Ethereum Web Wallet DIY

Francesco Canessa

@makevoid

Lead Software Developer

Applied Blockchain

Rise London - **Blockchain Week '18** - Development Workshop

Ethereum Web Wallet

1. Setup

1. Generate a Private Key

2. Derive an Ethereum address

3. Get the address balance

4. Sign transansaction

5. Broadcast transaction

Setup

Requirement:

Node installed 7.6+

Upgrade or use github:yortus/asyncawait

https://github.com/yortus/asyncawait#6-quick-start

Alternative: Runkit (https://runkit.com)

Setup

```
$ node -v
```

`v7.6+ (at least)`

`v8 is good (aim for v8.9.x)`

9 should be ok as well

Setup

Check Async-Await support

https://runkit.com/makevoid/check-async-support

if you don't have native support, you can use **github:yortus/asyncawait**

Hello Node

```
console.log("hello world")
```

Hello Bitcore-Lib

```
npm init -fy
```

```
npm install --save bitcore-lib
```

Generate a Private Key

```
const bitcore = require('bitcore-lib')
const PrivateKey = bitcore.PrivateKey

const privateKey = new PrivateKey()
console.log(privateKey.toString())
```

https://runkit.com/makevoid/bitcore-lib-privatekey

Derive the PublicKey

```
const bitcore = require('bitcore-lib')
const PrivateKey = bitcore.PrivateKey

const privateKey = new PrivateKey("711fd1eeecec8bb8c912129466504de109a17e
const publicKey  = privateKey.toPublicKey()
console.log(publicKey.toString())
```

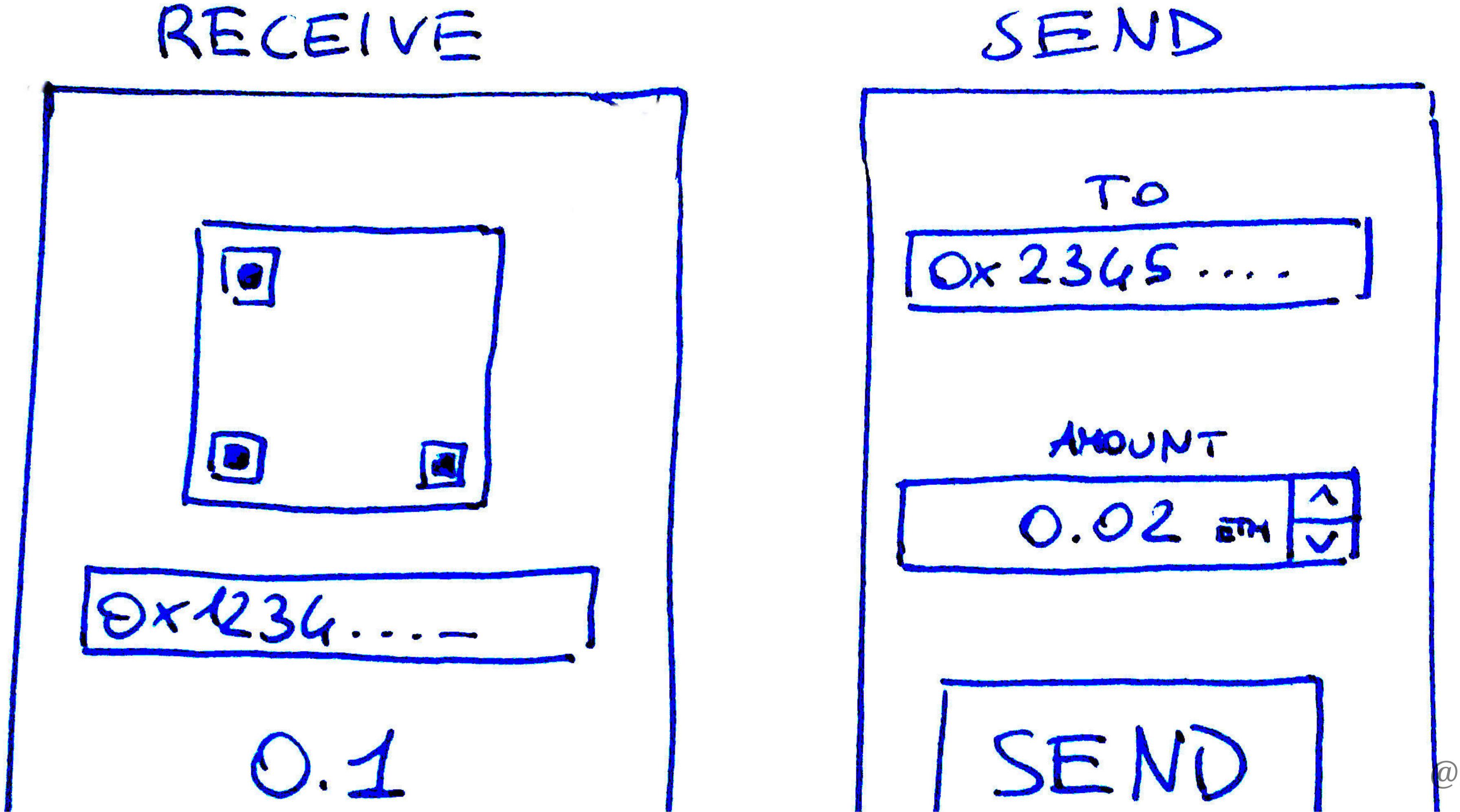
https://runkit.com/makevoid/bitcore-lib-address

Generate an address (bitcoin)

```
const bitcore = require('bitcore-lib')
const PrivateKey = bitcore.PrivateKey

const privateKey = new PrivateKey()
const publicKey  = privateKey.toPublicKey()
console.log(publicKey.toString())
const address    = publicKey.toAddress().toString()
console.log(address)
```

https://runkit.com/makevoid/bitcore-lib-address

Wallet Mockup



Setup Web3

```
const Web3 = require('web3')

const web3 = new Web3("https://mainnet.infura.io")
```

http://web3js.readthedocs.io/en/1.0/web3.html

Setup Web3

```
npm init -fy

npm i --save web3-eth-accounts
npm i --save isomorphic-fetch
npm i --save isomorphic-form-data
```

Optional:

```
npm i --save-dev  lerna
npm i --save      ethereum/web3.js#1.0
```

Setup Web3

```
const Web3 = require('web3')

const web3 = new Web3("https://kovan.infura.io")
```

Setup Web3

```
const Web3 = require('web3')

const provUrl  = "https://mainnet.infura.io"
const provider = new Web3.providers.HttpProvider(provUrl)

const web3 = new Web3(provider)
```

Setup Web3 (accounts)

```
const Accounts = require('web3-eth-accounts')

const accounts = new Accounts()
```

Derive an ethereum address

```
const bitcore = require('bitcore-lib')
const PrivateKey = bitcore.PrivateKey

const Accounts = require('web3-eth-accounts')
const accounts = new Accounts()

const privateKey = new PrivateKey()
const key = `0x${privateKey.toString()}`
const account = accounts.privateKeyToAccount(key)
console.log(account.address)
```

https://runkit.com/makevoid/bitcore-lib-web3-accounts

Generate an ethereum "account"

```
const Accounts = require('web3-eth-accounts')
const accounts = new Accounts()

const account = accounts.create()

console.log(account.address)
```

https://runkit.com/makevoid/web3-eth-accounts-address

Load the private key (Node) 1/2

```
const Accounts = require('web3-eth-accounts')
const accounts = new Accounts()

const account = accounts.create()

console.log(account.privateKey)
```

Extra step: save key into `private-key.txt`

https://runkit.com/makevoid/web3-eth-accounts-private-key

Load the private key (Node) 2/2

```
const fs = require('fs')
const readFileSync = fs.readFileSync
const Accounts = require('web3-eth-accounts')
const accounts = new Accounts()

const key = readFileSync("private-key.txt")

const account = accounts.privateKeyToAccount(key)

console.log(account.address)
```

Load the same key (Node)

```
const fs = require('fs')
const readFileSync = fs.readFileSync
const Accounts = require('web3-eth-accounts')
const accounts = new Accounts()

const key = readFileSync("private-key.txt")

const account = accounts.privateKeyToAccount(key)

console.log(account.address)
```

run this again - notice that it will return the same address! :D

Receive Ethers

https://duckduckgo.com

!qr 0x1234...

tip for future UI - handy QR npm package:

https://www.npmjs.com/package/davidshimjs-qrcodejs

Check transaction on block explorer

https://etherscan.io/address/0x1234...

https://kovan.etherscan.io/...

(for testnet)

Kovan Faucet

https://gitter.im/kovan-testnet/faucet

Check address balance

```
require('isomorphic-fetch')

const getBalance = async (address) => {
  const balanceUrl = `https://api.etherscan.io/api?module=account&action=
  let resp = await fetch(balanceUrl)
  resp = await resp.json()
  return Number(resp['result'] || 0)
}

(async () => {
  const balance = await getBalance("0x738d145faabb1e00cf5a017588a9c0f9983
  console.log(balance)
})()
```

https://runkit.com/makevoid/etherscan-getbalance

Etherscan API: https://etherscan.io/apis#accounts

Create Transaction - 1/2

```javascript
const createTx = async ({recipient, account, value}) => {
  const txData = {
    value: value,
    to: recipient,
    gas: 21000,
    gasPrice: 5000000000, // 5 gwei
    from: account.address,
    // nonce: 1,
  }
  const tx = await account.signTransaction(txData)
  const txRaw = tx.rawTransaction
  console.log("TX RAW", txRaw)
  return txRaw
}
```

Create Transaction - 2/2

```
(async () => {
  const fs = require('fs')
  const readFileSync = fs.readFileSync
  const Accounts = require('web3-eth-accounts')
  const accounts = new Accounts()

  const key = readFileSync("private-key.txt")
  const account = accounts.privateKeyToAccount(key)

  createTx({
    account:    account,
    recipient:  "0xD9dDF72Ef671261Cb2266B9D924c5980C5186699",
    value:      100000000000000, // 100 szabo
  })
})()
```

https://ethgasstation.info - https://converter.murkin.me

For reference:

```
const createTx = async ({recipient, account, value}) => {
  const txData = {
    value: value,
    to: recipient,
    gas: 21000,
    gasPrice: 5000000000, // 5 gwei
    from: account.address,
    // nonce: 1,
  }
  const tx = await account.signTransaction(txData)
  const txRaw = tx.rawTransaction
  console.log("TX RAW", txRaw)
  return txRaw
}

(async () => {
  const fs = require('fs')
  const readFileSync = fs.readFileSync
  const Accounts = require('web3-eth-accounts')
  const accounts = new Accounts()
```

Broadcast Transaction

https://etherscan.io/pushTx

(manual)

Broadcast Transaction

programmatically

```
require('isomorphic-fetch')
require('isomorphic-form-data')

const broadcastTransaction = async (rawTx) => {
  const broadcastUrl = "https://api.etherscan.io/api"

  // ...
}
```

Broadcast Transaction

```
const broadcastTransaction = async (rawTx) => {
  const broadcastUrl = "https://api.etherscan.io/api"
  const data = new FormData()
  data.append('module', 'proxy')
  data.append('action', 'eth_sendRawTransaction')
  data.append('hex', rawTx)
  data.append('apikey', '3DQFQQZ51G4M18SW8RDKHIMERD79GYTVEA') // please u
  let resp = await fetch(broadcastUrl, {
    method: "post",
    body: data,
  })
  resp = await resp.json()
  c.log("broadcast Tx:", resp)
  return resp
}
```

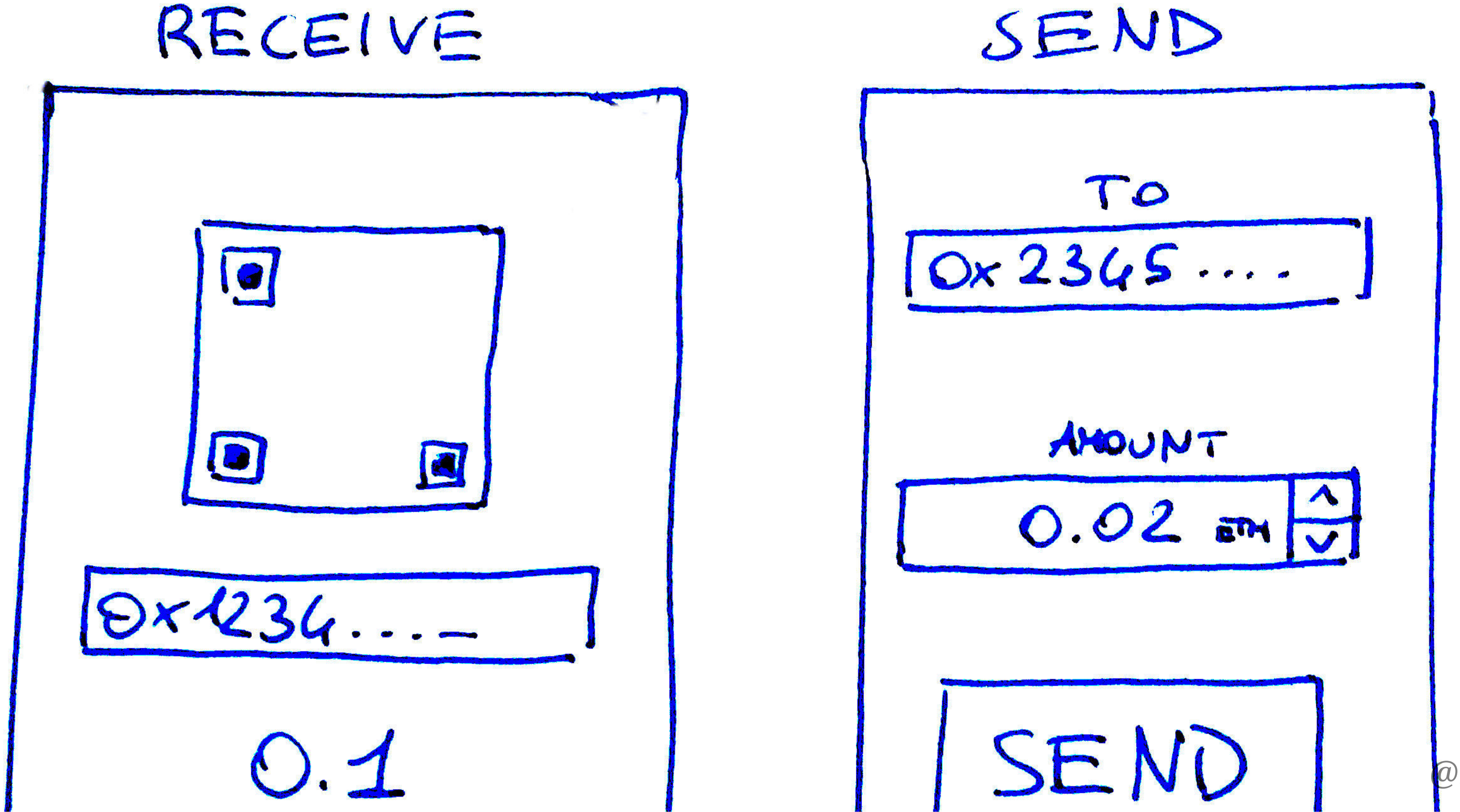
Broadcast Transaction

```
// ...

(async () => {
  const rawTx = "0xf86a0285012a05f20082520894d9ddf72ef671261cb2266b9d924c

  await broadcastTransaction(rawTx)
})()
```

For reference:

```
require('isomorphic-fetch')
require('isomorphic-form-data')

const broadcastTransaction = async (rawTx) => {
  const broadcastUrl = "https://api.etherscan.io/api"
  const data = new FormData()
  data.append('module', 'proxy')
  data.append('action', 'eth_sendRawTransaction')
  data.append('hex', rawTx)
  data.append('apikey', '3DQFQQZ51G4M18SW8RDKHIMERD79GYTVEA') // please u
  let resp = await fetch(broadcastUrl, {
    method: "post",
    body: data,
  })
  resp = await resp.json()
  c.log("broadcast Tx:", resp)
  return resp
}
```

Wallet Mockup



Tools

- Browserify

- Babel

```
npm i -g browserify babelify
```

Tools - Browserify

RUN:

```
browserify js/index.js > js/dist/bundle.js
```

ADD:

```
<script src="js/dist/bundle.js" charset="utf-8"></script>
```

Tools - Browserify (bonus)

```
npm i -g watchify
```

RUN:

```
watchify js/index.js -o js/dist/bundle.js
```

Tools - Babel

`.babelrc`

```
{
  "presets": ["env"]
}
```

Have fun!

`@makevoid`

Applied Blockchain

Few minutes left!

Time's up!

Thanks for attending!

`@makevoid`

Applied Blockchain