Exchange rate app

Story

It is very important to know what's going on at the currency market. Your company has been involved in creating nice looking information about current exchange rates.

Your job is to create a module displaying current exchange rates in a form of carousel.

What are you going to learn?

You will learn and practice the following topics using Angular:

- modules
- pipes
- more OOP
- web workers

Tasks

Title	Acceptance criteria
Get data from external API	Use [Exchange rates API](#) and present given rates: - British Pound to Euro - Swiss Franc to U.S. Dollar - U.S. Dollar to British Pound - Whenever carousel item switches, app retrieves fresh data

Represent given rates in a form of carousel	Each exchange rate should be displayed at the center of box containing carousel ● Elements are automatically switching between each other ● The views are looping ● User can manually switch between views ● Each currency has background with any image coming from one of countries
Rates are easy to change	I can easily change / add exchange rates to display ● As a developer, when I open the project, there should be configuration file ● Configuration file contains informations: 　○ Rates that application can display 　○ Each exchange rate being displayed so I can choose between them 　○ Each country currency has attached background image in a form of url ● Prepare more possible options of exchange rates, also corresponding background images
Represent rates in a form of an chart	**Each exchange rate can be displayed in a form of chart** ● **Implement lazy loading module which displays chart representing chosen exchange rate** ● **Using ngx-charts library display historical exchange rate details of selected currencies** ● **Selection is visible under the chart** ● **For mapping that amount of data use Web Workers API** ● **About background processing using web workers in Angular** ● **Since the module is lazy loaded, take care of splitting api services within the project - this will allow creating / destroying web workers at the right time.**

RxJS: Learnrxjs Animation Visualisation
Reactive Forms: Documentation
Architecture: Smart vs Presentational