

Тестовое задание на позицию фронтенд-разработчика в Skybonds

В рамках тестового задания предлагается решить две любые задачи из приведенных ниже.

К решению каждой задачи необходимо приложить:

- субъективную оценку сложности задачи по шкале от 1 до 10,
- предварительную оценку трудозатрат,
- фактические трудозатраты.

При оценке выполненных задач мы обращаем внимание на следующие факторы:

- сопроводительное письмо,
- время выполнения,
- решение задачи,
- производительность кода,
- объем и читаемость кода,
- наличие/качество тестов.

1. Долевое строительство

Дан массив из N долей, представленных в виде N рациональных:

```
[  
  '1.5',  
  '3',  
  '6',  
  '1.5'  
]
```

Задача

Написать программу, представляющую эти доли в процентном выражении с точностью до трех знаков после запятой:

```
[  
  '12.500',  
  '25.000',  
  '50.000',  
  '12.500'  
]
```

Ожидаемое решение

- программа на языке JavaScript в виде одного исходного файла или сниппета на любом открытом веб-хостинге, без использования сторонних библиотек,
- вычислительная сложность алгоритма и оценка необходимой памяти для его выполнения,
- ограничения на размер входного массива, при котором алгоритм будет выполняться разумное время (до 5 секунд, например).

2. Кэш данных об облигациях

Дана функция, которая получает из API данные о финансовых показателях облигаций за заданную дату по массиву идентификаторов облигаций (ISIN):

```
const getBondsData = async ({date, isins}) => {
  const result = await http.post({
    url: `/bonds/${date}`,
    body: isins
  });

  return result;
};
```

Пример вызова функции:

```
getBondsData({
  date: '20180120',
  isins: ['XS0971721963', 'RU000A0JU4L3']
});
```

Результат:

```
[{
  isin: 'XS0971721963',
  data: {...}
}, {
  isin: 'RU000A0JU4L3',
  data: {...}
}]
```

Задача

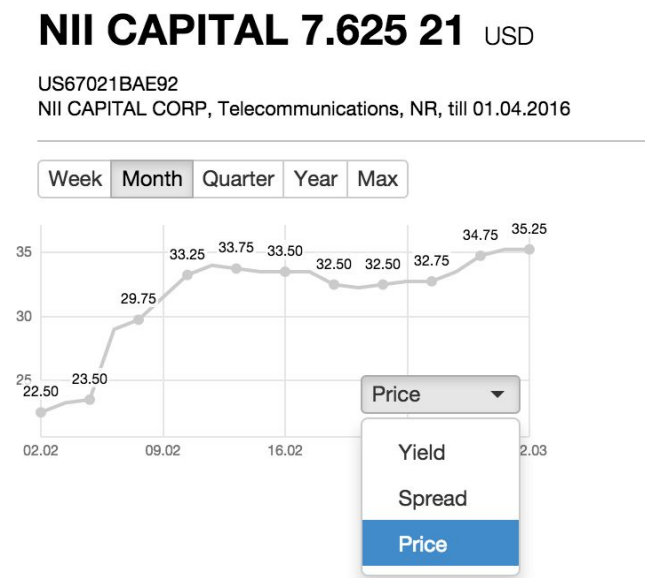
Изменить код функции, реализовав кэш на стороне клиента.

Ожидаемое решение

- программа на языке JavaScript в виде одного исходного файла или сниппета на любом открытом веб-хостинге, без использования сторонних библиотек,
- вычислительная сложность алгоритма и оценка необходимой памяти для его выполнения.

3. Карточка облигации

Дан макет карточки облигации:



Задача

Создать приложение по макету на основе React и Redux, которое реализует следующие функции:

- получает данные об облигации из API по идентификатору облигации (ISIN) и отображает их на карточке,
- позволяет выбрать период за который строится график,
- позволяет изменить параметр, по которому строится график.

Комментарии к задаче

- в качестве основы приложения рекомендуем использовать Create React App (<https://github.com/facebook/create-react-app>),
- библиотеку для построения графика можно выбрать любую,
- данные для графика можно сгенерировать случайным образом,
- все спорные моменты решайте в пользу уменьшения времени на выполнения задания и упрощения кода.

Ожидаемое решение

- код приложения на языке JavaScript в репозитории на GitHub.