

USDx 接口说明

History

时间	内容	版本	作者	备注
2019.06.10	添加接口说明	V0.1	Horsen / Snow	
2019.06.11	添加 Rinkeby 合约地址	V0.2	Snow	
2019.06.17	修改函数名称错误	V0.3	Snow	balance ==> Balance
2019.06.18	修正关于 Token 精度的描述	V0.4	Horsen	
2019.09.29	修正合约地址，合约接口描述	V0.5	Snow	

关于 USDx

USDx 是 dForce 团队开发的去中心化指数型稳定币铸币协议，采用以太坊智能合约开发，用户可以访问 <https://usdx.dforce.network> 进行交互，也可以用调用 USDx protocol 合约接口完成交互。

合约接口

DFProtocol 与 DFProtocolView 合约提供了合约调用接口，目前支持以太坊 Mainnet/Rinkeby 环境。

DFProtocol 接口说明

I. deposit(address _tokenId, uint _feeTokenIdx, uint _amount) return (uint)

描述：存入一种成分币，获得一定数量的 USDx；

参数说明：

Input：

- 1) _tokenId(类型: address): 成分币合约地址；
- 2) _feeTokenIdx(类型: uint): 指定手续费扣除的 Token, 0: DF, 其他暂不支持；
- 3) _amount(类型: uint): 存入成分币的数量；

Output：

(类型: uint): 实际获得的 USDx 数量；

II. withdraw(address _tokenId, uint _feeTokenIdx, uint _amount) return (uint)

描述：取回一种已存入但尚未铸成 USDx 的成分币；

参数说明：

Input：

- 1) _tokenId(类型: address): 成分币合约地址；
- 2) _feeTokenIdx(类型: uint): 指定手续费扣除的 Token, 0: DF, 其他暂不支持；
- 3) _amount(类型: uint): 指定取回成分币的数量；

Output：

(类型: uint): 实际取回的成分币数量, 0 表示无可取回数量；

III. destroy(uint _feeTokenIdx, uint _amount)

描述：销毁 USDx，按配方比例获得多种成分币；

参数说明：

Input:

1) `_feeTokenIdx`(类型: `uint`): 指定手续费扣除的 Token, 0: DF, 其他暂不支持;

2) `_amount`(类型: `uint`): 待销毁 USDx 的数量;

Output:

(无)

IV. `claim(uint _feeTokenIdx) return (uint)`

描述：认领已经铸成尚未分发的 USDx（存储在合约中）；

参数说明：

Input:

1) `_feeTokenIdx`(类型: `uint`): 指定手续费扣除的 Token, 0: DF, 其他暂不支持;

Output:

(类型: `uint`): 成功认领 USDx 的数量;

V. `oneClickMinting(uint _feeTokenIdx, uint _amount)`

描述：“一键铸币” - 用户按铸币比例和最小铸币量提供成分币，获得相应数量 USDx；

参数说明：

Input:

1) `_feeTokenIdx`(类型: `uint`): 指定手续费扣除的 Token, 0: DF, 其他暂不支持;

2) `_amount`(类型: `uint`): USDx 铸币数量;

Output:

(无)

DFProtocolView 接口说明

VI. `getUSDxForDeposit(address _tokenId, uint _amount) returns (uint)`

描述：查询存入某种成分币，当前可获取 USDx 的数量；

参数说明:

Input:

- 1) `_tokenId` (类型: `address`): 成分币合约地址;
- 2) `_amount` (类型: `uint`): 存入成分币的数量;

Output:

(类型: `uint`): 获得的 USDx 数量;

VII. `getUserMaxToClaim()` returns (`uint`)

描述: 查询用户当前可认领 USDx 的最大数量;

参数说明:

Input:

(无)

Output:

(类型: `uint`): 可认领的 USDx 数量;

VIII. `getColMaxClaim()` returns (`address[] tokenID, uint[] balance`)

描述: 查询某种成分币当前可认领 USDx 的最大数量;

参数说明:

Input:

(无)

Output:

- 1) `tokenID` (类型: `address` 数组): 成分币地址列表;
- 2) `balance` (类型: `uint` 数组): 成分币对应的可认领 USDx 的数量列表;

IX. `getMintingSection()` returns (`address[] tokenID, uint[] weight`)

描述: 查询当前铸币配方中各成分币的地址和权重;

参数说明:

Input:

(无)

Output:

- 1) tokenID (类型: address 数组): 当前铸币配方的成分币地址列表
- 2) balance (类型: uint 数组): 当前铸币配方的成分币权重列表 (权重既是比例也是最小铸币量);

X. getBurningSection() returns (address[] tokenID, uint[] weight)

描述: 查询当前销毁配方中各成分币的地址和权重;

参数说明:

Input:

(无)

Output:

- 1) tokenID (类型: address 数组): 当前销毁配方的成分币的地址列表;
- 2) balance (类型: uint 数组): 当前销毁配方的成分币权重值列表;

XI. getUserWithdrawBalance() returns (address[] tokenID, uint[] balance)

描述: 查询用户可取回的成分币及数量;

参数说明:

Input:

(无)

Output:

- 1) tokenID (类型: address 数组): 成分币地址列表;
- 2) balance (类型: uint 数组): 与成分币对应的可取回数量;

XII. getPrice(uint _tokenId) return (uint value)

描述: 查询系统中某代币的价格;

参数说明:

Input:

_tokenId(类型: uint): 查询代币的种类, 0: DF, 其它: 暂不支持;

Output:

(类型: uint): 带 18 位精度的美元价格, 如: 123,400,000,000,000,000,000 表示 123.4 美元;

XIII. `getFeeRate(uint _processIdx) return(uint value)`

描述： 查询指定操作的手续费率；

参数说明：

Input：

`_typeID`(类型: `uint`) : 操作类型, 0: deposit, 1:destroy, 2:claim, 3:withdraw
(目前仅支持 `destroy` 有手续费) ;

Output：

(类型: `uint`): 万分位的手续费率, 如: 10 表示手续费费率是万分之十 (千分之一) ;

XIV. `getDestroyThreshold() returns (uint)`

描述： 查询 `destroy` 操作支持的最小销毁量。

参数说明：

Input：

(无)

Output：

(类型: `uint`): 最小销毁量;

特殊说明

1) 根据 Ethereum 中关于 ERC-20 Token (以下简称 Token)精度的约定, Token 数量中的小数部分以 decimal 位的整数表示, 例如 decimal = 18, 则 211.33 USDx 的表示方法是: 211,330,000,000,000,000; 如 decimal = 6, 则 116.654 USDC 的表示方法是 116,654,000, 以此类推。

2) 用户在存入某种成分币时, 首先需要调用该成分币 ERC-20 合约中的 Approve 函数, 授权给 DFPool 合约, 例如:

```
poolAddr = 0xA0CC...b071;

tokenAddr = 0xf494...2db4;

function approveToken() {

    tokenAddr.approve.sendTransaction(

        poolAddr,

        1000000000 * 1000000000000000000, {

            gas: 1000000

        }

    );

}
```

3) 一键铸币功能需要首先获得所有成分币的授权, 授权给 DFPool 合约;

4) 销毁 USDx 时, 用户需要分别调用 USDx 和 DF 的 Approve 函数, 授权给 DFEngine 合约;

合约地址

Mainnet	Contract Address
DAI	0x89d24a6b4ccb1b6faa2625fe562bdd9a23260359
PAX	0x8e870d67f660d95d5be530380d0ec0bd388289e1
TUSD	0x00000000000085d4780B73119b644AE5ecd22b376
USDC	0xa0b86991c6218b36c1d19d4a2e9eb0ce3606eb48
DF	0x431ad2ff6a9c365805ebad47ee021148d6f7dbe0
USDx	0xeb269732ab75a6fd61ea60b06fe994cd32a83549
DFProtocol	0x5843f1ccc5baa448528eb0e8bc567cda7ed1a1e8
DFProtocolView	0x1351e109fb51888d8e9dd6b364297998b36bf55e
DFEngine	0x8dc753d30b9f9f86dac3380e37d5bc03d6b60202
DFPool	0x786bf554473f9ab733fd683c528212492a23d895

Rinkeby	Contract Address
DAI	0x506243424a778382f73bf6c24390d08fa9096092
PAX	0xd414e78d5db39e90c704070943e067ffc0eb3d86
TUSD	0xfeb2112e370091f25a2f96fb600484700a0ed603
USDC	0x71abccd90dbb09c37686e4d5026c2d9597d469cb
DF	0x5d378961e9d31c0ee394d34741fa1a18144f6fb5
USDx	0xdbcff49d5f48ddf6e6df1f2c9b96e1fc0f31371
DFProtocol	0x7ca26e42bba5543cf5a06a0d554a0fc607f3d4ac
DFProtocolView	0xaf47a307ea7d5dc7133b309c1a76c6a3f4ae1618
DFEngine	0x03463c0ac738a370ecd8c4e70a9be183255f25d
DFPool	0xccf31dc9dcb6cb788d3c6b64f73efedfb7e9f20b