

Coding Test

You're going to code (the fun part!).

We'd like to get an idea about how you like to code and if it fits with how we like to code.

Please don't spend on this task more than 6 hours. We don't expect anyone to create fully working application in a such short time. Plan your work respectively. Show us how you approach planning.

Exercise

We think everyone should have the opportunity to invest in property, which is why we're disrupting the status quo. This is lending and investing without the banks.

Mortgages simplified. We connect people who want to invest their money, with investments to those who want to borrow.

One of the important parts of our business is to give our investors a way to invest in a loan for them to earn a return (monthly interest payment).

Exercise description

A sample JSON file (current-loans.json) is provided to use as a stub data for imaginary end point.

You should create a small single page HTML/JS application, which would have the following functionality:

- 1) A sample JSON file (current-loans.json) is provided to use as a stub data.
 1. Don't worry about stubbing API response. Assume that this data is available within the app's source code (hard coded).

2) You should create a small single page HTML/JS application. It should have the following functionality:

1. User is able to see all current loans (you can find the data in current-loans.json file)
2. The User is able to see the number representing the total amount of possible investments.

3) After User clicks on a loan, the form should appear.

1. User can put numeric value (invested amount) in the input
2. User can click button labelled “Invest”
3. User can close popup

4) If User decides to invest:

1. the form should be closed
2. the available amount, for the loan User invested into, should decrease
3. the total available number should also adjust accordingly.
4. The loan on which User have invested should have some visual indication that you have invested in it

How to pass the test

- Submit working Angular code fully tested.
- If you know TDD a red-green-refactoring process would be very appreciative.
- Once done create a git bundle leaving all the commit history