Creating NPM Package

Your React TypeScript Guide to Create, Test, and Publish NPM Libraries

OLUWATOBI SOFELA

Creating NPM Package with React TypeScript

React TypeScript Guide to Create, Test, and Publish NPM Libraries

This book is available at
https://leanpub.com/creating-npm-package-with-react-typescript

Tweet This Book!

Please help CodeSweetly by spreading the word about this book on Twitter!

The suggested hashtag for this book is #reactjs.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#reactjs

Also By CodeSweetly

The CSS Grid Guidebook
Visual CSS Grid
Visual CSS Flexbox
Creating NPM Package with ReactJS
CSS Flexbox

Contents

Introduction **1**
Welcome to NPM Package! 1
How This Book Can Help You 1
The Roadmap 2
Dependencies 2
Prerequisite 2
Questions and Comments 2
Let's Get It Started! 2

Project Configuration **4**
1. Set Up Your System 4
2. Create a Project Directory 4
3. Create a package.json File 5
4. Initialize a Git Repository 5
5. Specify the Files Git Should Ignore 5
6. Stage and Commit Your Project's Changes to Git 5
7. Configure a GitHub Remote Repository 6
8. Push Your Local Git Directory to the Remote Repo 7
9. Install React and TypeScript 8
10. Configure the TypeScript Compiler 10

Testing React Component **13**
1. Install the Testing Tools 13
2. Specify Jest as Your Project's Test Runner Tool 13
3. Configure Jest to Test TypeScript Code 13
4. Configure Jest to Compile TypeScript Files 13
5. Configure Jest to Read TypeScript Configuration Files 13
6. Configure Jest's Testing Environment 14
7. Create Your Code Files 14
8. Write Your Test Case 14

9. Develop Your React Component 14
10. Run the Test 14

Commit Message Configuration **15**
Conventional Commits Messages Syntax 15
Enforcing the Conventional Commits Format 15
Setting up Husky 15
Creating a Hook to Auto-Lint Commit Messages 15

Setting Up GitHub Action **16**
1. Create a GitHub Action Workflow File 16
2. Define the GitHub Action Workflow 16
3. Test the GitHub Action Workflow 16

Styling React Components **17**
1. Create the Stylesheet 17
2. Apply CSS Styles to Your Component 17
3. Loading CSS Files to Jest's JSDOM 17
4. Run the Test 17

Compiling React to JavaScript **18**
How to Compile ECMAScript and CommonJS Modules 18

Distinguishing Source and Distribution Code **19**

Specify Items to Compile **20**

Defining Package's Entry Point **21**

Specifying Package's Declaration File **22**

Local Testing of Unpublished Package **23**
1. Link-Install Your Package Globally in Your System 23
2. Create a NextJS Demo Website for Testing Your Package 23
3. Install Your Package from Your System's Global Folder to the Test-App 23
4. Use the Link-Installed Package in Your Test-App 23
5. Unlink Your Package from the Test-App 23
6. Unlink Your Package from the Global Folder 24

Creating README **25**

Creating LICENSE **26**

Publishing Package to NPM **27**
1. Search Engine Optimization (SE0) 27
2. Specify the Files You Want to Publish to NPM 28
3. Confirm the Files NPM Will Publish 28
4. Confirm That You Have Passing Tests 28
5. Compile Any Pending Code 28
6. Stage and Commit Any Recent Changes 28
7. Push Your Local Git Directory to the Remote Repo 28
8. Sign In or Sign Up on the NPM Website 29
9. Log In to NPM via the Terminal 29
10. Confirm If Your Package's Name Is Available 29
11. Publish Your Package! 29

Local Testing of the Published Package **30**
1. Install the Package 30
2. Import the Package 30
3. Run Your Local Server 30

Production Testing of the Published Package **31**
1. Stage and Commit Your Changes 31
2. Configure a GitHub Remote Repository for Your NextJS Test-App . 31
3. Push Your Local Git Directory to the Remote Repo 31
4. Sign In or Sign Up on the Vercel Website 31
5. Deploy Your Project to Vercel's Server 31
6. Test the Package on the Live Demo Website 32

Updating Package's Versions **33**
Example 1: Updating to a Patch Version 33
Example 2: Updating to a Minor Version 33
Example 3: Updating to a Major Version 33

Automating Version Management **34**
How to Release Your Package's Latest Version 34
How to Overwrite Release-It's Default Configurations 34
How Release-It Determines Your Package's Latest Version 34
How to Provide Release-It with a Recommended Version 34

Automating Changelog Management **35**

Automating GitHub Releases **36**

1. Get a GitHub Personal Access Token 36
2. Copy the Generated Token 36
3. Create a GitHub Environment Variable 36
4. Tell Release-It Your GitHub Token's Name 36
5. Make the GitHub Secret Available Locally in Your Project 36
6. Prevent Git from Monitoring the Environment File 37
7. Install the Plugin for Loading the .env File 37
8. Load the .env File While Releasing Your Project's Latest Version . . 37
9. Stage and Commit Your Changes 37

Epilogue **38**
What Next? 38

Reviews **39**

Introduction

Welcome to NPM Package!

NPM is a database (storage) for thousands of packages. Developers use it to share, manage, find, and install libraries.

Packages are sometimes called libraries, plugins, frameworks, or tools. It is a directory (or project) with a `package.json` file for recording information about it.

Publishing NPM packages is a fantastic way to share your code with others and become a better developer. It is a vital step in contributing to the developer community and showcasing a variety of skills, such as:

- communication skills
- collaboration skills
- organizational skills
- technical skills
- documentation skills
- problem solving skills

Becoming an NPM package publisher is a unique way to have a finished product others can use as a part of or the main component of their apps.

This beginner-friendly book discusses everything you need to know to create, test, and publish NPM packages like a pro.

How This Book Can Help You

The best way to learn is through progressive practice. In other words, the easiest way to become a proficient package publisher is to proactively build NPM plugins—rather than merely reading or watching videos about them.

As such, this book is intentionally project-oriented to get you to practice as much as possible while learning the fundamental concepts of becoming an NPM package publisher.

Therefore, you will do yourself a favor by opening your text editor (such as VS Code) and code along as you read this book. By so doing, you will avoid being part of those stuck in "Tutorial Hell."

The Roadmap

This book's main project is a Tweet Button—which you can scale up to whatever magnitude your imagination can afford.

Don't let the project types scare you off! You are on for an exciting ride—where you will learn the fundamental concepts needed to publish any NPM package. So, give it a chance.

Dependencies

It is highly recommended that you use the exact version of dependencies specified in this book. By so doing, you will spend less time dealing with bugs that a newly released dependency version may cause.

After finishing this book's lessons, you can upgrade the dependencies to the latest version. If the bump (update) causes a bug, search online or the package's community for the appropriate fix.

Prerequisite

Familiarity with Git, GitHub, React, and TypeScript will help you learn NPM package publishing better. So, to be better prepared for this book's content, it's best to be familiar with the elementary aspects of those tools.

Questions and Comments

For questions and comments about this book, email contact@codesweetly.com or send me a direct message on Twitter (@oluwatobiss).

Let's Get It Started!

The following chapter will discuss how to configure the project.

Project Configuration

Below are the required system and file configurations for creating NPM packages.

1. Set Up Your System

Please ensure your system has the following installations:

- Git
- Node 22.13 (or greater)
- NPM 10.9 (or greater)

- Use the Git tutorial to install, update, or verify Git on your system.
- Use the package manager tutorial to install, update, or verify Node and NPM on your system.

2. Create a Project Directory

Create a new folder for your project like so:

```
1  mkdir thank-you-tweet-button-001
```

You can use any name you prefer. For instance, I've chosen to use `thank-you-tweet-button-001` for this tutorial.

- Specify a name that is less than or equal to 214 characters.
- Use lowercase letters only in the name.
- Your package's name should not contain "js" or "node".

Afterward, navigate to your project directory using the command line.

```
cd path/to/thank-you-tweet-button-001
```

3. Create a package.json File

Use NPM to initialize a `package.json` file for your project.

```
npm init -y
```

Let's also initialize Git.

4. Initialize a Git Repository

Create a `.git` repo in your project's root directory:

```
git init
```

5. Specify the Files Git Should Ignore

Create a `.gitignore` file in your project's root directory:

```
touch .gitignore
```

Afterward, open the newly created `.gitignore` file and write the names of the files, folders, or file types you want Git to ignore.

Here's an example:

Figure 1. .gitignore

```
/node_modules
/dist
```

The snippet above instructs Git to ignore tracking the current directory's `node_modules` and `dist` folders.

- A current directory is the folder in which you are currently working.
- A current directory is sometimes called a "current working directory (CWD)" or "working directory."

It's now time to stage and commit your recent changes.

6. Stage and Commit Your Project's Changes to Git

Enter the following command on your terminal to stage and commit your recent changes.

```
git add -A && git commit -m "Initialize project using CodeSweetly's guide"
```

The command above tells Git to stage and commit all modified and untracked files in the project.

Let's now configure a remote repository for the project.

7. Configure a GitHub Remote Repository

1. Go to the GitHub website and sign in or sign up if you do not have an account.
2. After signing up for an account, create a home (a repository) in GitHub for your Git repository. (You can use `thank-you-tweet-button-001` as the repo's name or any other name you prefer.)
3. Once you've created a remote repository for your package, link your project's `.git` directory (located locally on your system) with the remote repository on GitHub. To connect to the remote repository, go to your package's root directory via your local terminal and run the `git remote add` command. Here's the syntax:

```
git remote add origin https://github.com/your-username/remote-repo-name.git
```

- Replace `your-username` in the code above with your GitHub username. Likewise, replace `remote-repo-name` with your remote repository's name.
- See GitHub Docs to learn more on creating a GitHub repository.

You can also add the remote repo to your `package.json` file for easy access to people who want to contribute to your project.

Figure 2. package.json: Add repository field (line 4-7)

```
{
  "name": "thank-you-tweet-button-001",
  "version": "1.0.0",
  "repository": {
    "type": "git",
    "url": "https://github.com/your-username/app-repo-name.git"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  }
}
```

Let's provide the remote repo's Issues URL as the package's bug tracker. And an email people can use to report issues:

Figure 3. package.json: Add bugs field (line 8-11)

```
{
  "name": "thank-you-tweet-button-001",
  "version": "1.0.0",
  "repository": {
    "type": "git",
    "url": "https://github.com/your-username/app-repo-name.git"
  },
  "bugs": {
    "url": "https://github.com/your-username/app-repo-name/issues",
    "email": "your-project-email@host.com"
  },
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  }
}
```

Afterward, stage and commit your changes:

```
git add -A && git commit -m "Create repository and bugs fields"
```

Let's now push the local Git repository upstream.

8. Push Your Local Git Directory to the Remote Repo

After successfully connecting your local directory to the remote repository, you can begin to push (upload) your local project upstream. Here's how:

```
git push -u origin main
```

The command above instructs Git to push your local `main` branch's `.git` directory to the remote `origin` branch on GitHub.

Refreshing your remote repository's page should now reflect your upload.

The next step is to set up the project with React and TypeScript. So, let's do that now.

9. Install React and TypeScript

Install React and TypeScript as dev-dependencies:

```
npm i -D react@19.0.0 typescript@5.7.3 @types/react@19.0.8
```

The command above tells NPM to install `react`, `typescript`, and `@types/react` as your package's dev-dependencies.

Note

- The `react` package is a library containing React's core functionality for creating React components.
- The `typescript` package is a library containing TypeScript's core functionality for type-checking and compiling your React code to JavaScript.
- The `@types/react` package is a library containing React's type definitions, which TypeScript will use to type-check your React code.
- `i` is the shorthand notation for `install`.
- The `-D` flag is the shorthand notation for `--save-dev`.

We installed `typescript` and `@types/react` as dev-dependencies because you only need them for your package's development and testing purposes. Users do not need them in production.

But why also install React as a dev-dependency since users need it in production? Let's discuss the reason.

Why Install React as a Package's Dev-Dependency?

Installing React as a dev-dependency rather than a dependency package prevents making it mandatory for users to download React while installing your package.

React requires having only one copy of the `react` package in a project.

Therefore, specifying React as a dev-dependency prevents making it mandatory for users to download React while installing your package. If their project already has a React copy, they wouldn't need to install an additional copy.

A caveat of the dev-dependency installation is that your package will only work for projects having the React package preinstalled. However, there's a good workaround:

- Duplicate the `"react"` property of your `package.json`'s `"devDependencies"` field into the `"peerDependencies"` like so:

Figure 4. package.json: Add peerDependencies field (line 5-7)

```
1 {
2   "devDependencies": {
3     "react": "^19.0.0"
4   },
5   "peerDependencies": {
6     "react": "^19.0.0"
7   }
8 }
```

Specifying React as a peer-dependency tells package managers to check whether the app installing your package contains the listed `peerDependencies`. If so, the application has the dependencies your package needs to work.

But suppose the package manager could not find the `peerDependencies`. In that case, some NPM versions (like versions 7 and 8) will install them automatically. But other versions (like versions 3 to 6) will display a warning showing the dependencies your package needs to work, which the user must install manually.

dependencies vs. devDependencies vs. peerDependencies

- `"dependencies"` specify the packages a project depends on in production (when your app is running on a live server).
- The `"devDependencies"` field lists all the packages a project does not need in production—but requires for its local development and testing purposes.
- `"peerDependencies"` list all the packages a project expects its host application to have in its `node_modules` directory.

So, now that you've installed React and TypeScript, we can configure the TypeScript compiler.

10. Configure the TypeScript Compiler

Developers use a `tsconfig.json` file to specify a project's root directory and the options the compiler needs to compile the project's files.

TypeScript takes the directory where the `tsconfig.json` file resides as the project's root.

So, navigate to your package's root directory and create a `tsconfig.json` file:

```
1  touch tsconfig.json
```

Afterward, add the following configuration to the newly created file:

Figure 5. tsconfig.json

```
{
  "compilerOptions": {
    "jsx": "react",
    "module": "esnext",
    "moduleResolution": "node",
    "outDir": "dist/esm",
    "target": "es5",
    "allowSyntheticDefaultImports": true,
    "declaration": true,
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true,
    "isolatedModules": true,
    "noEmitOnError": true,
    "noImplicitReturns": true,
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "removeComments": true,
    "skipLibCheck": true,
    "strict": true
  }
}
```

The configuration above specifies how TypeScript should compile the project.

- `jsx` specifies how TypeScript should compile the JSX syntaxes in `.tsx files to plain JavaScript. The` “react”`value means TypeScript should change JSX syntaxes to`React.createElement‘ calls.
- `module` specifies the syntax TypeScript should use for importing and exporting modules in the compiled file.
- `moduleResolution` tells TypeScript to use NodeJS’ module resolution strategy to resolve (locate) the modules you import into your TypeScript files.
- `outDir` tells TypeScript to emit its ES Modules transpiled files into the `dist/esm` folder.
- `target` tells TypeScript the JavaScript version you want to compile your code into.
- `allowSyntheticDefaultImports` tells TypeScript to allow you to declare default imports like `import React from "react"` rather than `import * as React from "react"` when the module you are importing does not have a default export.

- `declaration` tells TypeScript to generate `d.ts` configuration files from your package's TypeScript and JavaScript files.
- `esModuleInterop` tells TypeScript to make its compiled code interoperable (compatible) between CommonJS and ES Modules codebase by creating namespace objects for all imports.
- `forceConsistentCasingInFileNames` tells TypeScript to throw an error when a program tries to reference a file with a casing different from the file's name. For instance, referencing a `codesweetly.ts` file with `./CodeSweetly.ts` will throw an error.
- `isolatedModules` tells TypeScript to warn you whenever you write any code that transpilers (such as Babel) cannot compile correctly.
- `noEmitOnError` specifies that TypeScript should not emit any compiled code if errors exist in the TypeScript project.
- `noImplicitReturns` tells TypeScript to issue an error if a function's code path does not return a value.
- `noUnusedLocals` tells TypeScript to issue an error if a codebase contains unused local variables.
- `noUnusedParameters` tells TypeScript to issue an error if a function has unused parameters.
- `removeComments` tells TypeScript to remove all comments when transpiling TypeScript files to JavaScript.
- `skipLibCheck` specifies that TypeScript should not type-check declaration (`.d.ts`) files.
- `strict` tells TypeScript to use "strict mode" to type-check your codebase.

Once you've configured the TypeScript compiler, proceed to the next chapter, where we will discuss writing your component's test case.

Testing React Component

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Install the Testing Tools

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Specify Jest as Your Project's Test Runner Tool

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Configure Jest to Test TypeScript Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Configure Jest to Compile TypeScript Files

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

5. Configure Jest to Read TypeScript Configuration Files

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

6. Configure Jest's Testing Environment

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

7. Create Your Code Files

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

8. Write Your Test Case

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

9. Develop Your React Component

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

10. Run the Test

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Commit Message Configuration

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Conventional Commits Messages Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Enforcing the Conventional Commits Format

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Setting up Husky

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Creating a Hook to Auto-Lint Commit Messages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Setting Up GitHub Action

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Create a GitHub Action Workflow File

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Define the GitHub Action Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Test the GitHub Action Workflow

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Styling React Components

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Create the Stylesheet

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Apply CSS Styles to Your Component

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Loading CSS Files to Jest's JSDOM

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Run the Test

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Compiling React to JavaScript

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

How to Compile ECMAScript and CommonJS Modules

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Distinguishing Source and Distribution Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Specify Items to Compile

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Defining Package's Entry Point

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Specifying Package's Declaration File

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Local Testing of Unpublished Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Link-Install Your Package Globally in Your System

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Create a NextJS Demo Website for Testing Your Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Install Your Package from Your System's Global Folder to the Test-App

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Use the Link-Installed Package in Your Test-App

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

5. Unlink Your Package from the Test-App

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

6. Unlink Your Package from the Global Folder

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Creating README

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Creating LICENSE

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Publishing Package to NPM

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Search Engine Optimization (SE0)

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Description

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Keywords

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Author

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

License

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Homepage

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Specify the Files You Want to Publish to NPM

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Confirm the Files NPM Will Publish

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Confirm That You Have Passing Tests

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

5. Compile Any Pending Code

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

6. Stage and Commit Any Recent Changes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

7. Push Your Local Git Directory to the Remote Repo

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

8. Sign In or Sign Up on the NPM Website

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

9. Log In to NPM via the Terminal

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

10. Confirm If Your Package's Name Is Available

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

11. Publish Your Package!

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Local Testing of the Published Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Install the Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Import the Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Run Your Local Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Production Testing of the Published Package

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Stage and Commit Your Changes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Configure a GitHub Remote Repository for Your NextJS Test-App

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Push Your Local Git Directory to the Remote Repo

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Sign In or Sign Up on the Vercel Website

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

5. Deploy Your Project to Vercel's Server

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

6. Test the Package on the Live Demo Website

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Updating Package's Versions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Example 1: Updating to a Patch Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Example 2: Updating to a Minor Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Example 3: Updating to a Major Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Automating Version Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

How to Release Your Package's Latest Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

How to Overwrite Release-It's Default Configurations

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

How Release-It Determines Your Package's Latest Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

How to Provide Release-It with a Recommended Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Automating Changelog Management

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Automating GitHub Releases

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

1. Get a GitHub Personal Access Token

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

2. Copy the Generated Token

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

3. Create a GitHub Environment Variable

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

4. Tell Release-It Your GitHub Token's Name

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

5. Make the GitHub Secret Available Locally in Your Project

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

6. Prevent Git from Monitoring the Environment File

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

7. Install the Plugin for Loading the .env File

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

8. Load the .env File While Releasing Your Project's Latest Version

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

9. Stage and Commit Your Changes

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Epilogue

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

What Next?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.

Reviews

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/creating-npm-package-with-react-typescript.