

# Frontend **Angular/React** Exercise: Currency Exchanger!

## What is Currency Exchanger?

Currency Exchanger is an app the helps customers to hand different currency conversions.

## **Technology constraints**

At the digital factory, we currently have a JavaScript web applications. We would like you to develop this application in **Angular** or **React**. We do *not* expect you to build the API – rather we would like you to use the below provided API.

### What are we looking to test?

- The overall software architecture of the application
- The structure and quality of the code itself
- The use of well-known patterns for JavaScript frameworks

## What are we not looking to test?

 We don't expect you to be a designer. Something simple and intuitive is all we need

### **External Resources**

- Fixer API
- Hint: Only utilize available free APIs to work around paid
  APIs limitations and achieve same application functionality

# 2

# Frontend **Angular/React** Exercise: Currency Exchanger!

## Please consider the following:

- Git
  - Push your code to a Git repository using (Github/GitLab/Bitbucket/AzureDevOps Repos) only.
  - Keep your development history intact and in multiple commits as you see fit.
- Structure
  - Utilize modularity and clean patterns.
- Style
  - Utilize your own CSS or Sass for styles as much as possible instead of relying totally on UI Frameworks (ex. Bootstrap, etc.).
  - Application should be responsive (mobile/Desktop)
- Typescript
  - Utilize Typescript features such as Interfaces/Types in you code when feasible.
- Rxjs
  - Utilize operators when feasible.
- Formatting/Linting
  - Make sure that you code is formatted and free of linting issues.
- Unit Tests (Bonus)
  - Include simple unit test when feasible

# Frontend **Angular/React** Exercise: Currency Exchanger!

The team is trying to ease the issues that customers face while handling different currency conversions. So they are planning to build a feature rich currency converter, Please find the stories backlog given below:

Sprint 1:

## FF-1

As a user I want to be able to convert currencies, for that we need a "Home" page with following features:

#### **ACCEPTANCE CRITERIA**

- □ I should see a sticky header with a Logo (left) and Nav links (right) as "EUR-USD Details" and "EUR-GBP Details" to go to respective "Details" page (see FF-3)
- ☐ I should see a converter sticky panel (use 1 EUR to USD as defaults) (see FF-2)
- ☐ I should see 3x3 cards grid with converted value of the entered amount to the 9 most popular currencies (Duplication of "From" and "To" currencies in cards grid are allowed)

## FF-2

As a user I want to able to convert currencies for that we need a converter panel with following features:

### **ACCEPTANCE CRITERIA**

- I should see a sticky panel with below elements:
  - An amount input (numbers only), all other elements should be disabled until a valid value entered
  - ☐ Dropdowns showing all the available currencies in a FROM and TO lists
  - Button to swap the currencies in FROM and TO, Button to convert, and button to redirect to details page for the selected FROM and TO.
  - Converted result displayed

## FF-3

As a user, I want to see a "Details" page of the 'From' and 'To' currencies with the following features:

**Sprint 2:** 

#### **ACCEPTANCE CRITERIA**

- I should see the full name of the "From" currency
- ☐ I should see a button to go back to Home page (see FF-1)
- I should see a sticky converter panel (see FF-2) with chosen currencies preselected, entered amount, details button hidden and 'From' dropdown disabled
- ☐ I should be able to change the "To" dropdown then click convert and it reflects on all details
- ☐ I should see a chart showing historical data for selected currencies (see FF-4)

## FF-4

As a user, I want to see a simple chart of historical data for selected currencies:

### **ACCEPTANCE CRITERIA**

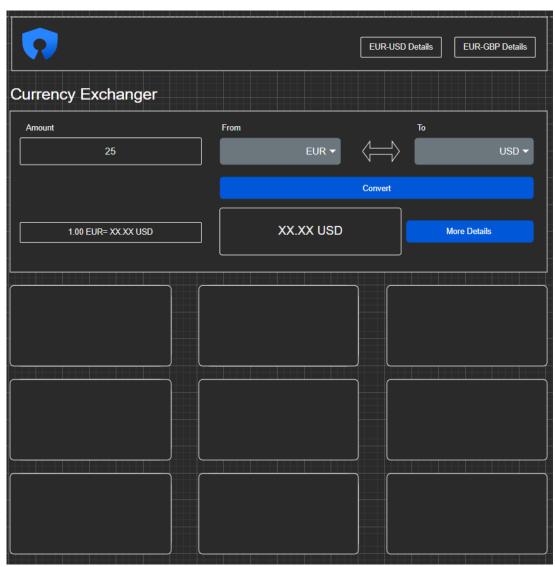
- I should see the monthly historical rates for the selected currencies for past year.
- I should see months on horizontal axis and Rates on vertical axis
- Monthly rate is calculated based on rate in last day of that month

TIP: focus on doing fewer stories but doing them right!

# Frontend **Angular/React** Exercise: Currency Exchanger!

Design Wireframes: (use this as a reference but feel free to be creative)

Home Page Wireframe



Details Page Wireframe

