LARGE SCALE APPS WITH REACT AND TYPESCRIPT

DAMIANO FUSCO

Large Scale Apps with React and TypeScript

Build Large and Scalable front-ends that leverage component isolation, a centralized state manager, internationalization, localization, Custom Component Libraries, API-client code that easily can switch between mocked data and live data and more.

Damiano Fusco

This book is for sale at http://leanpub.com/react-typescript

This version was published on 2023-01-30

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

LARGE SCALE APPS WITH REACT AND TYPESCRIPT

This book is a guide for developers looking to build large-scale front-end applications with React and TypeScript. With the growth of the web and mobile app development, there is an increasing demand for robust, scalable, and maintainable front-end solutions. This book provides a comprehensive approach to building large scale code bases that use React and TypeScript.

The book covers key concepts and best practices like:

- **Building front-ends** that can grow to a large code base that is organized and easy to expand and maintain.
- **Development of UI components in isolation** using an API client that can easily serve live data or mocked data.
- **Centralized State Manager** organized into domain/area modules, providing a unified and consistent way to manage the application state.
- **Internationalization and Localization** for language translation and number/dates formatting according to a specific culture, making it easier to reach a global audience.
- **TypeScript** type-checking at development time to decrease run-time bugs or errors, reducing the risk of costly bugs and enhancing the overall quality of the code.
- **Directory structure, file, and code naming conventions**, ensuring a consistent and organized project structure for both developers and future maintainers.
- **Hooks and Compositional Patterns**, providing a flexible and reusable way to implement functionality in components.
- **Components Libraries**, allowing developers to build a library of reusable components, reducing development time and increasing code quality.
- **Unit tests** for models and components, ensuring code quality and reducing the risk of regressions.

The book is designed for developers with intermediate to advanced React and TypeScript skills who are looking to take their applications to the next level. Whether you are building a new large-scale app or optimizing an existing one, this book will provide you with the tools and knowledge to achieve your goals. Throughout the book, practical examples and real-world scenarios are used to illustrate key concepts and best practices, providing you with a solid understanding of how to build large scale apps with React and TypeScript.

Preface

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

Goal

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

Audience

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

Text Conventions

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

Thanks

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

About me

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

Prerequisites

This book assumes that you are familiar with the **terminal** (**command prompt** on Windows), have already worked with the **Node.js** and **NPM** (**Node Package Manager**), know how to install packages, and are familiar with the **package.json** file.

It also assumes you are familiar with **JavaScript**, **HTML**, **CSS** and in particular with **HTML DOM** elements properties and events.

It will also help if you have some preliminary knowledge of **TypeScript**[1] as we won't get into details about the language itself or all of its features but mostly illustrate how to enforce type checking at development time with it.

You will need a text editor like **VS Code** or **Sublime Text**, better if you have extensions/plugins installed that can help specifically for React/JSX code. For VS Code for example, you could use extensions like **react-vscode-extension-pack**[2] (just search for it within the VS code extensions tab).

[1] https://www.typescriptlang.org

[2] jawandarajbir.react-vscode-extension-pack

Companion Code

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

Chapter 1 - Setting Up The Project

IMPORTANT: *This chapter assumes that you already have installed a recent version of **Node.js** on your computer. If you do not have it yet, you can download it here: https://nodejs.org/en/download/*

There are many different ways to create a React app. Here we'll be leveraging TypeScript and therefore will need to setup an project with a build/transpile process that will let us make changes and verify them in real time. You could manually create this project, install all the npm packages required, create each individual file. However, it is just much easier to do this by leveraging **vite**[3]

However, since **vite** is still relatively new at the time of this book writing, it might be harder for you to find help as your code-base grows larger, so this is something to keep in mind.

I added a bonus chapter at the end of the boook with instructions on how to setup the project using **create-react-app**[4] instead of vite if you prefer that. Keep in mind that the use of environment variables (which will cover in chapter 7 and 9), will differ.

Create Project Wizard

To set up the project, use the terminal and execute the following node.js command:

```
npm init vite@latest
```

If you do not have already installed the package `create-vite@latest`[5] it will prompt you to install it. In this case, type **y** and then **enter** to proceed:

```
Need to install the following packages:
  create-vite@latest
Ok to proceed? (y)
```

The create-vite wizard will start and will ask you the name of the project. The default is vite-project, so change this to **my-react-project** and hit enter:

[3] https://vitejs.dev

[4] https://github.com/facebook/create-react-app

[5] https://www.npmjs.com/package/create-vite

```
? Project name: › my-react-project
```

The second step will ask to select a framework. Use the keyboard arrows to scroll down the list and stop at **React**, then hit enter:

```
? Select a framework: › - Use arrow-keys. Return to submit.
    Vanilla
    Vue
□   React
    Preact
    Lit
    Svelte
```

The third step will asking which “variant” you want o use. Scroll down to **TypeScript** and hit enter:

```
? Select a variant: › - Use arrow-keys. Return to submit.
    JavaScript
□   TypeScript
```

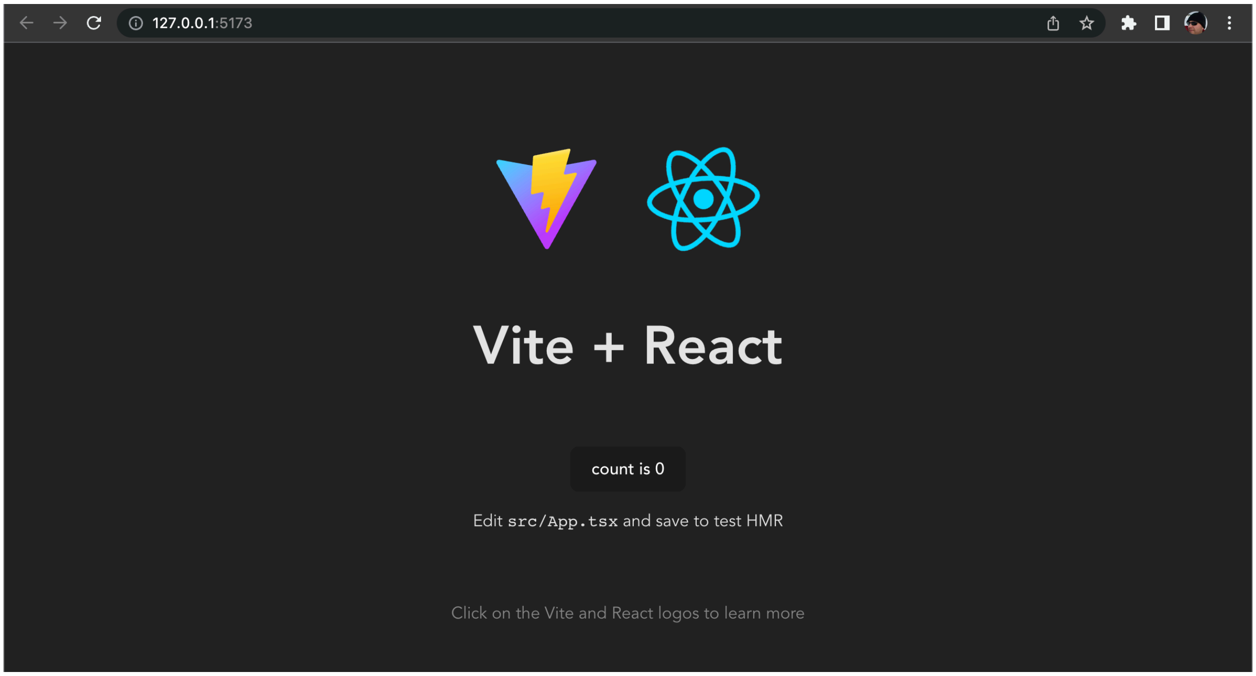
This will create a folder called **my-react-project** which is also the name of our project. At the end it should display a message similar to this:

```
Scaffolding project in /Volumes/projects/my-react-project...

Done. Now run:

  cd my-react-project
  npm install
  npm run dev
```

The first command will navigate to the current sub-directory called **my-react-project**, the second one will install all the npm dependencies, the third one will serve the app locally. You’ll see a message similar to this displayed:

```
VITE v3.1.4  ready in 303 ms

  □  Local:   http://127.0.0.1:5173/
  □  Network: use --host to expose
```

From the web browser, navigate to the http://localhost:5173 address and you'll see application home page rendered:



127.0.0.1:5173

Vite + React

count is 0

Edit `src/App.tsx` and save to test HMR

Click on the Vite and React logos to learn more

The **my-react-project** has been created with one main view called **App.tsx**.

Chapter 1 Recap

What We Learned

How to create the basic plumbing for a **React** app using the **vite** and `create-vite@latest`

- How to serve the app locally through the command **npm run dev**

Observations

- The app has been created with only preliminary code
- The app does not do much yet, has only the main App view with some static html in it

Based on these observations, there are a few improvements that will be making into the next chapter:

Improvements

- Expand our app functionality by creating our first component

Chapter 2 - Your First Component

The Items List

Let's now pretend we have been giving requirements for our app to have a component that displays a list of "**items**". We will keep this simple initially and as we move towards more advanced chapter expand on it to show how we can better structure our application to support:

- Quick prototyping and development using mocked data
- Component Organization
- Unit Testing
- State Management
- Internationalization support so we can render our user interface using different languages
- Localization for number and date formatting for different cultures

ItemsList Component Requirements

Your initial version of the **ItemsList** component, will have to implement the following requirements (later, in more advanced chapters, we will expand on these as we get into more advanced topics):

- The component will display a list of items
- An **item** will have 3 properties:

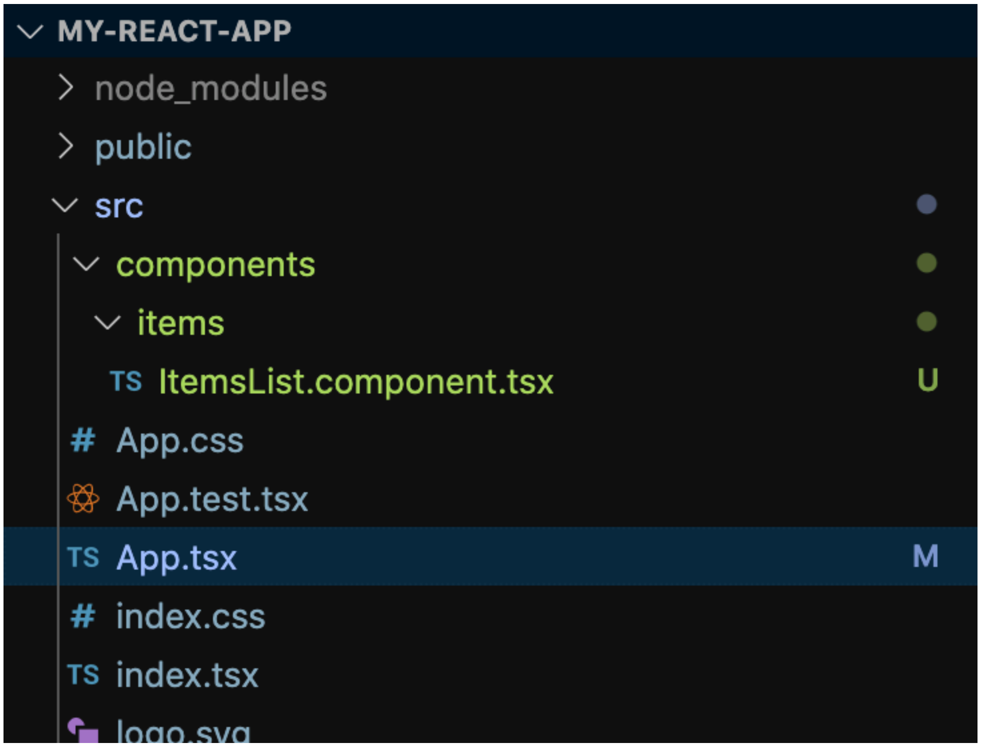
* **id**
* **name**
* **selected**

- The item **name** will be displayed to the user
- The user should be able to **select/deselect** one or more item
- An **icon** will be shown next to the name to indicate if the item is selected

ItemsList Component Code

Within the **src/components** directory, create a sub-directory called **items**. Within this folder add a new file called **ItemsList.component.tsx**[6]

Your directory structure will now look like this:



```
MY-REACT-APP
  node_modules
  public
  src                              ●
    components                     ●
      items                        ●
        TS ItemsList.component.tsx U
    # App.css
    App.test.tsx
    TS App.tsx                     M
    # index.css
    TS index.tsx
    logo.svg
```

Within the **ItemsList.component.tsx** file, paste the following code:

```
// example using const of type React.FC:
import React from 'react'

export const ItemsListComponent: React.FC<{
    items: any[]
  }> = (props) => {

  return (
    <div>
      <h3>Items:</h3>
      <ul>
```

[6] *We are following a file naming convention where higher level components' names are pascal-case and follow this format [ComponentName].component.tsx - Reference: Naming Conventions section at the end of this book*

```
        {
          props.items.map((item, index) => <li key={index}>{item.name}</li>
          )
        }
      </ul>
    </div>
  )
}
```

A few things to notice here. There are different ways to create a React component. You could just return a **function**, a **class**, or like in the example above, a **const** of type **React.FC**[7]

NOTE: Deciding whether to use a function or a class might be a matter of personal preference, or just abiding the coding standard you have defined with your team in your organization. If you google `React.FC vs class` you'll get several blogs/articles where it seems the majority of developers prefer pure function or classes, rather than React.FC. Going forward, I'll try to use classes or functions throughout the book and avoid React.FC (but there might be cases where I use any of the three)

When using a const of type React.FC, you will need to return the component html wrapped with parentheses. If using a class, you will need to implement a **render** function that returns the html.

For example, using the **class** syntax, the above component can be re-written as:

```
// example using class extending component
import React from 'react'

export class ItemsListComponent extends React.Component<{
  items: any[]
}> {
  constructor(props: {
    items: any[]
  }) {
    super(props)
  }

  render(): React.ReactNode {
    const { items } = this.props

    return <div>
        <h3>Items:</h3>
```

[7] *React.FC*

```
        <ul>
          {
            items.map((item: any, index: number) => <li key={index}>{item.name}</li>)
          }
        </ul>
      </div>
  }
}
```

Please go ahead and replace the code with the above using the class syntax. Then save and verify everything still renders as before without error in the browser console.

Here is what we are doing int he component code above:

For our html, we are returning a **<div>** element containing:

- a **<h3>** element with hard-coded text just saying ”**Items:**”
- a **<ul>** element with some code that will render all our **items** as **<li>** elements.

We use the JavaScript Array native **map** method to loop through the **Items** array and return an **<li>** element for each item in the array. The **<li>** element will display the item name in the browser. Note how we have to also specify the **key** attribute which is required to be unique within a list rendered by React. Here we leverage the fact the the **map** method returns the **index** of the item in the array as the second argument to our handler function (index). The index is good enough for now to use for the key attribute.

Note that with map you can either inline the return expression, thus not needing the keyword return:

```
items.map((item, index) => <li key={index}>{item.name}</li>)
```

Or you could use {} (curly braces) for the function body, and use the return keyword in this case:

```
items.map((item, index) => {
  return <li key={index}>{item.name}</li>
})
```

Which syntax you use is up to your preference. However, remember that in a team, especialy in a large organization, there will be coding standards that will dictacte how you consistently write code. You should always abide the standard that you and your team have agreeed upon.

Note also that we declared the **items** property as an array of **any**[8] for now (later we'll replace

[8] *With 'any', TypeScript does not enforce type-checking on a property or variable. However, this is considered a bad practice as we lose the main benefit of TypeScript. There might be exceptions to this rule when using older 3rd party packages/libraries/plugins that do not offer type definitions. However, even in those cases it would be strongly recommended to provide interfaces and types so that you can still avoid using 'any'.*

any with an interface we'll create):

```
...
  {
    items: any[] // avoid using "any", in later chapters we'll replace with a TS int\
erface
  }
...
```

Main App View

Open the **src/App.tsx** file. Replace the entire existing code with this:

```
//file: src/App.tsx

// component:
function App() {
  return (
    <div>
      ...
    </div>
  );
}

export default App
```

Let's start by adding at the top an import to reference our **ItemsList.component.tsx**:

```
//file: src/App.tsx

// import reference to your ItemsList component:
import { ItemsListComponent } from './components/items/ItemsList.component'
...
```

For now, quickly mock some data for our list of **items** that we will feed to our **ItemsListComponent**. For this we instantiate a local variable called **items** and initialize it with some hard-coded data[9].

We do this before the App function declaration:

[9] *Note: using hard-coded data is a bad practice and here we are only doing it to first illustrate how things flow, and later in the next chapters will remove in favor of best practices and patterns (see Chapter 5)*

```
//file: src/App.tsx

// import reference to your ItemsList component:
import { ItemsListComponent } from './components/items/ItemsList.component'

// mock data:
const items: any[] = [{
           id: 1,
           name: 'Item 1'
         }, {
           id: 2,
           name: 'Item 2'
         }, {
           id: 3,
           name: 'Item 3'
         }]

// component:
function App() {
...
```

Finally, we modify the content inside the <div> (within the return statement). Let just add our **ItemsListComponent** and pass the hard-coded **items** data to it through its property **items**.

The complete code within the **App.tsx** file should now look like this:

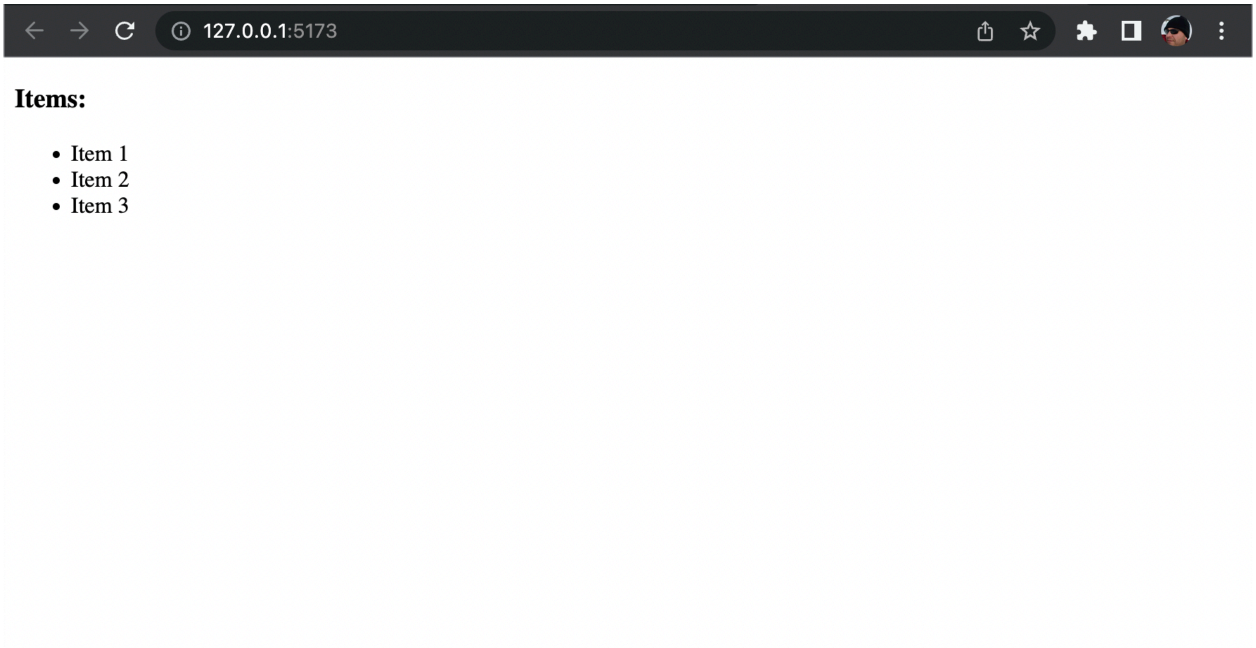
```
//file: src/App.tsx

// import reference to your ItemsList component:
import { ItemsListComponent } from './components/items/ItemsList.component'

// mock data:
const items: any[] = [{
  id: 1,
  name: 'Item 1'
}, {
  id: 2,
  name: 'Item 2'
}, {
  id: 3,
  name: 'Item 3'
}]
```

```
// component:
function App() {
  return (
    <div>
      <ItemsListComponent items={items}/>
    </div>
  );
}

export default App
```

Update src/main.tsx by removing or commenting the imported index.css file:

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
// import './index.css' // <-- comment out or remove this line

ReactDOM.createRoot(document.getElementById('root') as HTMLElement).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
)
```

Save the file. The web browser will refresh and display our preliminary items list being rendered more or less like this:



Items:

- Item 1
- Item 2
- Item 3

Chapter 2 Recap

What We Learned

- How to create a basic component that displays a list of items
- How to consume that component from another component or view

Observations

- The **items** property within the **ItemsList.component.tsx** is declared as an array of type **any**
- The **App.tsx** view contains hard-coded data (items) which is also declared as an array of **any**
- This means we are not leveraging strong-type checking at development time using TypeScript interfaces/models/types

Based on these observations, there are a few improvements that we will make in the next chapters:

Improvements

- Create a TypeScript interface called **ItemInterface** for enforcing type checking at development time for our **items** data
- Update our code so it uses the new **ItemInterface** interface

Chapter 3 - Data Models and Interfaces

In this chapter, we will delve into the power of **TypeScript** by leveraging its strong-type checking capabilities through the use of interfaces. One of the main challenges with pure JavaScript is its loosely typed nature, which can lead to unexpected behavior and bugs at run-time. This is due to the lack of checks on the type or properties of values or objects being passed around in the code. TypeScript solves this problem by providing developers with the ability to enforce strict type checking at development time through the use of interfaces, types, classes, and more.

By incorporating TypeScript into our project, we'll be able to catch potential issues and bugs before they reach the production environment, saving us time and resources in debugging and fixing. Moreover, strong-typing also improves the readability and maintainability of our code, making it easier for developers to understand the purpose and usage of values and objects in the codebase. In this chapter, we'll explore how to use interfaces and types to implement strong-typing in our project, and how it can help us ensure the reliability and quality of our code.

Models Directory

To lay the foundation for building large-scale applications, we will start by creating a new sub-directory under **src** called **models**. The organization of files and directories plays a critical role in the success of large-scale code bases. As such, it's essential to establish a consistent naming **convention** and directory structure from the outset. This will help ensure that the code remains organized, easy to understand, and maintainable as the application grows and the number of source files and directories increases.

You and your team are free to determine the standards that work best for you, but it's crucial to establish a set of conventions and stick to them. This will save you a significant amount of time and effort in the long run and prevent confusion and headaches as the application grows and evolves.

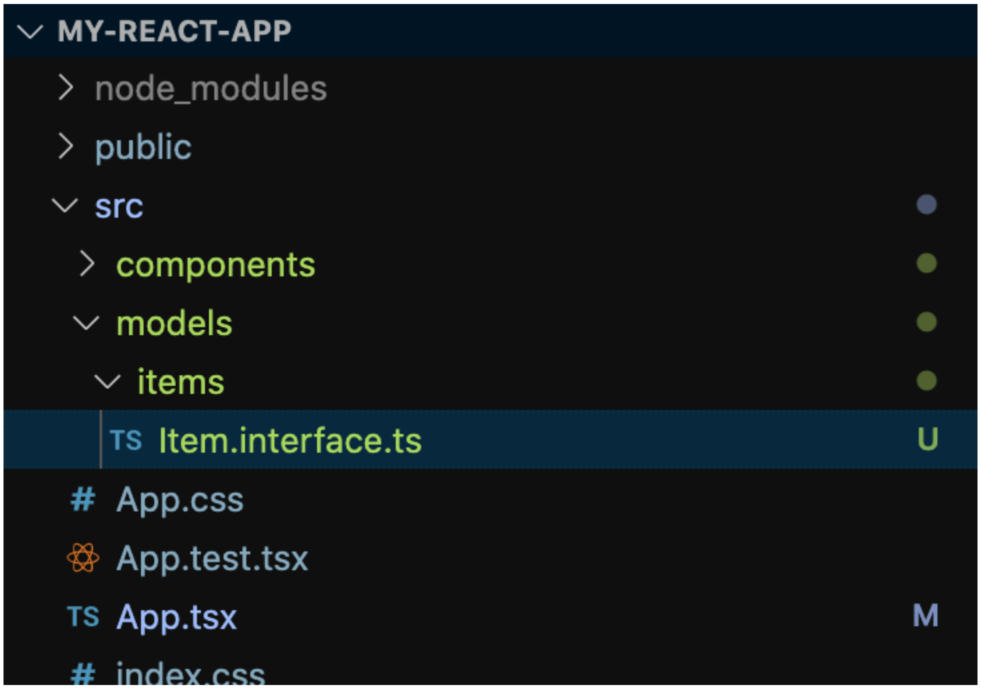
Interface ItemInterface

To create the interface for our items, we will create a new directory called **src/models/items** and add a TypeScript file named **Item.interface.ts**.

It's worth noting that there are different naming conventions for TypeScript interfaces, with some preferring to use a suffix like **Interface**, while others use a prefix like **I**. In this book, we will follow the suffix convention, using **Item.interface.ts** as the file name. However, you are free to choose your preferred naming convention or align with your organization's coding standards.

It's important to keep each interface in its own file, as this makes it easier to maintain and manage. For more information on naming conventions, please refer to the Naming Conventions section at the end of this book.

Your directory structure should now look similar to this:



```
MY-REACT-APP
  node_modules
  public
  src
    components
    models
      items
        TS Item.interface.ts        U
    # App.css
    App.test.tsx
    TS App.tsx                      M
    # index.css
```

Let's write an interface that represents one item that will be rendered in our **Item** component. Our interface will have three properties:

- **id**: this is a unique number for each item in the list
- **name**: is a string containing the name of the item
- **selected**: is a boolean value that shows if the user has selected the item

The code for your interface should look like this:

```
// file: src/models/items/Item.interface.ts

export interface ItemInterface {
  id: number
  name: string
  selected: boolean
}
```

For now, that is all we need. Since this will only represent a piece of data, we do not need to implement a class.

NOTE: In this case our ItemInterface only holds fields, but no methods. You can think of this more like the type struct in language like C or C#. Unfortunately TypeScript does not have an explicit struct type[10] *and their guidance is to use interfaces for this.*

ItemsList Component

Now that we have our interface, we can finally leverage TypeScript type checking ability by changing our items property on the items component from **any[]** to **ItemInterface[]**. First, import a reference for **ItemInterface**:

```
// file: src/components/items/ItemsList.component.tsx

import React from 'react'
// import reference to our interface
import { ItemInterface } from '../../models/items/Item.interface'
...
```

Then modify our items property declaration from any[] to ItemInterface[]:

[10] *There have been suggestions presented, but I do not think they will ever add a struct type. See the TypeScript team answers here:* https://github.com/microsoft/TypeScript/issues/22101

```
// file: src/components/items/ItemsList.component.tsx

// example using const of type React.FC:
import React from 'react'
// import reference to our interface
import { ItemInterface } from '../../models/items/Item.interface'

export const ItemsListComponent: React.FC<{
    items: ItemInterface[] // replace any[] with ItemInterface[]
  }> = (props) => {

...
```

The complete update code should look like this:

```
// file: src/components/items/ItemsList.component.tsx

// example using const of type React.FC:
import React from 'react'
// import reference to our interface
import { ItemInterface } from '../../models/items/Item.interface'

export const ItemsListComponent: React.FC<{
    items: ItemInterface[] // replace any[] with ItemInterface[]
  }> = (props) => {

  return (
    <div>
      <h3>Items:</h3>
      <ul>
        {
          props.items.map((item, index) => <li key={index}>{item.name}</li>
          )
        }
      </ul>
    </div>
  )
}
```

Or if you went with the class syntax:

```
// file: src/components/items/ItemsList.with-class-syntax.tsx

// example using class extending component
import React from 'react'
// import reference to our interface
import { ItemInterface } from '../../models/items/Item.interface'

export class ItemsListComponent extends React.Component<{
  items: ItemInterface[] // replace any[] with ItemInterface[]
}> {
  constructor(props: {
    items: ItemInterface[] // replace any[] with ItemInterface[]
  }) {
    super(props)
  }

  render(): React.ReactNode {
    const { items } = this.props

    return <div>
        <h3>Items:</h3>
        <ul>
          {
            items.map((item: any, index: number) => <li key={index}>{item.name}</li>)
          }
        </ul>
      </div>
  }
}
```

Make sure the terminal does not display any error, and that the web browser refreshed and no error are displayed in the browser console.

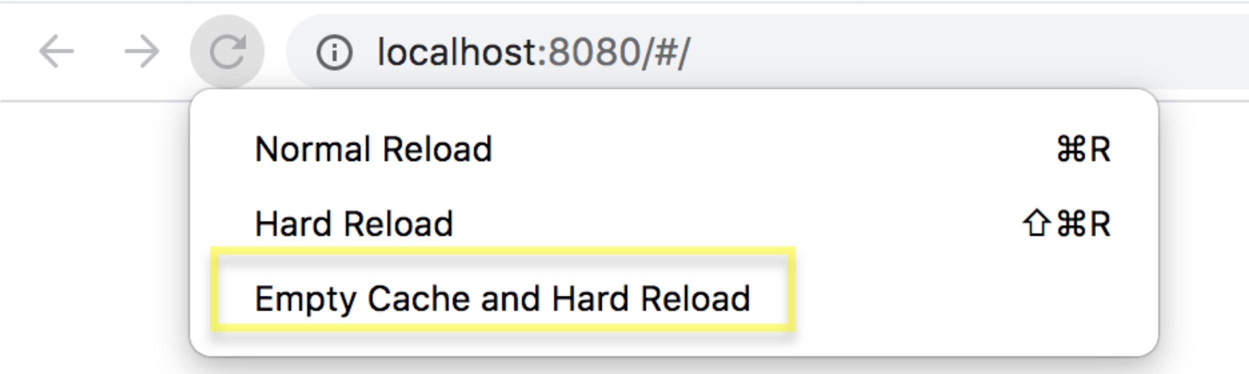
App View

We should also update the **App.tsx** code so it uses the **ItemInterface** interface for the locally private property also called **items**.

Please note, that as soon as you change the items property from any[] to ItemInterface[] it will complain that each item does not correctly implement the interface. This is because we did not initially include the selected property required by the interface. This is one of the powerful *things of using TypeScript correctly. It will help catch errors like this at development time rather than run time, increase the code quality and make it less prone to bugs. So make sure each item has now also a selected field with a default of false.*

```
// file: src/App.tsx

import './App.css'
// import reference to our interface
import { ItemInterface } from './models/items/Item.interface'
// import reference to your ItemsList component:
import { ItemsListComponent } from './components/items/ItemsList.component'

// mock data:
const items: ItemInterface[] = [{ // change any[] to ItemInterface[]
  id: 1,
  name: 'Item 1',
  selected: false // add selected: false to each item
}, {
  id: 2,
  name: 'Item 2',
  selected: false
}, {
  id: 3,
  name: 'Item 3',
  selected: false
}]
...
```

Again, make sure the terminal does not display any errors, and that the web browser refreshed and no error are displayed in the browser console. As you make changes is also a good idea occasionally to do an **Empty Cache and Hard Reload** by right clicking on the Chrome refresh icon and selecting the last option:


localhost:8080/#/
Normal Reload ⌘R
Hard Reload ⇧⌘R
Empty Cache and Hard Reload

Chapter 3 Recap

What We Learned

- It's important to follow files and directories naming convention and structure convention
- How to leverage **TypeScript** interfaces and avoid using **any** so that strong-type checking is enforced at development time and avoiding potential runtime errors or hidden bugs

Observations

- The **App.tsx** contains a local variable that holds hard-coded mocked data that enabled us to prototype our component quickly
- **ItemsList.component.tsx** just displays the list of items, but the user has still no ability to click on them to change their **selected** property

Based on these observations, there are a few improvements that we will make into the next chapter:

Improvements

- Update our component so that when a user clicks on an item displayed on the page, the item selected property will toggle from false to true (and vice versa)

Chapter 4 - Adding Events To the Items Component

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

ItemsList Component

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

Chapter 4 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 5 - Intro to Unit Testing While Refactoring a Bit

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

ItemComponent

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

Install npm dependencies for unit tests

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

Configuration

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

tsconfig.json file

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

vite.config.js files

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

package.json

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

ItemComponent Unit Tests

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

Item.rendering.test.tsx

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

Item.behavior.test.tsx

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

ItemsList component

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

Chapter 5 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 6 - Introducing State Management

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

Store Interfaces

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

Items Store Interfaces

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

ItemsState.interface.ts

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

Root Store Interfaces

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

RootStore.interface.ts

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

Store Implementation

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

Items Store instance

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

Items.slice.ts

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

Items.store.ts

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

Root Store Instance

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

Root.store.ts

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

App.tsx

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

Items.view.tsx

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

Back to the App.tsx file

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

Web Browser

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

ItemsList.component.tsx

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

Back to the Web Browser

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

Loader Component

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

Chapter 6 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 7 - Api Client

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

API Client Overview

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

Domains

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

The Main ApiClient

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

Items domain Api Client

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

ItemsApiClientOptions.interface.ts

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

ItemsApiClient.interface.ts

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

ItemsApiClient.model.ts

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

index.ts (barrel file)

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

Mock and Live Api Clients

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

Mock Api Client

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

Items domain mock API instance

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

Mock API instance

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

Live Api Client

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

Items domain live API instance

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

Live API instance

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

Environment Variables

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

Api Client Provider

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

Store Instance updates

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

Alternatives

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

Alternative: using json-server

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

Chapter 7 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 8 - Enhance the Api Client

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

HttpClient Interfaces and Models

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

Constants.ts

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

HttpRequestParams.interface.ts

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

UrlUtils.ts

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

HttpClient.interface.ts

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

HttpClient.axios.ts

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

HttpClient.fetch.ts

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

http-client/models/index.ts (barrel file)

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

http-client/index.ts (client factory)

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

UrlUtils Unit Tests

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

HttpClient: Unit Tests

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

HttpClientAxios tests

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

Testing a successful ”get” response

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

Testing an unsuccessful ”get” response

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

Testing a successful ”post” response

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

HttpClientFetch tests

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

Testing “get” responses

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

ItemsApiClientModel Update

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

Chapter 8 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 9 - App Configuration

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

vite-env.d.ts updates (or env.d.ts)

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

.env files updates

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

Config Interface

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

Config files

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

mock.json

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

jsonserver.json

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

localapis.json

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

beta.json

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

production.json

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

tsconfig.json updates

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

Config files map

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

Config provider

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

File utils.ts

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

File index.ts (config provider)

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

Unit Tests

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

Unit Tests against configsMap

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

Unit Tests against Config instances by environment

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

Tests config.mock.test.ts

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

HttpClient code updates

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

file src/http-client/index.ts

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

Api Client code updates

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

file src/api-client/index.ts

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

file src/api-client/mock/items/index.ts

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

file src/api-client/live/items/index.ts

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

Chapter 9 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 10 - Localization and Internationalization - Language Localization

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

Plugins: i18next, react-i18next

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

Config updates

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

ConfigInterface

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

file mock.json

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

Translation JSON data

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

API Client updates

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

file LocalizationApiClient.interface.ts

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

file LocalizationApiClientOptions.interface.ts

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

file LocalizationApiClient.model.ts

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

file src/api-client/models/localization/index.ts

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

Updates to ApiClient.interface.ts

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

Updates to the main models barrel file (api-client/models/index.ts)

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

Updates to ApiClient instances

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

localization mock instance (api-client/mock/localization/index.ts)

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

mock instance (api-client/mock/index.ts)

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

live instance (api-client/live/index.ts)

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

i18n initialization and useLocalization hook

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

file i18n.init.ts

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

file useLocalization.ts

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

file src/localization/index.ts (barrel file)

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

main.tsx or index.tsx updates

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

App.tsx updates

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

Browser

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

Test that our caching logic is working:

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

Important

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

Chapter 10 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 11 - Localization and Internationalization - Number and DateTime Formatters

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

Directory localization/formatters

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

File: useDateTimeFormatters.ts

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

File: useNumberFormatters.ts

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

File: index.ts

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

Component DebugFormatters.component.tsx

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

Updates to App.tsx

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

Chapter 11 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 12 - Adding Tailwind CSS

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

Chapter 12 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 13 - Intro to Primitives

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

Atomic Design and Similar Approaches

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

Conventions

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

General Strategies

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

Text Elements

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

Primitives View

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

Chapter 13 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 14 - More Primitives

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

Button Elements

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

Primitives View - update

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

Toggle/Checkbox Elements

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

Primitives View - one more update

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

Chapter 14 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 15 - A Primitive Modal

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

Icon: ElIconAlert

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

Interface ModalProps

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

File ElModal.ts (note: not .tsx)

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

File useModal.ts

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

Updates to Primitives.view.tsx

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

Browser

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

Chapter 15 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 16 - Higher-level components

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

Item Component - updates

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

ItemsList Component - updates

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

Summary

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

Chapter 16 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 17 - Creating Component Libraries

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

Create my-component-library

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

Remove obsolete files

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

Add main entry index.ts file

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

Update vite.config.ts

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

Update package.json commands

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

Create Counter.tsx component

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

Create SampleComp.tsx component

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

Add components/index.ts barrel file

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

Build our library

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

Consuming our library

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

Chapter 17 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 18 - Creating a JavaScript library

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

Create my-js-helpers

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

Remove obsolete files

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

Add main entry index.ts file

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

Update vite.config.ts

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

Update package.json commands

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

random-id

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

random-id unit tests

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

Build the library

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

Consuming the my-js-helpers library

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

Chapter 18 Recap

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

What We Learned

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

Observations

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

Improvements

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

Chapter 19 - Publish a library as a NPM package

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

Create an NPM user account

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

Create an Organization under your NPM profile

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

Update my-js-helpers package.json

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

Publishing the library

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

Consuming your NPM package

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

Chapter 19 Recap

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

What We Learned

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

Observations

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

Improvements

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

(More Chapters Coming Soon)

Bonus Chapter - using create-react-app

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

Env Variables

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

Bonus Chapter - Vitest

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

Remove Jest dependencies and setup files

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

tsconfig.json updates

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

package.json updates

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

Add Vitest

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

tsconfig.json updates

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

package.json updates

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

vite.config.js updates

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

test-utils code

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

Unit Tests updates

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

Item component unit tests updates

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

Item.rendering.test.tsx updates

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

Item.behavior.test.tsx updates

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

HttpClient unit tests updates

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

HttpClient.request.get.test.ts updates

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

HttpClient.request.post.test.ts updates

Naming Conventions

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

Coding Standards

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

Directory/File Naming and Structure

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

Directory Names

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

File Names

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

Interface File Names

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

Components File Names

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

Views/Pages File Names

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

Unit Tests file names

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

Directory src

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

Directory tests/unit

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

Resources

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

Websites

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

Blogs

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

Atomic Design

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

Books

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