

uma_bbnc_project 北斗UMA项目

Hi, welcome to view our implementation of ProjectX!

你好，欢迎了解“北斗UMA - ProjectX项目”！

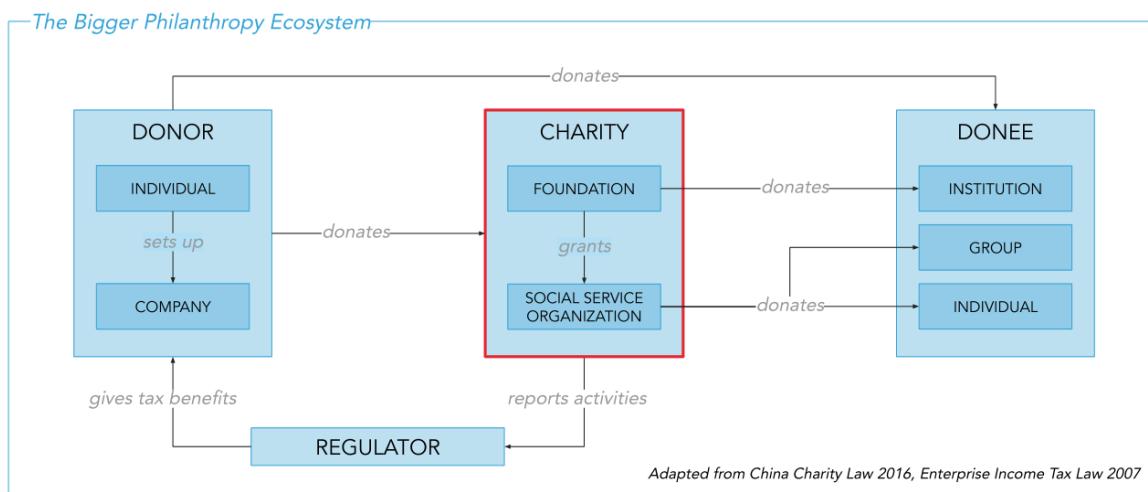
Our main contract: [ProjectX.sol](#).

Our front-end code: [websitedemo/](#).

Our Demo website: <http://donate.confluxcharity.xyz/>.

Conflux Charity 去中心化慈善系统

The traditional operating model of charities is highly prone to corruption, fraud, and misuse of funds; mainly due to the high number of intermediaries and the untraceable nature of cash.



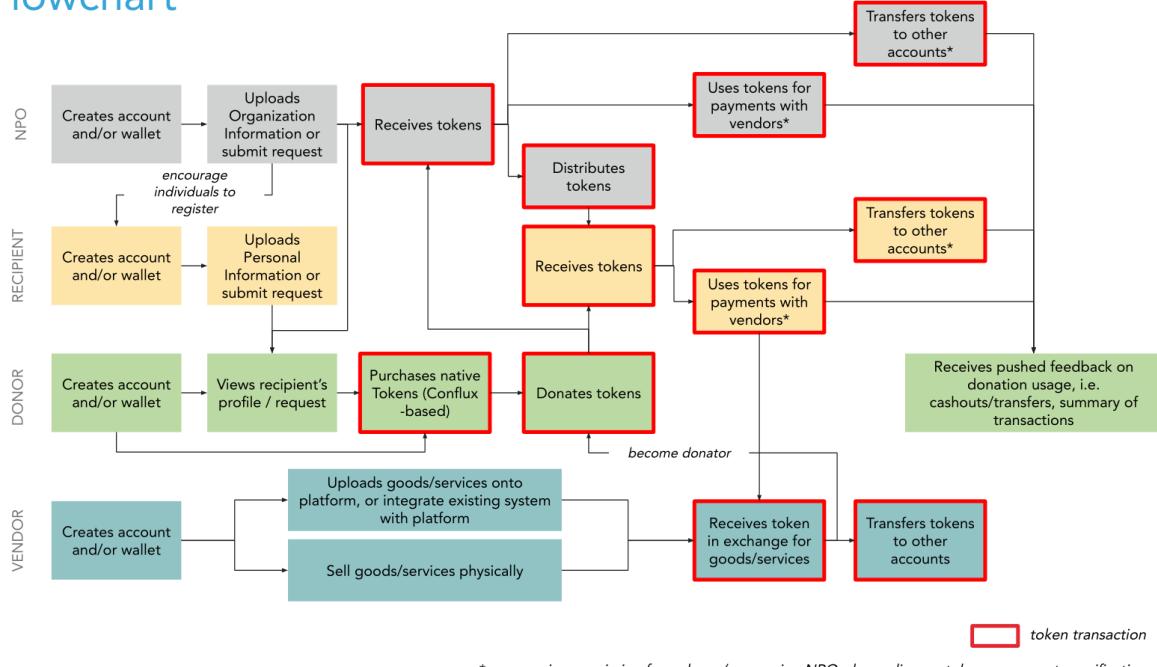
[Project X \(Conflux Charity\)](#) solves this issue as a Conflux dApp platform which:

- 1) ensures transparency over the end-to-end flow of funds;
- 2) provides peer-to-peer donation option between donor and recipient;
- 3) enables customization on funds usability.

Flowchart 流程设计

We designed a new flowchart to support our ideas on decentralized charity as dApp.

Flowchart

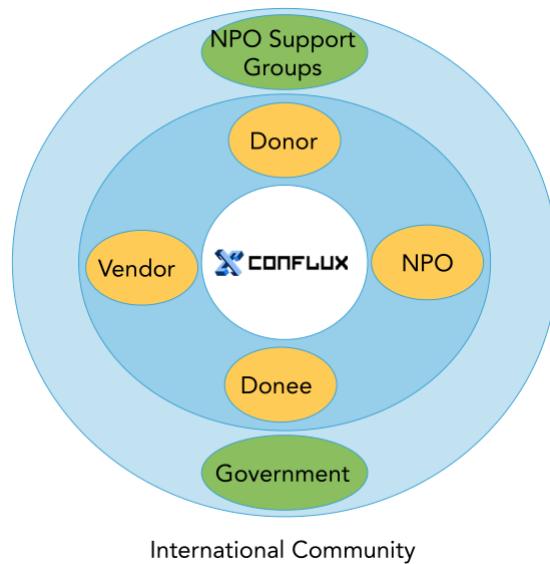


There are four basic role of participants in our initial design:

- **NPO**: Non-profitable Organization
- **Recipient**: personal account for anyone who needs help
- **Donor**: personal or some group account who wants to help others
- **Vendor**: suppliers who supply goods to make sure using the donation is properly

Win-Win for Everyone 理想帕累托改进

Our ultimate goal is to bring a new era on charity, and everyone will benefit from it.



- ▶ **Society** - overall higher social welfare
- ▶ **Donee** - faster, cheaper, and clearer flow of funds
- ▶ **NPO** - increased visibility, efficiency, and accountability
- ▶ **Donor** - higher impact AND returns (more targeted charitable activities, taxation deductions, social credit points)
- ▶ **Vendor** - ESG participation
- ▶ **Government** - strategic partner for sector supervision and administration (incl. taxation & relevant public policies)
- ▶ **Conflux** - ESG, network externalities, business ecosystem

Contract 后端设计

Our main contract: [ProjectX.sol](#).

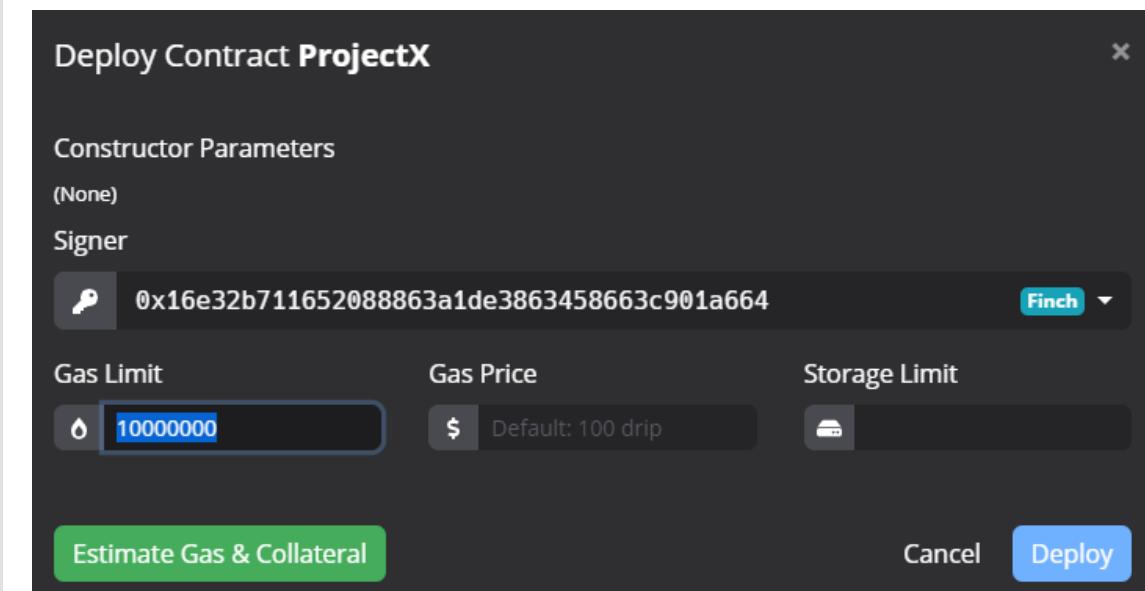
Functionality Overview 功能总览

The code is developed according to the flowchart we designed.

Function	Description
<i>register, describe, validate, stake, quit</i>	user/account/address creation, deletion and validation related functionalities
<i>donate, donate_to,</i>	for transferring funds to non-donors
<i>buy_goods, NPO_distrbute</i>	for buying from vendors and distributing donations
<i>vendor_register_goods, vendor_cancel_goods, vendor_withdraw, vender_withdraw_all, recipient_withdraw, recipient_withdraw_all</i>	vendor and donation recipient related functionalities
<i>sponser_gas, sponser_collateral</i>	handles transaction sponsorship mechanisms
<i>get_admin_donation_pool, get_donation_record, get_Recipient_donation_pool, get_NPO_donation_pool ...</i>	functions the admin uses to get info about the transactions and overall system
<i>set_staking_price, lock/unlock_address</i>	functions admin uses to regulate accounts and the charity system

All the functions supporting is implemented in a single contract.

Notice: Local deployment require manual setting on the gas limit(>10000000).



Coding Details 代码细节

Our main contract:

```
4
5 contract ProjectX {
6     // admin records
7     address project_admin;
8     uint256 admin_donation_pool;
9
10    // Unit informations
11    mapping (address => uint) private types;
12    mapping (address => bool) private validations;
13    mapping (address => bool) private lock;           // admin lock control
14    mapping (address => bytes32) private descriptions; // sha256( description )
15
16    // global records
17    mapping (address => uint256) private staking_fund;
18    mapping (address => uint256) private donation_record;
19    mapping (address => uint256) private NPO_donation_pool;
20    mapping (address => uint256) private Recipient_donation_pool;
21    mapping (address => uint256) private CrowdFund_donation_pool;
22    mapping (address => uint256) private CrowdFund_donation_target;
23    mapping (address => uint256) private Vendor_pool;
24    mapping (uint256 => uint256) private goods_price;   // goods_number to goods_price
25    mapping (uint256 => bytes32) private goods_description;
26    mapping (uint256 => address) private goods_providing_vendor;
27    mapping (uint256 => bool) private goods_valid;
28
29    // type info
30    mapping (string => uint) private type_map;
31    string[] private type_list = ["default", "Donor", "NPO", "Recipient", "CrowdFund", "Vendor"];
32    mapping (string => uint256) private staking_price;
33
34    // initialization
```

Initialization

We set *admin info*, *type info*, *staking info* and *sponsorship info* in the constructor.

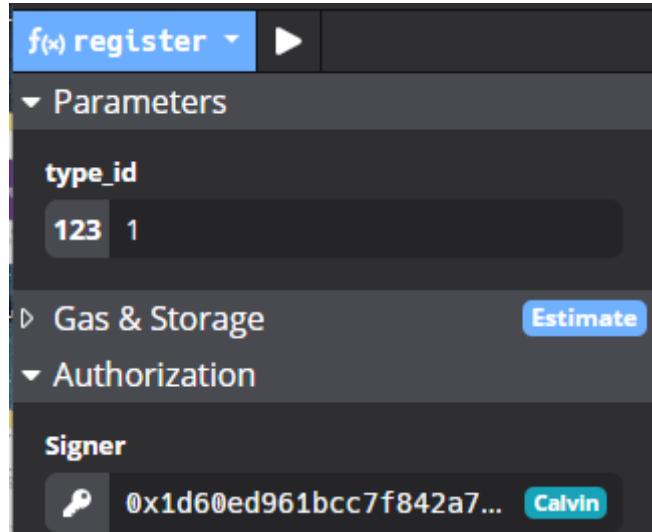
Registration

Our project requires registration and validation before any form of participation.

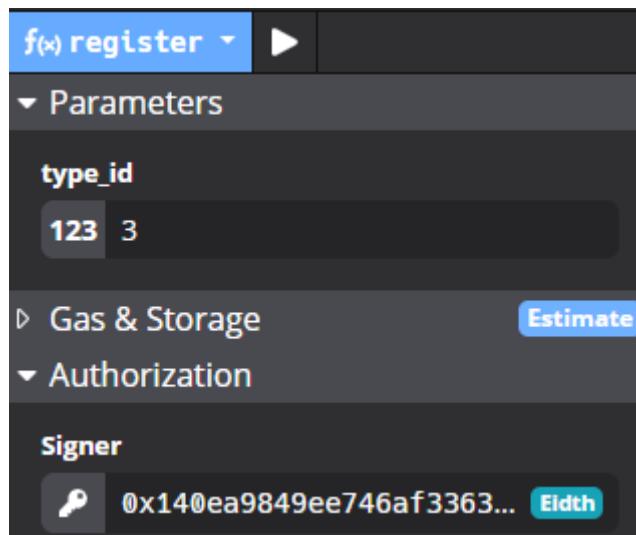
```
type_list = ["default", "Donor", "NPO", "Recipient", "CrowdFund", "Vendor"];
```

Index of the type in the correspond type_id.

Register as a `Donor` (`type_id = 1`):



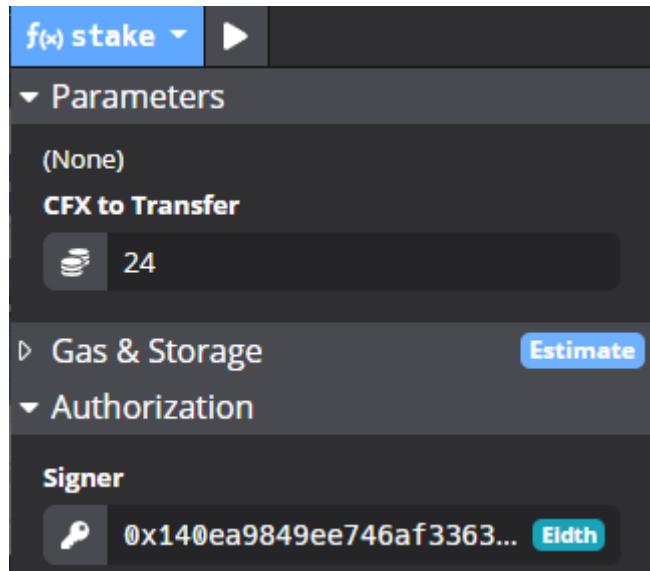
Register as a `Recipient` (`type_id = 3`):



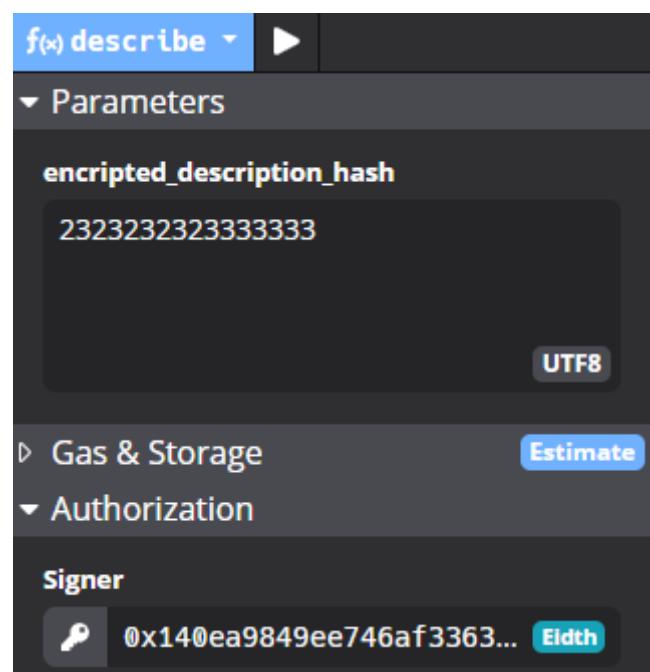
Everyone except Donors need to validate themselves.

First, `stake`.

```
// staking price initializations
staking_price["Recipient"] = 23 * 1e18;
staking_price["NPO"] = 233 * 1e18;
staking_price["Vendor"] = 666 * 1e18;
staking_price["CrowdFund"] = 2333 * 1e18;
```

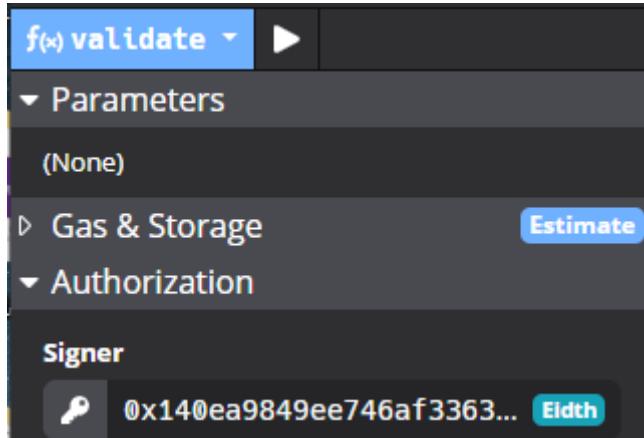


Second, `describe` oneself:



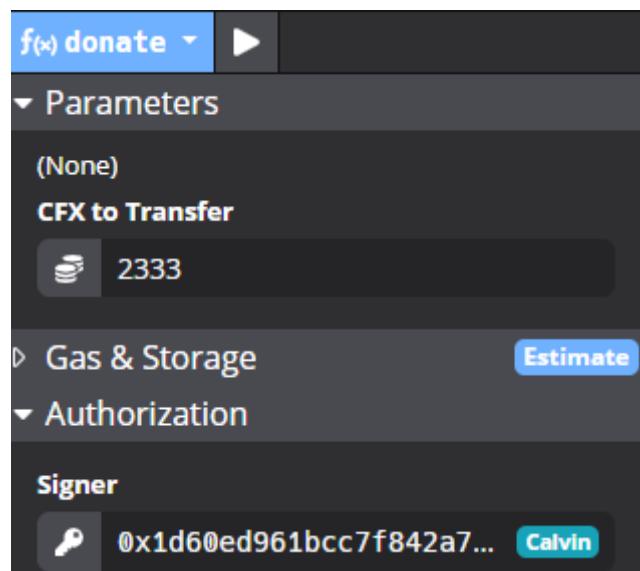
Finally, `validate`:

```
1 // validate an address
2 function validate() public {
3     require(!is_locked(msg.sender), "Locked address, can't be validated!");
4     require(!is_valid(msg.sender), "Only invalid address can be
5         validated!");
6     require(staking_fund[msg.sender] != 0, "Need staking before
7         validation.");
8     require(descriptions[msg.sender] != bytes32(0x0), "Need description
9         before validation.");
10 }
```



Donation

`donate` to the contract pool:



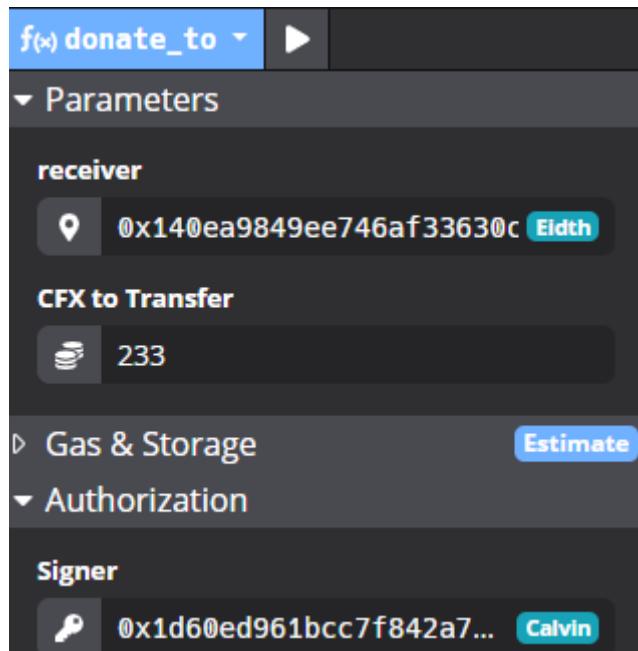
`donate_to` as p2p donation:

```

1 // Donor: p2p donation using CFX
2 function donate_to(address payable receiver) public payable {
3     require(is_valid(msg.sender), "Invalid sender!");
4     require(is_valid(receiver), "Invalid receiver!");
5
6     require(types[receiver] != uint(0x0), "Unregistered receiver!");
7     require(types[msg.sender] == type_map["Donor"], "Only donors can
8     donate!");
9     require(types[receiver] != type_map["Donor"], "Only NOT donors can be
10    donated to!");
11    require(types[receiver] != type_map["Vendor"], "Only NOT vendors can be
12    donated to!");
13
14    require(msg.value > 0, "Invalid donation!");
15
16    // Seperate situations
17    if(types[receiver] == type_map["Recipient"]){
18        // support transferring directly from sender to receiver
19
20        Recipient_donation_pool[receiver] += msg.value;
21        //receiver.transfer(msg.value);
22        donation_record[msg.sender] += msg.value;
23    }
24}
```

```

20     }
21     else if(types[receiver] == type_map["NPO"]){
22         NPO_donation_pool[receiver] += msg.value;
23         donation_record[msg.sender] += msg.value;
24     }
25     else if(types[receiver] == type_map["CrowdFund"]){
26         require(CrowdFund_donation_pool[receiver] <
27 CrowdFund_donation_target[receiver], "Donation target completed!");
28         uint256 lim = CrowdFund_donation_target[receiver] -
29 CrowdFund_donation_pool[receiver];
30         if(msg.value > lim){
31             // refund the overflowed donations
32             msg.sender.transfer(msg.value - lim);
33             CrowdFund_donation_pool[receiver] =
34 CrowdFund_donation_target[receiver];
35             donation_record[msg.sender] += lim;
36         }
37     }
38 }
39 }
```



Consumption

We support goods consumption in the system.

`buy_goods`:

```

1 // Buy all kinds of goods from vendors
2 function buy_goods(uint256 goods_number, uint quantity) public payable {
3     require(is_valid(msg.sender), "Invalid sender!");
4     require(types[msg.sender] == type_map["Recipient"] || types[msg.sender]
5 == type_map["NPO"], "Only recipients and NPOS can buy goods from vendors!");
6     require(goods_valid[goods_number], "Invalid goods.");
7
8     if(types[msg.sender] == type_map["Recipient"]){
9 }
```

```

8     require(Recipient_donation_pool[msg.sender] >=
9         goods_price[goods_number] * quantity, "Not enough tokens.");
10    vendor_pool[ goods_providing_vendor[goods_number] ] += 
11        goods_price[goods_number] * quantity;
12        // // refund the overflows
13        //msg.sender.transfer(msg.value - goods_price[goods_number] *
14        quantity);
15        Recipient_donation_pool[msg.sender] -= goods_price[goods_number] *
16        quantity;
17    } else{
18        require(NPO_donation_pool[msg.sender] >= goods_price[goods_number] *
19        quantity, "Insufficient NPO's donation pool.");
20        vendor_pool[ goods_providing_vendor[goods_number] ] += 
21        goods_price[goods_number] * quantity;
22        NPO_donation_pool[msg.sender] -= goods_price[goods_number] *
23        quantity;
24    }
25
26    // refund misprovided CFXs (if there are any)
27    msg.sender.transfer(msg.value);
28 }
```

Specified Functionality

As we mentioned in the overview.

NPOs can redistribute donations into recipients just like the flowchart has designed.

Vendors shall manage the goods.

The system offer vendors and recipients a way to withdraw their tokens. Because all the donations is running under the contract, and the contract is the only one holding the tokens for now.

We can add safety mechanism through this, for instance we can `lock` one's address to avoid `withdraw` in case bad behaviors happend.

<code>buy_goods, NPO_distrbute</code>	for buying from vendors and distributing donations
<code>vendor_register_goods, vendor_cancel_goods,</code> <code>vendor_withdraw, vender_withdraw_all,</code> <code>recipient_withdraw, recipient_withdraw_all</code>	vendor and donation recipient related functionalities

Sponsorship

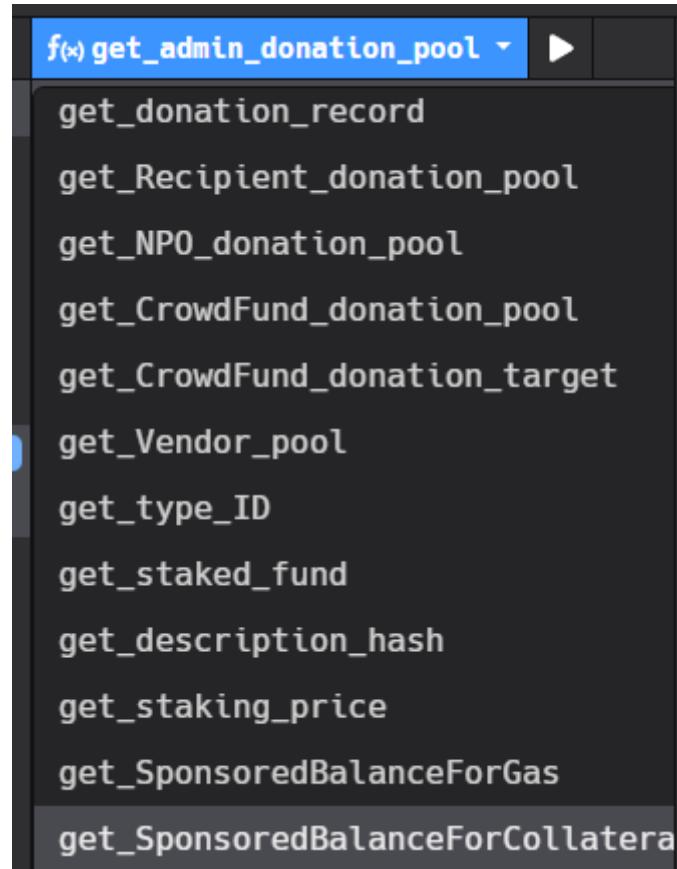
We support sponsorship in the contract.

`sponsor_gas` and `sponsor_collateral`. So the participants won't pay for the transaction.

We might need to add some safety mechanism in the future to avoid attacks like DDoS.

View Functions

There are also some view-only functions to display the system parameters.



```
f(x) get_admin_donation_pool ▾ ►
get_donation_record
get_Recipient_donation_pool
get_NPO_donation_pool
get_CrowdFund_donation_pool
get_CrowdFund_donation_target
get_Vendor_pool
get_type_ID
get_staked_fund
get_description_hash
get_staking_price
get_SponsoredBalanceForGas
get_SponsoredBalanceForCollatera
```

Administration

We enable the administration management in the system for now, until we figured out a better safety insurance design.

```
1 /// admin control
2 /**
3 // Caution: amount's unit is Drip.
4 function set_staking_price(uint type_id, uint256 amount) public {
5     require(msg.sender == project_admin, "Admin permission denied.");
6     require(type_id > 0 && type_id < type_list.length, "Invalid type_id!");
7     staking_price[type_list[type_id]] = amount;
8 }
9 // admin can lock addresses with suspicious behaviors
10 function lock_address(address a) public {
11     require(msg.sender == project_admin, "Admin permission denied.");
12     lock[a] = true;
13     validations[a] = false;
14 }
15 function unlock(address a) public {
16     require(msg.sender == project_admin, "Admin permission denied.");
17     lock[a] = false;
18 }
19 function admin_distribute(address receiver, uint256 amount) public {
20     require(msg.sender == project_admin, "Admin permission denied.");
21     require(is_valid(receiver), "Invalid receiver!");
```

Future Developing 可扩展性

Despite finished all the initial design of our project, we still have many ideas on the developing in the future.

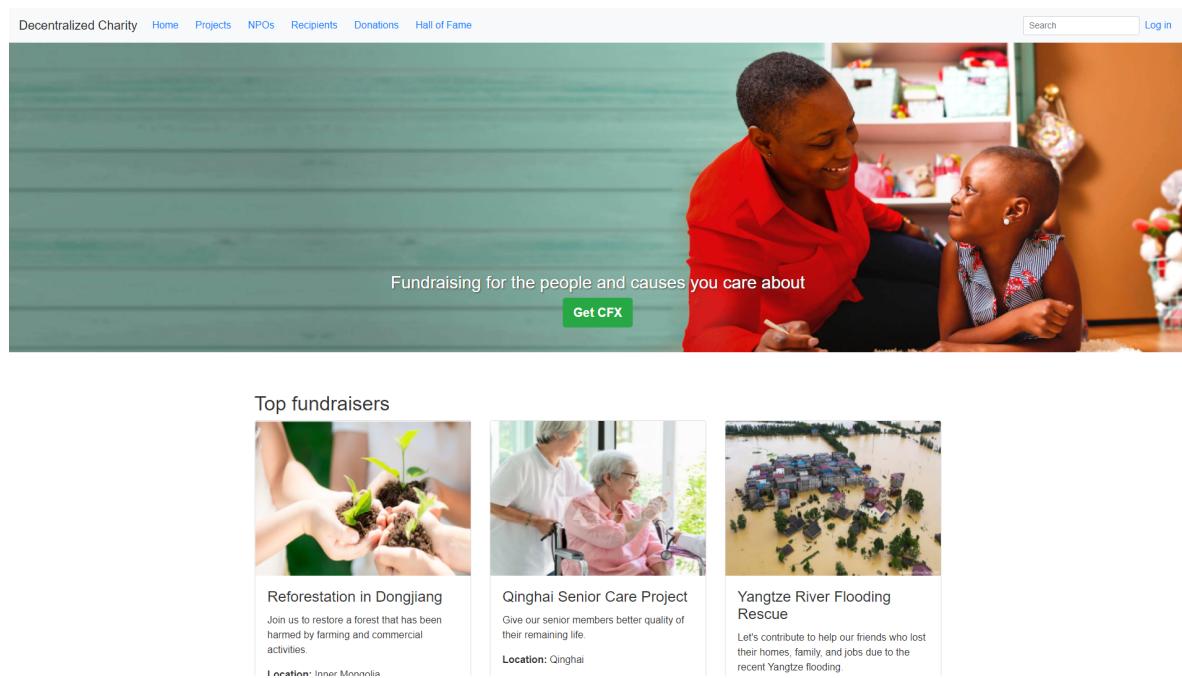
We also supported `Crowdfund` in the contract design because it's becoming more and more trending.

We designed `CouponCoin` for the future, cause we wanna introduce a couple system someday to make sure that specified donation is becoming coupons that can only be used in specified scenario or specified goods.

Website 前端设计

Our front-end code: [websitedemo](#).

Our Demo website: <http://donate.confluxcharity.xyz/>.



Functionality Overview 功能总览

We developed a friendly user interface to enhance our project's accessibility.

Log In Interface

Our project needs users to register and verify themselves both on the website and also on the blockchain.

We shall support full functionality with searching in the future development.

Log In

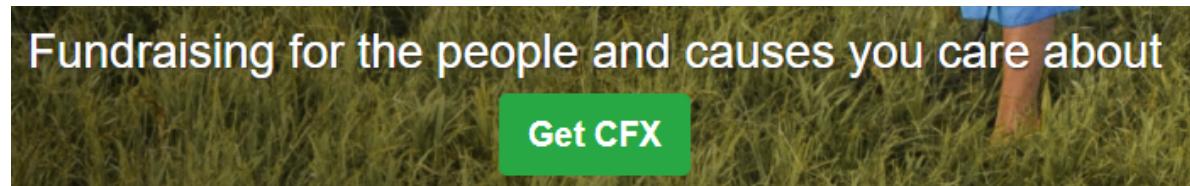
[Sign in to Decentralized Charity](#)

© 2020 Decentralized Charity

Swapping Support Channel

In our initial design, we use a swaping market (providing an URL link as "Get CFX") to supply a channel to those who aren't very familiar with Conflux, and they could get CFXs from the swapping.

This can be easily modified in the future, if we add some moonswap-like mechanism into our system. And through this kind of independant design, we can focus more on the effective part of functionality.



Project Page

We have a `Projects` page to show all the `crowdFund` donation states as we mentioned in the contract design part.

The screenshot shows a grid of five project cards:

- Qinghai Senior Care Project**: Give our senior members better quality of their remaining life.
- Yangtze River Flooding Rescue**: Let's contribute to help our friends who lost their homes, family, and jobs due to the recent Yangtze flooding.
- Reforestation in Dongjiang**: Join us to restore a forest that has been harmed by farming and commercial activities.
- Clean Clothes by P&G**: Give a detergent pack to rural families in China for every 1 yuan you donate.
- Breast Cancer Marathon**: Join our marathon for Breast Cancer Awareness Day!

NPOs Page

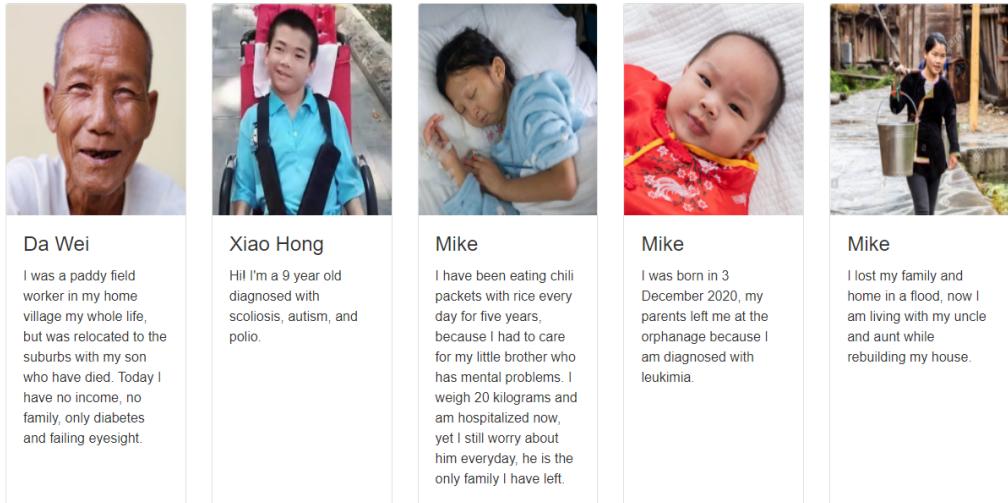
We have a `NPOS` page to show all the `NPO` organizations as we mentioned in the contract design part.

The screenshot shows a grid of five organization cards:

- China Green Carbon Foundation**: First nation-wide non-profit public funding foundation dedicated to combating climate change by increasing carbon sink in China.
- China Primary Health Care Foundation**: Developing Primary Health Care in China.
- China Red Cross Society**: National Red Cross Society in the People's Republic of China.
- China Children and Teenagers' Fund**: We provide education and assistance to our friends diagnosed with autism.
- Blue Sky Rescue Group**: Rescue group focused on disaster relief efforts.

Recipients Page

We have a `Recipients` page to show all the `Recipient` informations as we mentioned in the contract design part.



Donations Page

We have a [Donations](#) page to show all the donation transactions on the blockchain as we mentioned in the contract design part.

Basic info: **Donor Name, Amount, Recipient Type, Date, To, Tx Id.**

The transactions can be sorted by [Amount](#) or [Date](#).

Donor Name	Amount	Recipient Type	Date	To	Tx Id
Geneva	22	Recipient	2020-11-09 02:39:52	0x1084232c04628c147c9e5c6dde1f2b1ce4d81d8d8	E8091EA72BA0E5D0150128B2A4AA058D1F2B55497AA27F3279C2
Jami	13	CrowdFund	2020-10-26 12:14:55	0x10e632b711652088863a1de3863458863c901a664	d768deaa8b7acd43ea1122155140d33604cef1fa22a9f22548cd2872a7654b84
Larsen	11	CrowdFund	2020-10-04 00:36:01	0x10e632b711652088863a1de3863458863c901a664	138A4C381D792DA6C908E94BFF62E970369F138C77A60D8AEEA42
anonymous	2	CrowdFund	2020-09-25 09:00:57	0x10e632b711652088863a1de3863458863c901a664	9EA08DC1193A8497D50820DBFE11B178F43928A6F5EA0432D6D4
anonymous	3	NPO	2020-09-23 23:22:45	0x1d60ed961bcc7f842a77381934d1756502085fc	7F7A5B0C20CEE075FB105FB0648ABBD4C48683E8F608CEAD957B3
Leonid Korobov	6	NPO	2020-08-28 09:06:05	0x1d60ed961bcc7f842a77381934d1756502085fc	0xtaf5c7f3d53569f6c4eeefcb481d52fc1f2a1133d3740ea053c519
CryptoVoin	999	NPO	2020-08-27 02:09:29	0x1d60ed961bcc7f842a77381934d1756502085fc	95BABABCCC2CF8C07E7111807A915ABF510212BD3F2A83281FD
anonymous	88	NPO	2020-08-22 21:41:41	0x1d60ed961bcc7f842a77381934d1756502085fc	4c3b6fa2587bdd6db16486c980cb21a09ef073c4df15bac9e0ce0eca629c7f3
John	100	NPO	2020-11-16 22:37:35	0x1d60ed961bcc7f842a77381934d1756502085fc	7E875BC96056A16C051BE55E67717D7A9CA40281B999F2257FB
John	30	Recipient	2020-11-11 05:09:10	0x1084232c04628c147c9e5c6dde1f2b1ce4d81d8d8	80DC4542EDDD1FDBB157C49A31D3AECC73F0F06ED2F825D84E6AC26
John	22	Recipient	2020-11-09 02:39:52	0x1084232c04628c147c9e5c6dde1f2b1ce4d81d8d8	E8091EA72BA0E5D0150128B2A4AA058D1F2B55497AA27F3279C2
John	13	NPO	2020-10-26 12:14:55	0x1d60ed961bcc7f842a77381934d1756502085fc	d768deaa8b7acd43ea1122155140d33604cef1fa22a9f22548cd2872a7654b84
John	11	NPO	2020-10-04 00:36:01	0x1d60ed961bcc7f842a77381934d1756502085fc	138A4C381D792DA6C908E94BFF62E970369F138C77A60D8AEEA42

© 2020 Decentralized Charity

Hall of Fame

We wanna develop a mechanism that can show statistics on the donations, and there will be a ranking system based on the statistics. And donors or NPOs with huge amount of donations can be broadcast from the [Hall of Fame](#) page to function as an incentive system to encourage the social environment keep growing. This part will be developed in the future.

Local Deployment 本地操作

Install packages

```
1 | cd Project\ x/websitedemo/  
2 | npm install
```

Run

```
1 | npm run serve
```



Top fundraisers



Reforestation in Dongjiang

Join us to restore a forest that has been harmed by farming and commercial activities.

Location: Inner Mongolia

Associated vendors/organization:
China Green Carbon Foundation

¥120,000 raised of ¥400,000



Qinghai Senior Care Project

Give our senior members better quality of their remaining life.

Location: Qinghai

Associated vendors/organization:
Healthcare

¥100000 raised of ¥75,000,000



Yangtze River Flooding Rescue

Let's contribute to help our friends who lost their homes, family, and jobs due to the recent Yangtze flooding.

Location: Jiangxi

Associated vendors/organization:
Blue Sky Rescue Group

¥100000 raised of ¥75,000,000