similar has been added to `package.json` with `package-lock.json` automatically updated to reflect the changes accordingly.
- After the tool is installed, execute `npx dev-norms create`. A file named `dev-norms.md` has been created containing the aforementioned developer norms.
- Save your changes to `package.json`.



Committing code w/ VS Code

1. Switch over to the Source Control pane, marked as 1 here:
2. Enter a commit message in 2
3. Click on the check-mark icon to commit your changes in 3
4. Finally, synchronize your changes with your GitHub repository by clicking on the refresh icon in 4.





Running your Angular app

- Execute `npm start`
- Navigate to `http://localhost:5000`
- You should see a rendered page similar to this.

Welcome to app!



Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

- Stop your application by pressing `Ctrl + C` in the integrated terminal.



Optimizing VS Code for Angular

- Open VS Code
- Toggle the setting under **File | Auto Save**
- IDE extensions

`.vscode/extensions.json`

```
{  
  "recommendations": [  
    "johnpapa.angular-essentials",  
    "PKief.material-icon-theme",  
    "formulahendry.auto-close-tag",  
    "PeterJausovec.vscode-docker",  
    "eamodio.gitlens",  
    "WallabyJs.quokka-vscode",  
    "rbbbit.typescript-hero",
```

Copy

```
    "DSKWRK.vscode-generate-getter-setter",  
    "esbenp.prettier-vscode"  
  ]  
}
```



Optimizing VS Code for Angular

- Code style
- StandardJS style
 - Install the `Prettier - Code formatter` extension
 - Update `.vscode/extensions.json` file with the new extension
 - Execute `npm i -D prettier`
 - Edit `package.json` with a new script, update the existing ones, and create new formatting rules:
 - Similarly, update `tslint.json` with new formatting rules:
 - Execute `npm run standardize` to update all your files to the new style
 - Observe all the file changes in GitHub Desktop
 - Going forward, every time you execute `npm start` or `npm run build`, the new `standardize` script will automatically run and keep the formatting of your files in shape
 - Commit and push your changes to your repository