

# LARGE SCALE APPS WITH **SVELTE** AND TYPESCRIPT



DAMIANO FUSCO

# Large Scale Apps with Svelte 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/svelte-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.

© 2022 - 2023 Damiano Fusco

# Contents

<b>LARGE SCALE APPS WITH SVELTE AND TYPESCRIPT . . . . .</b>	<b>1</b>
<b>Preface . . . . .</b>	<b>3</b>
Goal . . . . .	3
Audience . . . . .	3
Text Conventions . . . . .	3
Thanks . . . . .	3
About me . . . . .	3
<b>Prerequisites . . . . .</b>	<b>4</b>
<b>Companion Code . . . . .</b>	<b>5</b>
<b>Chapter 1 - Setting Up The Project . . . . .</b>	<b>6</b>
Create Project Wizard . . . . .	6
Chapter 1 Recap . . . . .	9
<b>Chapter 2 - Your First Component . . . . .</b>	<b>10</b>
The Items List . . . . .	10
ItemsList Component Requirements . . . . .	10
ItemsList Component Code . . . . .	11
App.svelte . . . . .	12
Chapter 2 Recap . . . . .	16
<b>Chapter 3 - Data Model Interfaces . . . . .</b>	<b>17</b>
Models Directory . . . . .	17
Interface ItemInterface . . . . .	17
ItemsList Component . . . . .	19
App.svelte . . . . .	20
Chapter 3 Recap . . . . .	22
<b>Chapter 4 - Adding Events To the Items Component . . . . .</b>	<b>23</b>
ItemsList Component . . . . .	23

## CONTENTS

Chapter 4 Recap . . . . .	23
<b>Chapter 5 - Intro to Unit Testing While Refactoring a Bit . . . . .</b>	<b>24</b>
ItemComponent . . . . .	24
Add unit tests support to our project . . . . .	24
ItemComponent Unit Tests . . . . .	25
ItemsList component . . . . .	25
Chapter 5 Recap . . . . .	25
<b>Chapter 6 - State Management . . . . .</b>	<b>27</b>
Store Interfaces . . . . .	27
Store Implementation . . . . .	28
Items.view.svelte . . . . .	28
App.svelte . . . . .	28
Web Browser . . . . .	29
ItemsList.component.svelte . . . . .	29
Back to the Web Browser . . . . .	29
Loader Component . . . . .	29
Chapter 6 Recap . . . . .	29
<b>Chapter 7 - Api Client . . . . .</b>	<b>31</b>
API Client Overview . . . . .	31
Domains . . . . .	31
The Main ApiClient . . . . .	31
Items domain Api Client . . . . .	31
Mock and Live Api Clients . . . . .	32
Environment Variables . . . . .	33
Api Client Provider . . . . .	33
Store Instance updates . . . . .	33
Alternatives . . . . .	33
Chapter 7 Recap . . . . .	34
<b>Chapter 8 - Enhance the Api Client . . . . .</b>	<b>35</b>
HttpClient Interfaces and Models . . . . .	35
UrlUtils Unit Tests . . . . .	36
HttpClient: Unit Tests . . . . .	36
ItemsApiClientModel Update . . . . .	37
Chapter 8 Recap . . . . .	37
<b>Chapter 9 - App Configuration . . . . .</b>	<b>39</b>
vite-env.d.ts updates (or env.d.ts) . . . . .	39

## CONTENTS

.env files updates . . . . .	39
Config Interface . . . . .	39
Config files . . . . .	39
tsconfig.json updates . . . . .	40
Config files map . . . . .	40
Config provider . . . . .	40
Unit Tests . . . . .	41
HttpClient code updates . . . . .	41
Api Client code updates . . . . .	42
Chapter 9 Recap . . . . .	42
<b>Chapter 10 - Localization and Internationalization - Language Localization . . . .</b>	<b>44</b>
Plugins: i18next, svelte-i18n . . . . .	44
Config updates . . . . .	44
Translation JSON data . . . . .	44
API Client updates . . . . .	45
Updates to ApiClient.interface.ts . . . . .	45
Updates to ApiClient instances . . . . .	46
i18n initialization and useLocalization hook . . . . .	46
App.svelte updates . . . . .	47
Browser . . . . .	47
Chapter 10 Recap . . . . .	47
<b>Chapter 11 - Localization and Internationalization - Number and DateTime</b>	
<b>Formatters . . . . .</b>	<b>48</b>
Directory localization/formatters . . . . .	48
Chapter 11 Recap . . . . .	49
<b>Chapter 12 - Adding Tailwind CSS . . . . .</b>	<b>50</b>
Chapter 12 Recap . . . . .	50
<b>Chapter 13 - Intro to Primitives . . . . .</b>	<b>51</b>
Atomic Design and Similar Approaches . . . . .	51
Conventions . . . . .	51
General Strategies . . . . .	51
Text Elements . . . . .	51
Primitives View . . . . .	51
Chapter 13 Recap . . . . .	52
<b>Chapter 14 - More Primitives . . . . .</b>	<b>53</b>
Button Elements . . . . .	53

## CONTENTS

Primitives View - update . . . . .	53
Toggle/Checkbox Elements . . . . .	53
Primitives View - one more update . . . . .	53
Chapter 14 Recap . . . . .	53
<b>Chapter 15 - A Primitive Modal . . . . .</b>	<b>55</b>
Icon: ElIconAlert . . . . .	55
Interface ModalProps . . . . .	55
File ElModal.svelte . . . . .	55
File useModal.ts . . . . .	55
Updates to Primitives.view.svelte . . . . .	55
Browser . . . . .	56
Chapter 15 Recap . . . . .	56
<b>Chapter 16 - Higher-level components . . . . .</b>	<b>57</b>
Item Component - updates . . . . .	57
ItemsList Component - updates . . . . .	57
Summary . . . . .	57
Chapter 16 Recap . . . . .	57
<b>Chapter 17 - Creating Component Libraries . . . . .</b>	<b>59</b>
Create my-component-library . . . . .	59
Chapter 17 Recap . . . . .	60
<b>Chapter 18 - Creating a JavaScript library . . . . .</b>	<b>62</b>
Create my-js-helpers . . . . .	62
Chapter 18 Recap . . . . .	63
<b>Chapter 19 - Publish a library as a NPM package . . . . .</b>	<b>64</b>
Create an NPM user account . . . . .	64
Create an Organization under your NPM profile . . . . .	64
Update my-js-helpers package.json . . . . .	64
Publishing the library . . . . .	64
Consuming your NPM package . . . . .	64
Chapter 19 Recap . . . . .	65
<b>(More Chapters Coming Soon) . . . . .</b>	<b>66</b>
<b>Naming Conventions . . . . .</b>	<b>67</b>
Coding Standards . . . . .	67
<b>Resources . . . . .</b>	<b>69</b>

## CONTENTS

Websites . . . . .	69
Tutorials . . . . .	69
Blogs . . . . .	69
Books . . . . .	69

# LARGE SCALE APPS WITH SVELTE AND TYPESCRIPT

This book is a guide for developers looking to build large-scale front-end applications with Svelte 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 Svelte 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 Svelte 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 Svelte and TypeScript.



Copyright © 2022 by Damiano Fusco (first published in January 2022)

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the author and publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the author and publisher, addressed “Attention: Permissions Coordinator,” at the email [me@damianofusco.com](mailto:me@damianofusco.com).

# Preface

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

## Goal

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

## Audience

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

## Text Conventions

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

## Thanks

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

## About me

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/svelte-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**<sup>1</sup> 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 Svelte code. For VS Code for example, you could use extensions like **Svelte for VS Code**<sup>2</sup> (just search for it within the VS code extensions tab).

---

<sup>1</sup><https://www.typescriptlang.org>

<sup>2</sup>[svelte.svelte-vscode](https://svelte.dev/docs#svelte-vscode)

# Companion Code

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/svelte-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 Svelte 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**<sup>3</sup>

## 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`<sup>4</sup> 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-svelte-project** and hit enter:

```
? Project name: > my-svelte-project
```

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

---

<sup>3</sup><https://vitejs.dev>

<sup>4</sup><https://www.npmjs.com/package/create-vite/v/2.6.6>

```
? 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 **svelte-ts** (this is for the version that uses TypeScript) and hit enter:

```
? Select a variant: > - Use arrow-keys. Return to submit.
  svelte
❑ svelte-ts
```

*NOTE: in most recent version of Site the options might be slightly different, make sure you select svelte-ts or TypeScript.*

This will create a folder called **my-svelte-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-svelte-project...
```

Done. Now run:

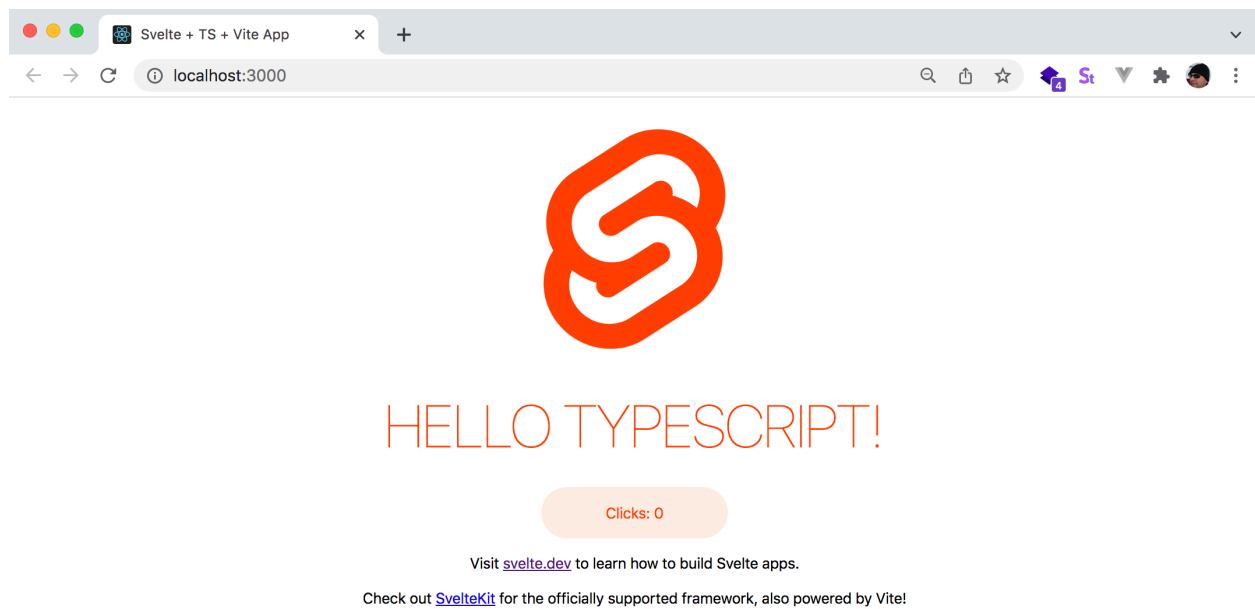
```
cd my-svelte-project
npm install
npm run dev
```

The first command will navigate to the current sub-directory called **my-svelte-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 v3.1.0 dev server running at:
```

```
> Local: http://localhost:3000/
> Network: use `--host` to expose
```

From the web browser, navigate to the <http://localhost:3000> address and you’ll see application home page rendered:



The **my-svelte-project** has been created with the main view called **App.svelte** and a component called **Counter.svelte**.

# Chapter 1 Recap

## What We Learned

How to create the basic plumbing for a **Svelte** 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 and a reference to a simple component called Counter

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:

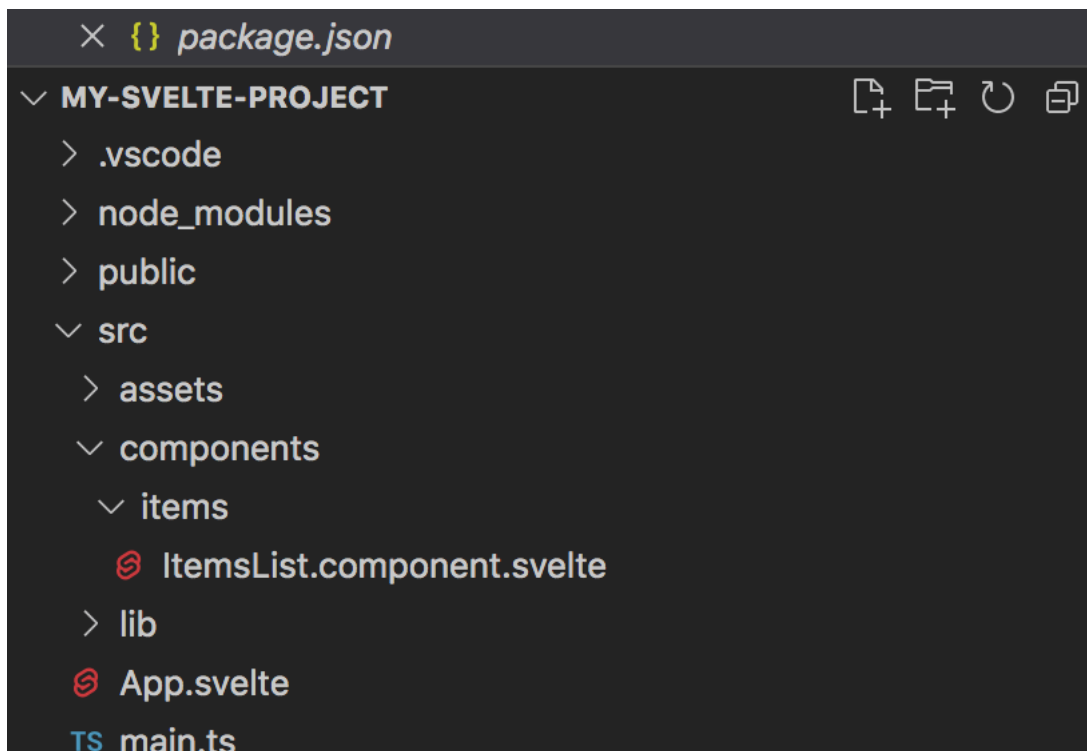
- \* **id**
- \* **name**
- \* **selected**

- 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.svelte`<sup>5</sup>

Your directory structure will now look like this:



Within the `ItemsList.component.svelte` file, paste the following code:

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

<script lang="ts">
  // expose a property called items with a default value of a blank array
  export let items: any[] = [] // explicetely using any[] as we'll replace this with\
  an interface in the next chapters
</script>

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

---

<sup>5</sup>We are following a file naming convention where higher level components' names are pascal-case and follow this format `[Component-Name].component.svelte` (Reference: Naming Conventions section at the end of this book)

```

    {#each items as item}
      <li>
        {item.name}
      </li>
    {/each}
  </ul>
</div>

```

A few things to notice here. First, we specify the **lang** attribute on the `<script>` element with the value **ts** so we can use **TypeScript**. We export a variable called **items**. This is how we create a component property in Svelte<sup>6</sup>. Note that **items** is just using the type **any**<sup>7</sup> for now (later we'll replace **any** with an interface we'll create).

For our html template, we added a `<h3>` element with hard-coded text just saying "Items:". Then a `<ul>` with a `#each` binding that will render all our **items** within `<li>` elements.

## App.svelte

Open the existing **App.svelte** file. Replace the existing code with this:

```

// file: src/App.svelte

<script lang="ts">
  // TODO ....
</script>
<main>
  <div class="home">
    ... TODO
  </div>
</main>

```

Within `<script>` the section, import a reference to the **ItemsListComponent**:

<sup>6</sup><https://svelte.dev/docs#component-format-script-1-export-creates-a-component-prop>

<sup>7</sup>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'.

```
// file: src/App.svelte
```

```
<script lang="ts">
  // import a reference to our ItemsList component
  import ItemsListComponent from './components/items/ItemsList.component.svelte'
  ...
```

For now, also mock some data for our list of items that we will feed to our **ItemsListComponent**. For this we instantiate a local const called **items** and initialize it with some hard-coded data<sup>8</sup>. Let's just add it after the import statement:

```
// file: src/App.svelte
```

```
<script lang="ts">
  // import a reference to our ItemsList component
  import ItemsListComponent from './components/items/ItemsList.component.svelte'

  // mock some data:
  const items: any[] = [{ // explicetely using any[] as we'll replace this with an i\
nterface in the next chapters
    id: 1,
    name: 'Item 1'
  }, {
    id: 2,
    name: 'Item 2'
  }, {
    id: 3,
    name: 'Item 3'
  }]
</script>
...
```

Finally, we add an **<ItemsListComponent>** element within the **<main>** markup. Insert it within the **<div class="home">** element. Add an attribute called **items** to our **<ItemsListComponent>**. This is how we pass properties to from parent to child components in Svelte<sup>9</sup>. The **items** data is fed into the component **items** property this way. The complete code within the **App.svelte** file should now look like this:

---

<sup>8</sup>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)

<sup>9</sup><https://svelte.dev/docs#template-syntax-attributes-and-props>

```
// file: src/App.svelte

<script lang="ts">
  // import a reference to our ItemsList component
  import ItemsListComponent from './components/items/ItemsList.component.svelte'

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

<main>
  <div class="home">
    <ItemsListComponent items={items}/>
  </div>
</main>
```

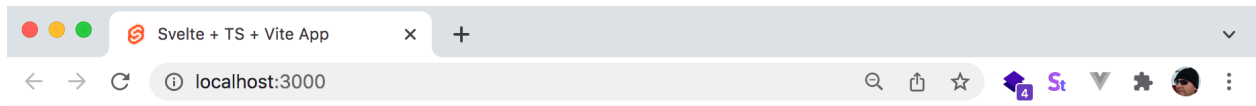
Comment out or remove the app.css import reference from the `src/main.ts` file:

```
// import './app.css' // <-- remove this line
import App from './App.svelte'

const app = new App({
  target: document.getElementById('app')
})

export default app
```

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

**My 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/view

### Observations

- The **items** property within the **ItemsList.component.svelte** is declared as an array of type **any**
- The **App.svelte** 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 Model 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

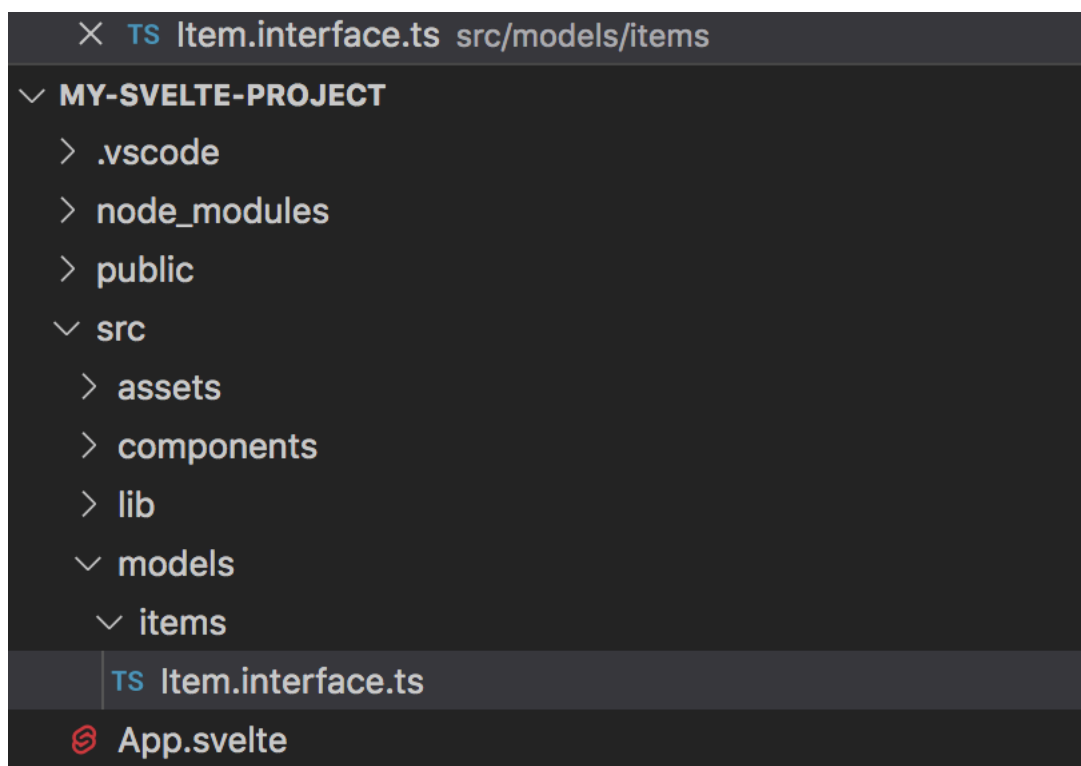
To create the interface for our items, we will create a new directory called **src/models/items** and add a TypeScript file named **Item.interface.ts**.



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:



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<sup>10</sup> 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` and modify our `items` property declaration from type `any[]` to type `ItemInterface[]`:

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

```
<script lang="ts">  
  // import a reference to our ItemInterface  
  import type { ItemInterface } from '../models/items/Item.interface'  
  // expose a property called items with a default value of a blank array  
  export let items: ItemInterface[] = [] // here replace any[] with ItemInterface[]  
</script>  
...
```

The complete update code should look like this:

---

<sup>10</sup>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.svelte

<script lang="ts">
  // import a reference to our ItemInterface
  import type { ItemInterface } from '../models/items/Item.interface'
  // expose a property called items with a default value of a blank array
  export let items: ItemInterface[] = []
</script>

<div>
  <h3>My Items:</h3>
  <ul>
    {#each items as item}
      <li>
        {item.name}
      </li>
    {/each}
  </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.svelte

We should also update the **App.svelte** 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.*

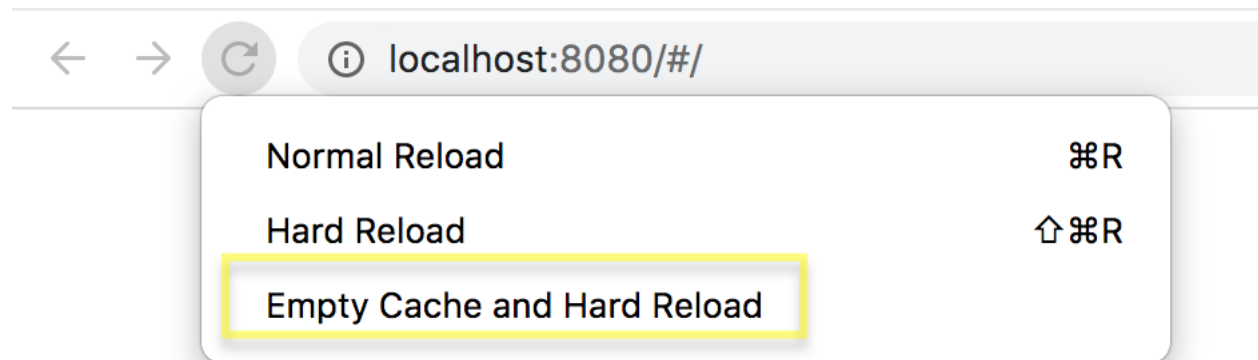
```
// file: src/App.svelte

<script lang="ts">
  // import a reference to our ItemInterface
  import type { ItemInterface } from './models/items/Item.interface'
  // import a reference to our ItemsList component
  import ItemsListComponent from './components/items/ItemsList.component.svelte'

  // mock some data:
  const items: ItemInterface[] = [{ // here 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
  }]
</script>

...
```

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:



## 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.svelte** contains a local variable that holds hard-coded mocked data that enabled us to prototype our component quickly
- **ItemsList.component.svelte** 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/svelte-typescript>.

## ItemsList Component

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

## Chapter 4 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## ItemComponent

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

## Add unit tests support to our project

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

## Dependencies

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

## Configuration

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

## tsconfig.json file

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

### **package.json**

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

## **ItemComponent Unit Tests**

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

### **Item.rendering.test.ts**

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

### **Item.behavior.test.ts**

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

## **ItemsList component**

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

## **Chapter 5 Recap**

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



## What We Learned

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

## Observations

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

## Improvements

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

# Chapter 6 - State Management

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

## Store Interfaces

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

## Items Store Interfaces

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

### ItemsState.interface.ts

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

### ItemsStore.interface.ts

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

## Root Store Interfaces

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

### RootStore.interface.ts

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

## Store Implementation

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

### Items Store instance

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

#### Items.store.ts

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

### Root Store Instance

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

#### Root.store.ts

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

### Items.view.svelte

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

### App.svelte

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

## Web Browser

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

## ItemsList.component.svelte

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

## Loader Component

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

## Chapter 6 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## API Client Overview

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

## Domains

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

## The Main ApiClient

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

## ItemsApiClientOptions.interface.ts

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

## **ItemsApiClient.interface.ts**

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

## **ItemsApiClient.model.ts**

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

## **Mock Api Client**

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

### **Mock API instance**

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

## **Live Api Client**

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

## Live API instance

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

## Environment Variables

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

## Api Client Provider

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

## Store Instance updates

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

## Alternatives

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



## Chapter 7 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

### Constants.ts

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

### HttpRequestParams.interface.ts

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

### UrlUtils.ts

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

### HttpClient.interface.ts

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

### HttpClient.axios.ts

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

## **HttpClient.fetch.ts**

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

## **UrlUtils Unit Tests**

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

## **HttpClient: Unit Tests**

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

## **HttpClientAxios tests**

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

## HttpClientFetch tests

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

## Testing "get" responses

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

## ItemsApiClientModel Update

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

## Chapter 8 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## **.env files updates**

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

## **Config Interface**

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

## **Config files**

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

## **mock.json**

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

## **jsonserver.json**

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

## **localapis.json**

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

## **beta.json**

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

## **production.json**

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

## **tsconfig.json updates**

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

## **Config files map**

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

## **Config provider**

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

## File utils.ts

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

## Unit Tests

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

## HttpClient code updates

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

## **Chapter 9 Recap**

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

## **What We Learned**

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

## Observations

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

## Improvements

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

## Plugins: i18next, svelte-i18n

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

## Config updates

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

## ConfigInterface

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

## file mock.json

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

## Translation JSON data

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

## API Client updates

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

### file utils.ts

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

### file useLocalization.ts

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

## App.svelte updates

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

## Browser

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

## Important

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

## Chapter 10 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Directory localization/formatters

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

### File: useDateTimeFormatters.ts

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

### File: useNumberFormatters.ts

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

### File: index.ts

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

### Component DebugFormatters.component.svelte

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

## Updates to App.svelte

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

## Chapter 11 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Chapter 12 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Conventions

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

## General Strategies

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

## Text Elements

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

## Primitives View

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

## Chapter 13 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Button Elements

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

## Primitives View - update

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

## Toggle/Checkbox Elements

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

## Chapter 14 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Icon: ElIconAlert

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

## Interface ModalProps

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

## File ElModal.svelte

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

## File useModal.ts

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

## Updates to Primitives.view.svelte

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

## Browser

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

## Chapter 15 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Item Component - updates

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

## ItemsList Component - updates

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

## Summary

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

## Chapter 16 Recap

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

## What We Learned

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



## Observations

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

## Improvements

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

## Remove obsolete files

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

## Updates to enable TypeScript types declaration for .svelte files

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

## Svelte version

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

## Create Counter.svelte component

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

## Create SampleComp.svelte component

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

## Build our library

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

## Consuming our library

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

## Chapter 17 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Remove obsolete files

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

## random-id

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

## Build the library

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

## Chapter 18 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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

## Publishing the library

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

## Chapter 19 Recap

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

## What We Learned

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

## Observations

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

## Improvements

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



# (More Chapters Coming Soon)

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

# Naming Conventions

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

## Coding Standards

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

### TypeScript any

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

### Directory Names

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

### File Names

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

### Interface File Names

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

## **Components File Names**

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

## **Directory src**

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

## **Directory tests/unit**

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

# Resources

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

## Websites

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

## Tutorials

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

## Blogs

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

## Atomic Design

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

## Books

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