



## Test Assignment – Frontend (SPA): Реализовать UI для обмена валют

14 сентября 2022 в 17:02

От: Evgenii Zomchak<ezomchak@coinloan.io>

Кому: Evgenii Zomchak<016@rambler.ru>

### Общие требования:

реализовать на `next.js`;  
стили на `SCSS`;  
шаблоны на  `pug`;  
чистый код.

### Задача делится на две части:

Подготовка данных для UI обмена валют.

Реализация UI обмена валют.

### 1. Подготовка данных для UI обмена валют

Предположим, что back-end часть для обмена еще не готова, поэтому необходимо реализовать mock API для того, чтобы разработка front-end части не стопорилась. Для mock-а мы имеем ряд статичных данных, на базе которых можем динамически генерировать данные, которые впоследствии будем получать по API.

Статичные данные:

список из 10ти+ валют (выберите любые фиатные валюты);

список из 5ти комиссий (от 1го до 5ти процентов):

1.00

2.00

3.00

4.00

5.00

Динамически сгенерированные данные:

список валютных пар;

список курсов валют.

На базе указанных выше статичных данных нужно сгенерировать список валютных пар, с которыми будет работать UI обмена валют. При этом:

каждая валюта должна иметь пару со всеми другими валютами;

каждой полученной паре нужно присвоить одну случайно выбранную комиссию из пяти указанных выше вариантов.

Итоговый список валютных пар может выглядеть так:

```
[
  {
    "base_currency": "EUR",
    "quote_currency": "USD",
    "commission": "5.00"
  },
  ...
]
```

**Валютная пара** (англ. currency pair) — отношение цен двух валют, входящих в данную пару, на валютном рынке. Первая валюта в паре называется базовой, а вторая — валюта котировки. Отсюда и название полей base/quote.

Кроме того, необходимо сгенерировать список курсов валют на базе полученного списка валютных пар, при этом:

список курсов будет отдавать отдельный API endpoint, поэтому нельзя вшивать курсы в список пар — необходим отдельный список;

для каждой пары будет один курс;

значения курсов можно генерировать случайным образом в любом разумном диапазоне, к примеру от 10 до 100, не опираясь на реальные курсы валют.

Итоговый список курсов валют может выглядеть так:

```
[
  {
    "pair": "EUR/USD",
    "rate": "10"
  },
  ...
]
```

## 2. Реализация UI обмена валют.

После того, как мы получили mock API, который выдает список валютных пар с нужными параметрами, а так же отдельно — курсы валют по данным парам, можно приступить к

реализации интерфейса обмена валют.

Интерфейс должен иметь:

Input суммы для базовой валюты;  
выпадающий список для выбора базовой валюты;  
Input суммы для котируемой валюты;  
выпадающий список для выбора котируемой валюты;  
кнопку подтверждения обмена.

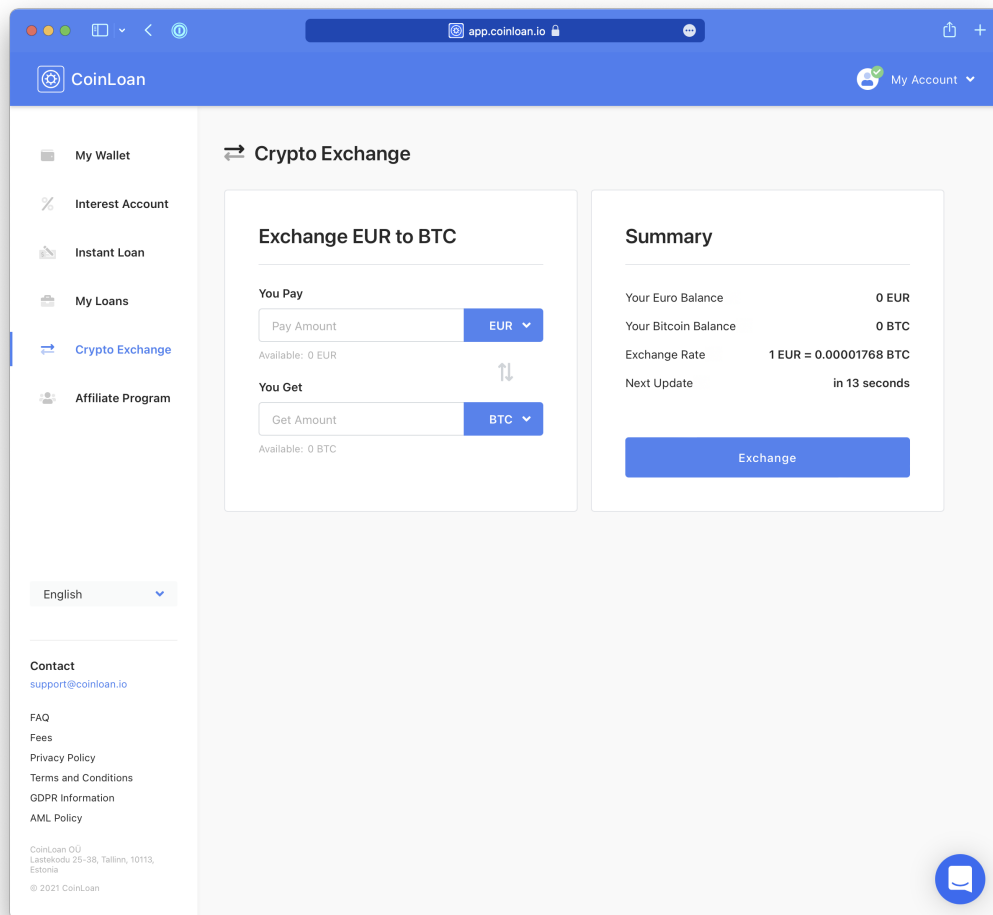
Обмен должен уметь:

в выпадающем списке базовых валют выводить все валюты, которые доступны в качестве базовых валют на базе списка валютных пар;  
в выпадающем списке котируемых валют выводить список валют, которые доступны для выбранной базовой валюты в качестве котируемой валюты на базе списка валютных пар;  
тихо валидировать вводимые в input данные как float/decimal (в идеале сделать универсальный компонент input-а для float/decimal данных), то есть в случае ввода не валидных данных просто очищать поле;  
при изменении суммы в любом из input-ов пересчитывать значение в другом input-е;  
при пересчете одной валюты в другую помимо курса должна учитываться комиссия, указанная для данной валютной пары — то есть, чтобы получить значение для котируемой валюты, необходимо перевести значение базовой валюты в котируемую по курсу и отнять процент комиссии от полученного значения;  
раз в 30 секунд получать новые курсы валют и пересчитывать значения в input-ах по новому курсу, если в них что-то указано (UI таймера реализовывать не нужно);  
после нажатия на кнопку подтверждения обмена показать экран успешного обмена.

Пример UI: <https://app.coinloan.io/exchange>

Интерфейс реализовать опираясь на предоставленный пример, навыки дизайна проверяться не будет, интересует верстка.

Screenshot



## Результат

Итоговый проект необходимо залить в публичный или приватный репо (на ваш выбор) Github или Gitlab.

---

Прикреплённые файлы: