

Mister-BITCoin

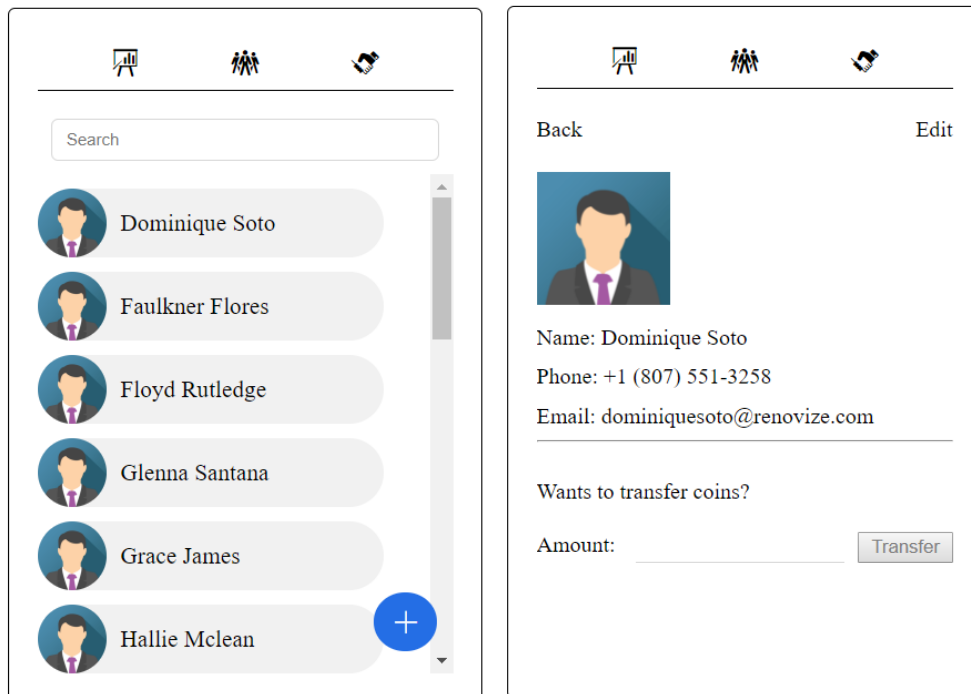
Building stuff with React!




Lets build a digital wallet for holding my bitcoins and sending (paying) them to my contacts.

One step at time, lets start from managing the contacts.


Part 1 Contacts

Implement full CRUD for a **Contact** entity : `{_id, name, phone, imgUrl}`



Back






Name:


Phone:

Email:

Save

Back
Delete



Name:

Phone:

Email:

Save

Services

Implement a `ContactService` with the following functions
(`Promise.resolve()` for now):

- `getContacts(query = null)`
- `getContactById(id)`
- `deleteContact(id)`
- `saveContact(contact)`
- `getEmptyContact()`

Copy the JSON from [here](#) into your service

```
{
  "_id": "5a566402ed1cf349f0b47b4d",
  "imgUrl": "../../../assets/image_avatar.png",
  "name": "Rachel Lowe",
  "email": "rachellowe@renovize.com",
  "phone": "+1 (911) 475-2312"
},
```

Pages

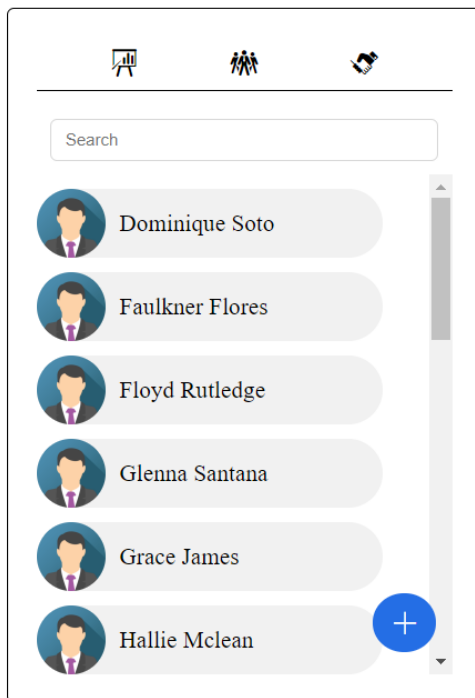
`<Home>` (route: `/`)

Some intro text for now

`<ContactApp>` (route: `/contact`)

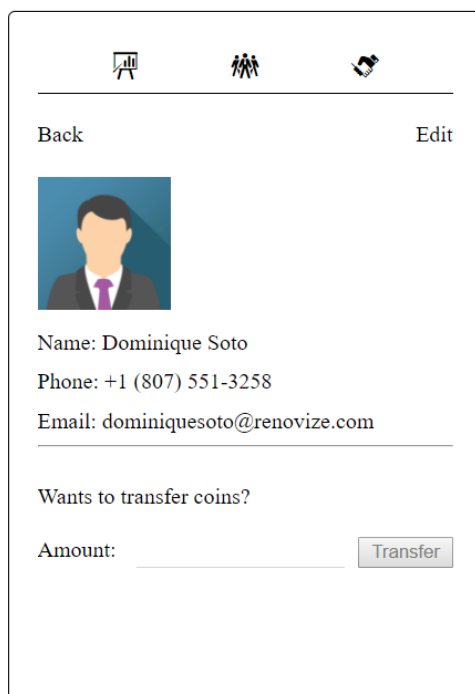
Gets contacts from `ContactService` and renders a `<ContactList>` component, passing down the contacts.

It will later also render the `<ContactFilter>` and pass down only a filtered list of contacts.



`<ContactDetails>` (route: `/contact/:id`)

Receive an id as URL param and gets a contact from the `ContactService`, display that contact in full






`<ContactEdit>` (route: `/contact/edit/:id?`)

Allows Adding and Editing a contact


- Gets a contact from the service by id or start with a new contact

- Allow editing the name and phone of that contact

CREATE MODE:

Back






Name:

Phone:

Email:


Save

EDIT MODE:

Back

Delete



Name:

Phone:

Email:

Save

Components

<ContactPreview>

Props: contact

Render a div with an image and a span for preview

<ContactList>

Props: contacts

Render each contact previews inside a Link

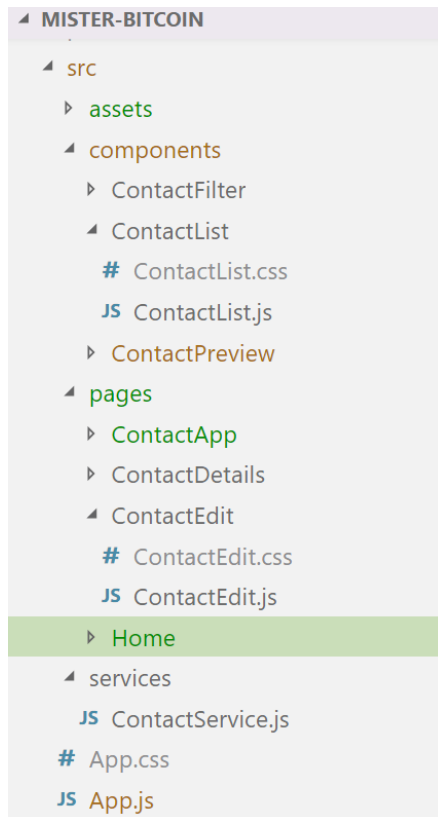
<ContactFilter>

Props: onFilter

Allows free text search by name / phone and calls onFilter() on every keypress (onChange), passing a filter object e.g. : {term: 'puk'}

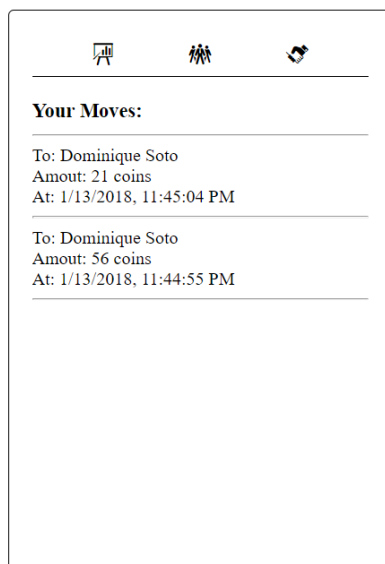
Directory structure

Please follow the following structure and naming conventions:



Part 2 Moving Coins

- Build a <Signup> page, when user is not known we route to this page
- The Signup page just requests a name
- New user gets 100 coins when signup
- On the Home page, show the Bitcoin rate (Use a Bitcoin value API such as [this](#))
- Inside <ContactDetails> show a Transfer Fund form (with an amount field)
- when submitted, we
 - (1) reduce from the user balance (this money goes nowhere!)
 - (2) call MoveService.addMove() to add a move to the user.moves[] array:
{at: 2652712571, amount: 2, to: "Moshiko"}
- Build the <MoveList> page that shows a list of moves (transactions)



Services:

<UserService>

This service does NOT work with Promises, its all synchronous here

Functions:

- getUser() (returns user synchronously)
- saveUser(user) - save to localStorage
- getEmptyUser()

User object should look like:

```
{
  name: 'Puki',
  coins: 100,
  moves: []
}
```

Use the local storage to save/ load the user.

<MoveService>

Functions:

- addMove(move)
- getMoves()
- setMoves(moves)

Move Object should contain the props:

```
{  
  at: 173989218,  
  amount: 100,  
  to: "Popo"  
}
```

<BitcoinService>

Functions:

- getRate(coins) (returns Promise)

Pages

<Signup> (route: '/signup')

Ask for user name and save the new user in local storage and local variable using the [UserService](#)

Example:

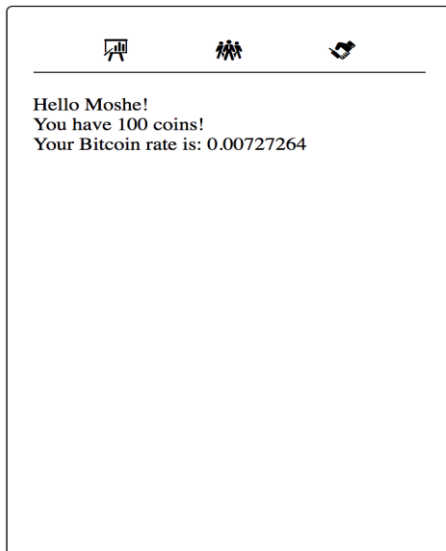


<Home> (route: '/')

Get the user from [UserService](#) and display:

- Name
- Coins
- Bitcoin rate (using the api)

Example:



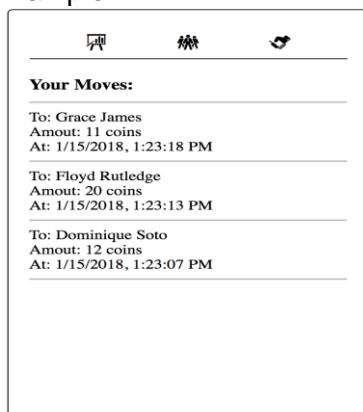
<ContactDetails> (route: `'/contact/:id'`)

The same component that you already created, but now need to add the component **Transfer**

<UserMoves> (route: `'/move'`)

Should render a list of moves

Example:






Components

<Transfer>


Props: `onTransfer`

When clicking on the button the `onClick` event should invoke the `'onTransfer'` and pass into him the coins amount.

Example:



BackEdit



Name: Dominique Soto

Phone: +1 (807) 551-3258

Email: dominiquesoto@renovize.com

Wants to transfer coins?

Amount:

Part 3 Getting serious

- Add a store to your solution and manage the state in there

Part 4 Over the edge

Implement a PWA , or switch to React Native