

project 最终报告

姓名: 陈谱一 学号: 16340026 专业: 软件工程(数媒)

Github 地址

https://github.com/chenpy26/angel_invest

webpack 打包部分代码参考了 truffle 官网的 demo。

选题背景

我选择的项目是关于天使投资, 我们都知道天使投资是权益资本投资的一种形式, 这个名词最早出现在美国, 是指具有一定财富的人对具有巨大发展潜力的高风险的初创企业进行早期的直接投资。当创业者的创业设想只有基本框架但没有资金支持继续发展时, 就需要天使投资方来给予资金支持。天使投资人一般是指拥有多余资本的富豪或者是大企业的高管, 他们可以通过自己的经验去判断投资项目风险, 根据评估结果来选择是否投资。

当今世界美国的天使投资发展迅速, 很多商业巨头都是天使投资人或曾经受到过天使投资的资助。但是在中国, 天使投资还局限在一定的区域, 中国不缺少有钱人, 但是他们并没有天使投资的尝试和创新精神, 这大大阻碍了天使投资活动的拓展。我认为这种现象的原因在于天使投资背后存在很多尚未解决的问题, 例如创业者不守信用, 或者不按照原本的计划进行资金分配利用。基于这样的背景, 我计划设计一个供投资者和创业者使用的 DAPP 以解决现存的问题, 这样可以让中国出现更多好的天使投资项目。

通过该应用, 创业者可以获得他们需要的资金数目, 在创业过程中, 他们需要实时更新当前的创业计划和资金去向, 这些内容和信息都被写在链上, 所以是无法更改的, 一旦有任何不守信的情况都可以根据链上的记录追究责任。当创业者实现盈利且资金到达一定数目时, 他们需要回报投资人, 直接通过区块链进行资金转移。通过这种方式可以节省双方很多的时间, 也避免出现失信现象导致投资的失败。

使用说明

首先要进行环境的搭建:

1. 安装 node.js: <https://nodejs.org/en/download/>
2. 安装 Truffle: 命令行输入命令: `npm install -g truffle`
3. 安装 MetaMask: <https://metamask.io/>

在目录文件夹打开命令行并输入 `truffle.cmd develop` 命令生成十个账户, 默认运行在本地的 9545 端口, 并且可以通过该 `truffle(develop)`进行编译和部署合约。如下图:

```
C:\Users\17980\Desktop\angel_invest>truffle.cmd develop
Truffle Develop started at http://127.0.0.1:9545/

Accounts:
(0) 0x627306090abab3a6e1400e9345bc60c78a8bef57
(1) 0xf17f52151ebef6c7334fad080c5704d7216b732
(2) 0xc5fdf4076b8f3a5357c5e395ab970b5b54098fef
(3) 0x821aea9a577a9b44299b9c15c88cf3087f3b5544
(4) 0x0d1d4e623d10f9fba5db95830f7d3839406c6af2
(5) 0x2932b7a2355d6fecc4b5c0b6bd44cc31df247a2e
(6) 0x2191ef87e392377ec08e7c08eb105ef5448eced5
(7) 0x0f4f2ac550a1b4e2280d04c21cea7ebd822934b5
(8) 0x6330a553fc93768f612722bb8c2ec78ac90b3bbc
(9) 0x5aeda56215b167893e80b4fe645ba6d5bab767de
```

```

Private Keys:
(0) c87509a1c067bbde78beb793e6fa76530b6382a4c0241e5e4a9ec0a0f44dc0d3
(1) ae6ae8e5ccbf04590405997ee2d52d2b330726137b875053c36d94e974d162f
(2) 0dbbe8e4ae425a6d2687f1a7e3ba17bc98c673636790f1b8ad91193c05875ef1
(3) c88b703fb08cbea894b6aeff5a544fb92e78a18e19814cd85da83b71f772aa6c
(4) 388c684f0ba1ef5017716adb5d21a053ea8e90277d0868337519f97bede61418
(5) 659cbb0e2411a44db63778987b1e22153c086a95eb6b18bdf89de078917abc63
(6) 82d052c865f5763aad42add438569276c00d3d88a2d062d36b2bae914d58b8c8
(7) aa3680d5d48a8283413f7a108367c7299ca73f553735860a87b08f39395618b7
(8) 0f62d96d6675f32685bbdb8ac13cda7c23436f63efbb9d07700d8669ff12b7c4
(9) 8d5366123cb560bb606379f90a0bfd4769eccc0557f1b362dcae9012b548b1e5

Mnemonic: candy maple cake sugar pudding cream honey rich smooth crumble sweet treat

aa Important aa : This mnemonic was created for you by Truffle. It is not secure.
Ensure you do not use it on production blockchains, or else you risk losing funds.

truffle(develop)> _

```

另外开一个命令窗口运行服务端，因为项目的部分代码参考了 truffle 官方的 demo，所以直接输入 npm run dev 命令即可运行服务端。如下图：

```

C:\Users\17980\Desktop\angel_invest>npm run dev

> truffle-init-webpack@0.0.2 dev C:\Users\17980\Desktop\angel_invest
> webpack-dev-server

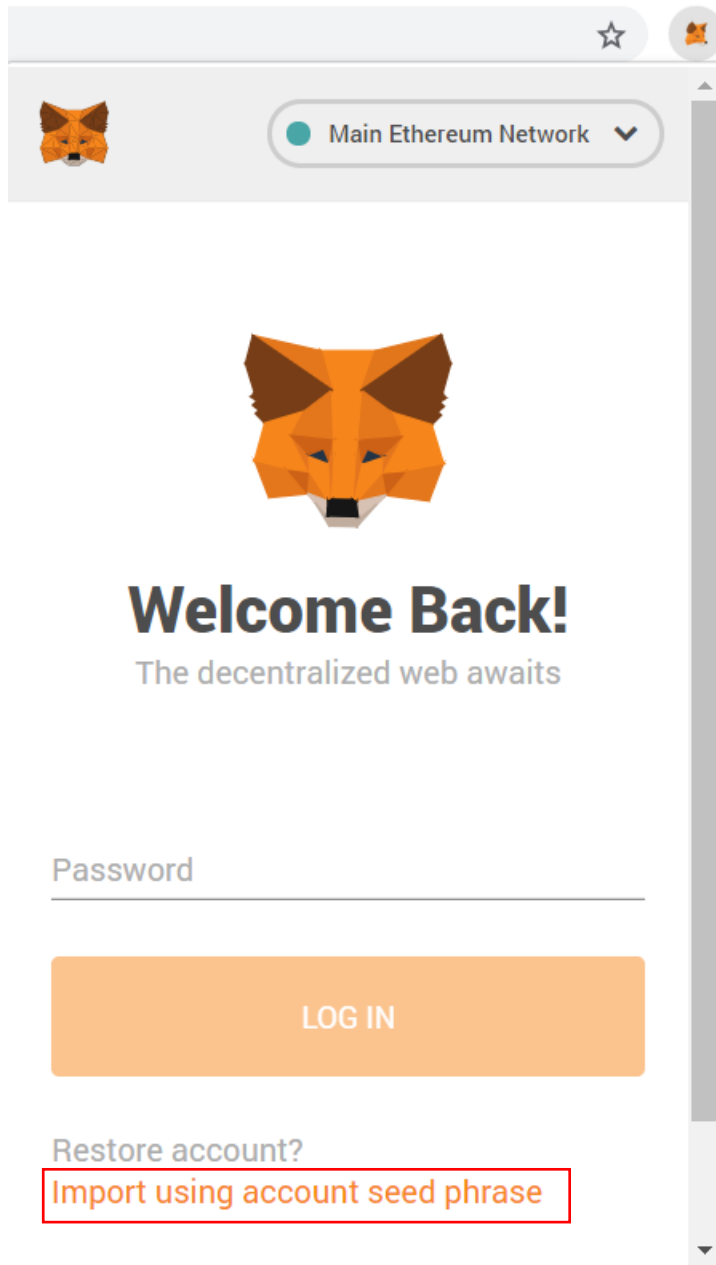
i awdsa: Project is running at http://localhost:8080/
i awdsa: webpack output is served from /
a awdma: Hash: c1b61c5300fdec33f324
Version: webpack 4.28.2
Time: 6435ms
Built at: 2018-12-28 16:29:10
    Asset      Size  Chunks             Chunk Names
  app.js    721 KiB          0  [emitted]  [big]  main
app.js.map 2.61 MiB          0  [emitted]             main
index.html 1.16 KiB          0  [emitted]
Entrypoint main [big] = app.js app.js.map
[31] ./node_modules/url/url.js 22.8 KiB {0} [built]
[48] ./node_modules/events/events.js 8.13 KiB {0} [built]
[96] multi (webpack)-dev-server/client?http://localhost:8080 ./app/scripts/index.js 40 bytes {0} [built]
[97] (webpack)-dev-server/client?http://localhost:8080 7.78 KiB {0} [built]
[103] ./node_modules/strip-ansi/index.js 161 bytes {0} [built]
[105] ./node_modules/loglevel/lib/loglevel.js 7.68 KiB {0} [built]
[106] (webpack)-dev-server/client/socket.js 1.05 KiB {0} [built]
[108] (webpack)-dev-server/client/overlay.js 3.58 KiB {0} [built]
[113] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {0} [built]
[115] (webpack)/hot/emitter.js 75 bytes {0} [built]
[116] ./app/scripts/index.js 4.74 KiB {0} [built]
[117] ./app/styles/app.css 1.18 KiB {0} [built]
[122] ./node_modules/web3/index.js 193 bytes {0} [built]
[200] ./node_modules/truffle-contract/index.js 437 bytes {0} [built]
[309] ./build/contracts/MetaCoin.json 147 KiB {0} [built]
+ 295 hidden modules

```

然后在网页输入 <http://localhost:8080/> 即可进入主界面，但是此时还无法正确显示界面，会出现如下提示，原因是没有设置客户端的端口号，无法运行账户。



前面提到了我使用的以太坊客户端是 Truffle Develop，所以默认端口是 9545，端口需要通过 MetaMask 来设置，使用谷歌浏览器，点击网页顶端的 MetaMask 小图标，如下图：



点击最下方的 Import using account seed phrase，把 Truffle Develop 的 Mnemonic 的单词粘贴到 Wallet Seed 部分，并创建自己的密码。点击 RESTORE 保存：

```
Private Keys:
(0) c87509a1c067bbde78beb793e6fa76530b6382a4c0241e5e4a9ec0a0f44dc0d3
(1) ae6ae8e5ccbf04590405997ee2d52d2b330726137b875053c36d94e974d162f
(2) 0dbbe8e4ae425a6d2687f1a7e3ba17bc98c673636790f1b8ad91193c05875ef1
(3) c88b703fb08cbea894b6aeff5a544fb92e78a18e19814cd85da83b71f772aa6c
(4) 388c684f0ba1ef5017716adb5d21a053ea8e90277d0868337519f97bede61418
(5) 659cbb0e2411a44db63778987b1e22153c086a95eb6b18bdf89de078917abc63
(6) 82d052c865f5763aad42add438569276c00d3d88a2d062d36b2bae914d58b8c8
(7) aa3680d5d48a8283413f7a108367c7299ca73f553735860a87b08f39395618b7
(8) 0f62d96d6675f32685bbdb8ac13cda7c23436f63efbb9d07700d8669ff12b7c4
(9) 8d5366123cb560bb606379f90a0bfd4769eccc0557f1b362dcae9012b548b1e5

Mnemonic: candy maple cake sugar pudding cream honey rich smooth crumble sweet treat
```

Wallet Seed

candy maple cake sugar pudding cream honey rich smooth
crumble sweet treat

New Password (min 8 chars)

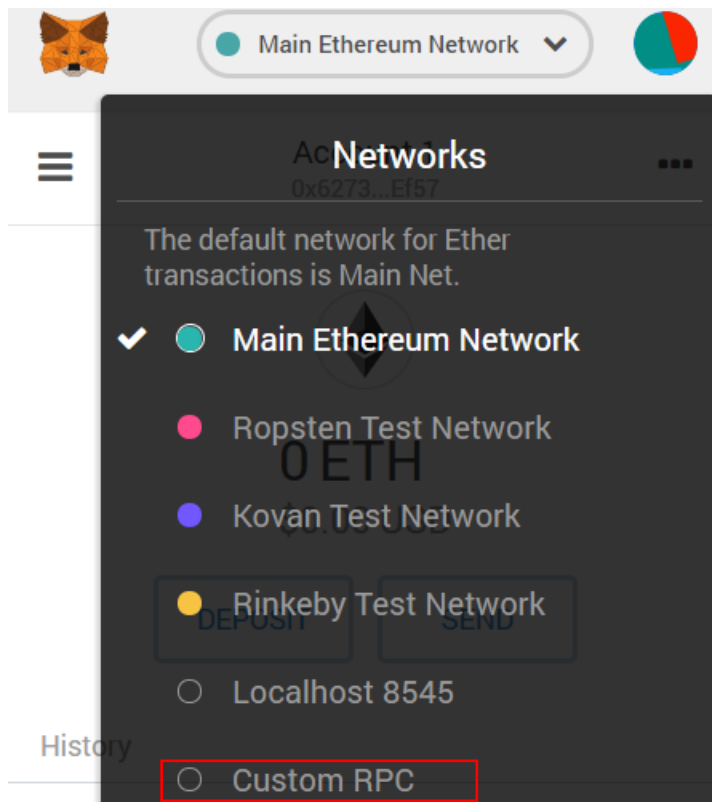
.....

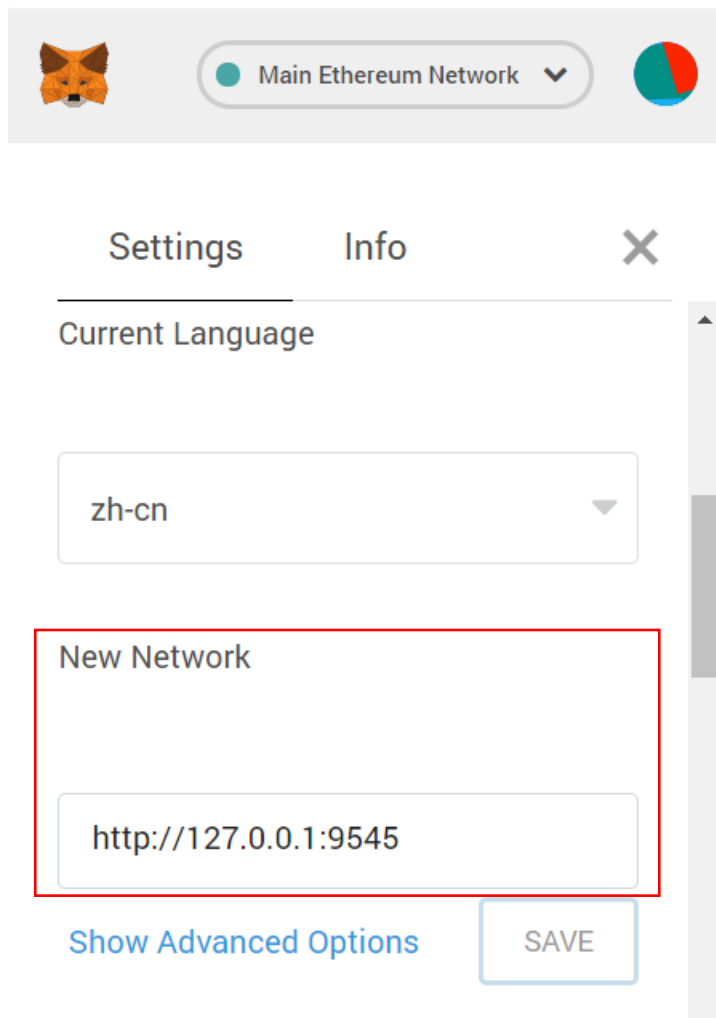
Confirm Password

.....

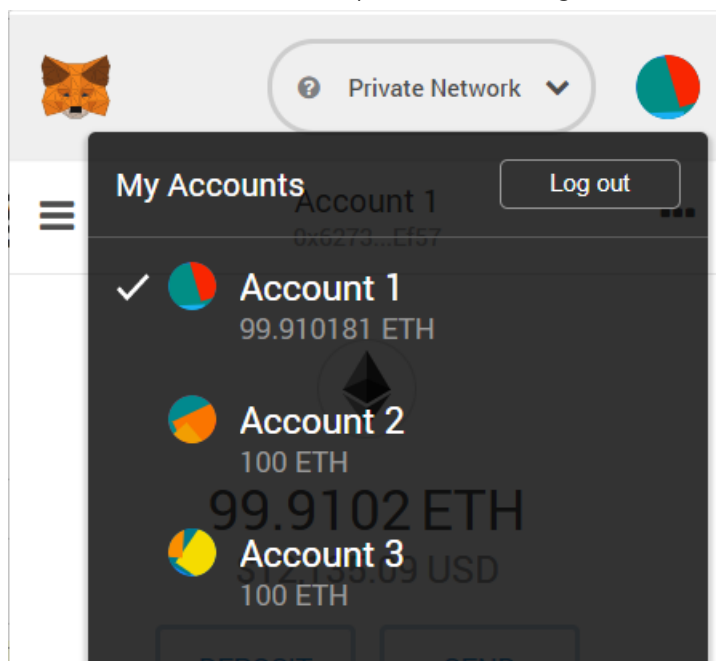
RESTORE

保存之后再打开 MetaMask 小界面选择网络, 点击最下方的 CustomRPC, 在 URL 部分输入:
<http://127.0.0.1:9545> 然后保存,





设置端口后可以在窗口看到随机生成的十个账户，选择任意一个账户，然后进入应用界面就可以正常运行所有的功能了。最后要提示的是，如果完成上述步骤仍然找不到账户，一般解决方法就是在 Truffle Develop 命令行输入 `migrate -reset` 重新部署合约。



测试结果

主界面如下图：



默认情况是 Account1 是投资人, 其账户有 10000 的资金, 所以选择 Account2 作为创业者, 在上面窗口输入目标资金和转账地址。例如输入 Account1 的账户地址, 转账金额为 2000, 在确认之前资金是 0:



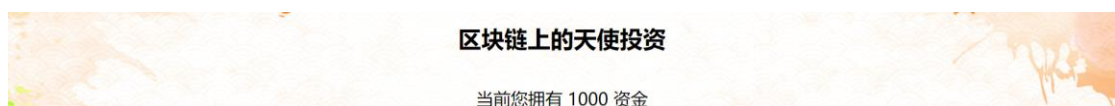
在弹出的窗口点击 CONFIRM 即可确认交易, 如下图, 可见此时账户 Account2 的账户上金额变为 2000 了:



创业者需要在创业过程中把资金去向和项目计划都写在区块链上, 这样可以保证记录不会被篡改。如果创业者有能力回报投资者了, 那么只需要填写回报资金并提交。例如回报资金为 1000, 转账地址不变, 同时填写资金去向:



同样需要 CONFIRM 交易, 在确认交易后 Account2 的账户上只有 1000 了, 而 Account1 的账户上还有 9000, 下图是投资者和创业者的账户余额, 可以看到创业者当前拥有 9000 资金, 创业者只有 1000 资金, 说明两次交易都成功了。



使用 Remix 可以看到区块的详细信息，可以看到项目的计划和资金去向已经都写在了区块中，说明合约部署成功，实现了目标要求。

[block:39 txIndex:0] from:0xf17...6b732 to:MetaCoin.setPlan(string) 0x0a1...b92f5 value:0 wei data:0x4a0...00000 logs:0 hash:0x5b4...b1347

Debug

status	0x01
transaction hash	0x5b4e3485fe8ca8e2feba351b2497cae24e38dd4d814b7da80f35c50a3b8b1347
from	0xf17f52151ebef6c7334fad080c5704d77216b732
to	MetaCoin.setPlan(string) 0x0a143bdf026eabaf95d3e88abb88169674db92f5
gas	86156 gas
transaction cost	86156 gas
hash	0x5b4e3485fe8ca8e2feba351b2497cae24e38dd4d814b7da80f35c50a3b8b1347
input	0x4a0...00000
decoded input	{ "string plan": "部分资金用来招聘人才 未来计划上市" }
decoded output	-
logs	
value	0 wei

综上，测试结果说明合约基本功能全部实现，该合约可以把转账和计划过程简单化，这样很好地解决了创业者和投资者之间的信用问题以及转账问题，对于促进国内的天使投资有非常积极和深远的意义。