

USING ONLY FREE RESOURCES

Quick and free

eShop Project



JOÃO OSÓRIO

Introduction	3
Skill requirements for this book	3
Employed Technologies	3
Bootstrap	3
Angular	3
Google Sheets + API	4
Firebase	4
Amazon Affiliate Links	4
Chapter 1: Get your online shop running on your PC	5
Get some eShop design and layout templates	5
Get the local development tools	5
Create a new Angular Web Application	6
Get your homepage to be like the template for the product catalog	7
Chapter 2: Storing the product catalog on Google Drive	12
Creating a Google Sheets Repository	12
Setting up your affiliate account with Amazon	13
Picking products for your catalog from Amazon	14
Chapter 3: Activating Google Sheets API on GCP	15
Access the GCP Console and activate the API	15
Authenticating your API access	16
Test the API access to the catalog worksheet	18
Chapter 4: Connecting your eShop to your Google Sheet	19
Setting up the HTTP client	19
Developing the code to call the API	20
Displaying the products on the homepage	23
Chapter 5: Hosting your eShop using Firebase	25
Creating a new Firebase project	26
Adding Firebase to the project	27
Deploying the application	29
Chapter 6: Securing your API from unwanted access	30
Setting up Application and API restriction on the API credential	31
Chapter 7: Next Steps	33
Updating the Product Catalog	33
Further Functionalities	34
Annex I - section.component.ts	35

Introduction

This eBook directs you to the tools you need for a quick online shopping catalog project implementation. It uses only free resources on the Web, as well as free tier cloud services.

Skill requirements for this book

Some coding skills are required. You will be building files and content with the following formats:

- **Typescript** - TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale.
- **HTML** - Hypertext Markup Language, a standardized system for tagging text files to achieve font, color, graphic, and hyperlink effects on World Wide Web pages.
- **CSS** - stands for Cascading Style Sheets · CSS describes how HTML elements are to be displayed.
- **JSON** - JavaScript Object Notation is a lightweight format for storing and transporting data.

Also, you should be familiar with the API concept. APIs (Application Programming Interface) are mechanisms that enable two software components to communicate with each other using a set of definitions and protocols. In the case at hand, we will build a product catalog using a spreadsheet. The online shop web app will “talk” to the spreadsheet via API and show the products on your site.

Employed Technologies

Through the steps of implementing our online shopping catalog, will use the following frameworks, platforms and services:

Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Angular

Angular is a TypeScript-based, free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations.

Google Sheets + API

Google Sheets is an online spreadsheet app that lets you create and format spreadsheets and work with other people. It's included as part of the free, web-based, Google Drive document editors.

The Google Sheets API is a RESTful interface that lets you read and modify a spreadsheet's data.

Firebase

Developers can create iOS, Android, and Web apps using the Google-sponsored application development platform known as Firebase.

Tools are available from Firebase for monitoring analytics, reporting and resolving app errors, as well as developing marketing and product experiments.

Amazon Affiliate Links

Bloggers, publishers, and content producers can monetize their traffic with the aid of the Amazon Associates Program. With millions of products and programs available on Amazon, associates can earn from qualifying purchases and programs by directing their audience to their suggestions using simple link-building tools.

Chapter 1: Get your online shop running on your PC

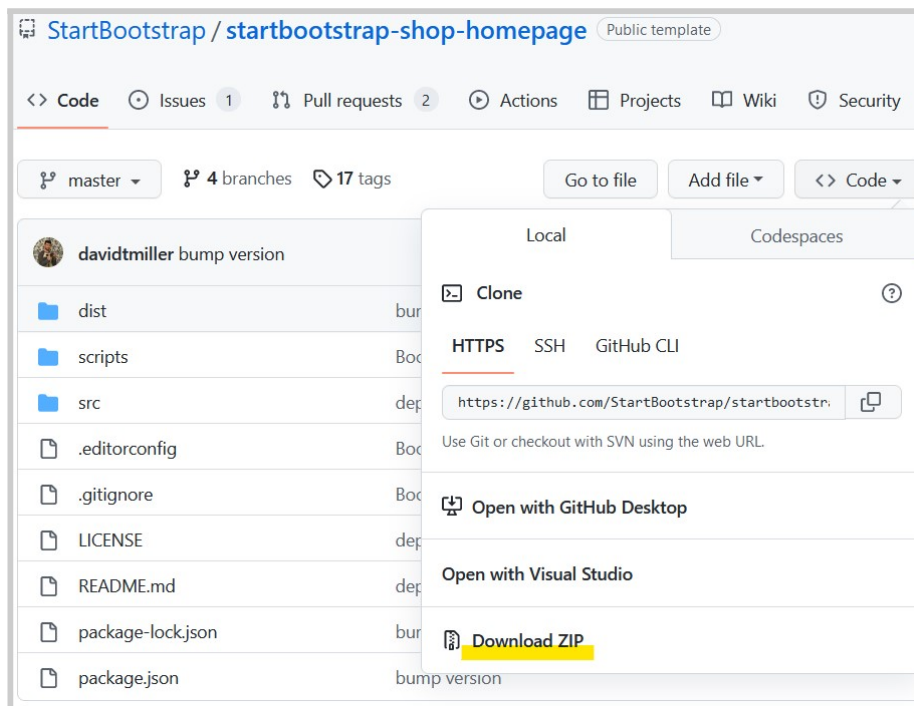
Get some eShop design and layout templates

The shop's pages will be developed reusing free templates based on the Bootstrap framework. Bootstrap is designed for front-end web development that prioritizes mobile responsiveness. It includes design templates for typography, forms, and numerous interface elements based on HTML, CSS, and JavaScript.

We will have 2 templates: [product catalog](https://github.com/ozzie-eu/startbootstrap-shop-homepage) and [product detail](https://github.com/ozzie-eu/startbootstrap-shop-item). These templates can be downloaded from these two Github projects:

- Product Catalog: <https://github.com/ozzie-eu/startbootstrap-shop-homepage>
- Product Detail: <https://github.com/ozzie-eu/startbootstrap-shop-item>

Download and unzip the files into different folders:



Keep them unaltered if you ever need to undo some changes to the shop's layout.

Get the local development tools

You will need to install some development tools on your PC to be able to edit, run and test your eShop locally. All of the tools mentioned here are free:

1. A code/text editor: We recommend [Visual Studio Code](https://code.visualstudio.com/). It's free, multi-platform and, if you desire, has all sorts of extensions to help you edit and publish your files. This will be the editor used on the screen captures presented.
2. [NodeJS](https://nodejs.org/en/): a JavaScript runtime built on Chrome's V8 JavaScript engine.

3. [Angular](#): a TypeScript-based, free and open-source web application framework that is governed by a group of people and organizations, including the Angular Team at Google.

The editor, Visual Studio Code, and NodeJS have installation assistants to deploy the software locally. Angular uses an NPM (Node Package Manager) install command. As such, Angular should be installed last.

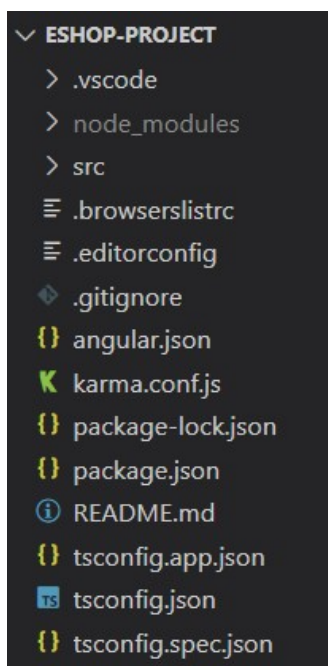
Create a new Angular Web Application

Open a new terminal window, on the folder you want your local project stored. Run the following Angular CLI command:

```
> ng new eshop-project
```

On the options asked, give the following answers:

- Would you like to add Angular routing? **Yes**
- Which stylesheet format would you like to use? **CSS**



Run Visual Studio Code and open the recently created local folder. The folder structure should be similar to the one on the left.

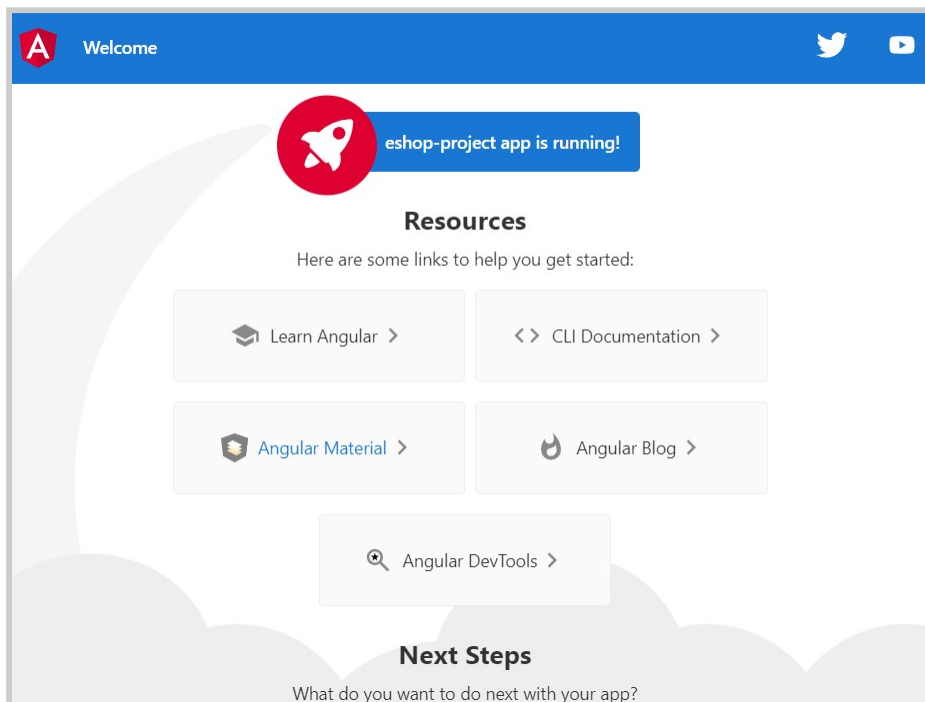
The src folder contains all the items we add and edit, and it also contains the three primary folders app, assets, and environment. The materials needed for the project, including CSS files, JavaScript files, and images, are created inside the assets folder, while all the components and pages are created inside the app folder.

By default, the Angular CLI sets up the folder as a Git repository. It's optional to use it, but I encourage it because if you do something wrong, there's always a way to revert your changes.

On the terminal window, inside the project folder, run the command to test the project locally:

```
> ng serve
```

On success, it should open the default project page:



This means everything is OK and we can start working on customizing our application's look and feel.

Get your homepage to be like the template for the product catalog

Go to the folder where you downloaded and unzipped the catalog template files and open the file “\startbootstrap-shop-homepage-master\dist\index.html”. Browsing through the HTML comments, you can identify the following sections:

```
...
<!-- Navigation-->
...
<!-- Header-->
...
<!-- Section-->
...
<!-- Footer-->
...
```

Let's build a different Angular component for each layout section. Run these four command on the terminal window:

```
> ng generate component layout/navigation
> ng generate component layout/header
> ng generate component layout/section
> ng generate component layout/footer
```

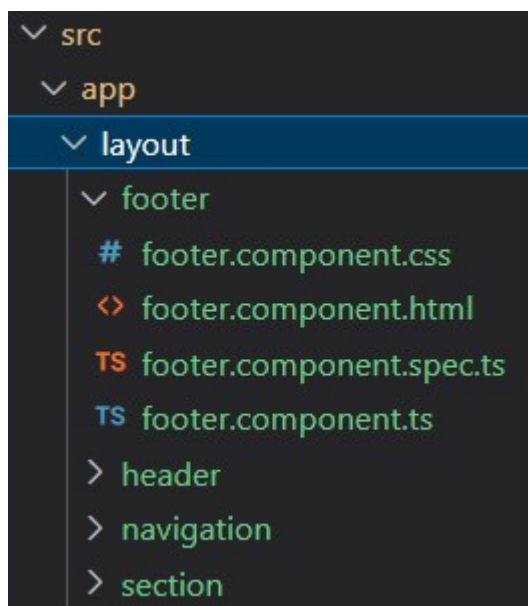
The created components have three file each:

- HTML: Contains the markup for the component. The contents of this file are also known as the template. The template determines the view or what you see in the browser.
- TS: Also known as the class, contains the logic behind the UI interactions.
- CSS: Contains the styles for the component. Use this file to define styles that apply to this specific component.
- SPEC.TS: file used for unit testing.

The output for each command is similar to the following:

```
CREATE src/app/layout/navigation/navigation.component.html (25 bytes)
CREATE src/app/layout/navigation/navigation.component.spec.ts (627 bytes)
CREATE src/app/layout/navigation/navigation.component.ts (291 bytes)
CREATE src/app/layout/navigation/navigation.component.css (0 bytes)
UPDATE src/app/app.module.ts (498 bytes)
```

Inside each newly created folder are the same kind files. The updated folder structure is this:



Now we must place the corresponding HTML inside each of the components. The navigation component will take the following code: