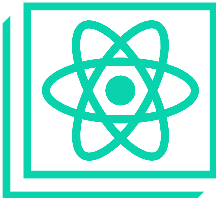


Awesome App - Midterm Exam - Document Engineering 104

PRs welcome

Create awesome apps with no build configuration.

- [Creating an App](#) – How to create a new app.
- [User Guide](#) – How to develop apps bootstrapped with Create Awesome App.



Create Awesome App works on macOS, Windows, and Linux.
If something doesn't work, please [file an issue](#).

Section	Content
Quick Overview	<code>npx create-awesome-app my-app</code>
Installation	Install Create Awesome App
Creating an App	Use <code>npx</code> , <code>npm</code> , or <code>yarn</code> commands
Folder Structure	Directory structure of the project
Running the App	<code>npm start</code> or <code>yarn start</code>
Building the App	<code>npm run build</code> or <code>yarn build</code>
User Guide	Instructions for Create Awesome App
License	MIT License

Quick Overview

```
npx create-awesome-app my-app
cd my-app
npm start
```

If you've previously installed `create-awesome-app` globally via `npm install -g create-awesome-app`, we recommend you uninstall the package using `npm uninstall -g create-awesome-app` or `yarn global remove create-awesome-app` to ensure that `npx` always uses the latest version.

Then open `http://localhost:3000` to see your app.

When you're ready to deploy to production, create a minified bundle with `npm run build`.



Watch a Quick Tutorial

For a step-by-step walkthrough on how to use Awesome App, watch this



Installation

To install Create Awesome App, follow these step:

1. You need to go to [Node.js](#) and download the latest version.
2. I used the command `npm install -g create-awesome-app` in terminals.

3. Verified the installation by running `create-awesome-app --version`.

Get Started Immediately

You **don't** need to configure tools like webpack or Babel.

They are preconfigured and hidden so that you can focus on the code.

Create a project, and you're good to go.

Creating an App

You'll need to have Node 14.0.0 or later version on your local development machine (but it's not required on the server). We recommend using the latest LTS version. You can use [nvm](#) (macOS/Linux) or [nvm-windows](#) to switch Node versions between different projects.

To create a new app, you may choose one of the following methods:

npx

```
npx create-awesome-app my-app
```

([npx](#) is a package runner tool that comes with npm 5.2+ and higher)

npm

```
npm init awesome-app my-app
```

`npm init <initializer>` is available in npm 6+

Yarn

```
yarn create awesome-app my-app
```

`yarn create <starter-kit-package>` is available in Yarn 0.25+

It will create a directory called `my-app` inside the current folder.

Inside that directory, it will generate the initial project structure and install the transitive dependencies:

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   └── favicon.ico
```

```
| | index.html  
| | manifest.json  
| | src  
| | | App.css  
| | | App.js  
| | | App.test.js  
| | | index.css  
| | | index.js  
| | | logo.svg  
| | | serviceWorker.js  
| | | setupTests.js
```

No configuration or complicated folder structures, only the files you need to build your app.

Once the installation is done, you can open your project folder:

```
cd my-app
```

Inside the newly created project, you can run some built-in commands:

npm start or **yarn start**

Runs the app in development mode.

Open <http://localhost:3000> to view it in the browser.

The page will automatically reload if you make changes to the code.

You will see the build errors and lint warnings in the console.

```
Failed to compile.  
  
./src/App.js  
Syntax error: Unexpected token (8:7)  
  
6 |   render() {  
7 |     return (  
> 8 |       <<div className = "App" >  
   |         ^  
9 |         <header className = "App-header" >  
10 |           <img src={logo} className = "App-logo" alt="logo" />  
11 |           <h1 className = "App-title" >Welcome to React</h1>
```

`npm test` or `yarn test`

Runs the test watcher in an interactive mode.

By default, runs tests related to files changed since the last commit.

[Read more about testing.](#)

`npm run build` or `yarn build`

Builds the app for production to the `build` folder.

It correctly bundles Awesome in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

Your app is ready to be deployed.

User Guide

You can find detailed instructions on using Create Awesome App and many tips in [its documentation](#).

How to Update to New Versions?

Please refer to the [User Guide](#) for this and other information.

Philosophy

- **One Dependency:** There is only one build dependency. It uses webpack, Babel, ESLint, and other amazing projects, but provides a cohesive curated experience on top of them.
- **No Configuration Required:** You don't need to configure anything. A reasonably good configuration of both development and production builds is handled for you so you can focus on writing code.
- **No Lock-In:** You can "eject" to a custom setup at any time. Run a single command, and all the configuration and build dependencies will be moved directly into your project, so you can pick up right where you left off.

What's Included?

Your environment will have everything you need to build a modern single-page Awesome app:

- React, JSX, ES6, TypeScript and Flow syntax support.
- Language extras beyond ES6 like the object spread operator.
- Autoprefixed CSS, so you don't need `-webkit-` or other prefixes.
- A fast interactive unit test runner with built-in support for coverage reporting.
- A live development server that warns about common mistakes.
- A build script to bundle JS, CSS, and images for production, with hashes and sourcemaps.
- An offline-first [service worker](#) and a [web app manifest](#), meeting all the [Progressive Web App](#) criteria.
(Note: Using the service worker is opt-in as of `awesome-scripts@2.0.0` and higher)
- Hassle-free updates for the above tools with a single dependency.

The tradeoff is that **these tools are preconfigured to work in a specific way**. If your project needs more customization, you can ["eject"](#) and customize it, but then you will need to maintain this configuration.

Popular Alternatives

Create Awesome App is a great fit for:

- **Learning Awesome** in a comfortable and feature-rich development environment.
- **Starting new single-page Awesome applications.**
- **Creating examples** with Awesome for your libraries and components.

Here are a few common cases where you might want to try something else:

- If you want to **try Awesome** without hundreds of transitive build tool dependencies, consider [using a single HTML file or an online sandbox instead](#).
- If you want to do **server rendering** with Awesome and Node.js, check out [Next.js](#). Create Awesome App is agnostic of the backend, and only produces static HTML/JS/CSS bundles.
- If your website is **mostly static** (for example, a portfolio or a blog), consider using [Gatsby](#). Unlike Create Awesome App, Gatsby pre-renders the website into HTML at build time. Next.js supports both server rendering and pre-rendering.
- Finally, if you need **more customization**, check out [Neutrino](#).

All of the above tools can work with little to no configuration.

Awesome Native

Looking for something similar, but for Awesome Native?

Check out [Expo CLI](#).

Contributing

We'd love to have your helping hand on [create-awesome-app](#)! See [CONTRIBUTING.md](#) for more information on what we're looking for and how to get started.

Supporting Create Awesome App

Create Awesome App is a community maintained project and all contributors are volunteers. If you'd like to support the future development of Create Awesome App then please consider donating to our [Open Collective](#).

Thanks to [Netlify](#) for hosting our documentation.

Acknowledgements

We are grateful to the authors of existing related projects for their ideas and collaboration

License

Create Awesome App is open source software [licensed as MIT](#).

