How to perform liquidation on lendf.me

About lendf.me

（lendf.me introduction）
（github link）

Contract Address

MainNet:
moneyMarket : 0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea
liquidator: 0x45b1953611da41f841def7dceff318f83409739d
USDX: 0xeb269732ab75a6fd61ea60b06fe994cd32a83549 (https://dforce.network)
WETH: 0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2 (https://weth.io/)
USDT: 0xdac17f958d2ee523a2206206994597c13d831ec7 (https://tether.to/)
PriceOracle: 0xe8a616fd9d7e82cfcaef3f8a90c6a7eea97e0856
PriceProxy: 0x21f2a370d02996bc914e1b92160d47d279d9f15a

MainNet Test:
MoneyMarket: 0xeda3849869fd560b49dab8c110be3a020f46c79e
Liquidator: 0xea922ec85171590c99373355e2d02de3c5bc2112
NCStandard: 0xafbab35f485ebd7a54685aa6c4e80dde3ff8c84d
NCStableCoin: 0x8be69acc4e4529ba0cfbf9203a3c751371b07717
USDx(allocateTo): 0x3a9e75afcffcd89613037989ea0ed6cec44a4353(UUDD)
WETH(allocateTo): 0x06a1cd567e61b7edda49c30d3d32e60f607fd646
USDT: 0x622B859cf4f1013642F6c177bA713d482fF5b483
Oracle: 0x5b4e06e712c851454b071a755B61fc0a86A8680E
proxyOracle: 0x55d19f7F257544ef2E5c5f9B6d31a2416f8d5146

Rinkeby:
moneyMarket : 0x2fd380b99e0c6ff16f569fb8214b40509f776764
liquidator: 0x8e8627fc8e2359fdc9492540d646c3f4d979a44a
USDX: 0xaf21bb8ae7b7a5eec37964e478583cd486fd12e2
WETH: 0xc8b1a5ef2e19937dd6c0f804df2e3efe9f093b1e
USDT: 0xa1e525f7d24d7ccb78a070bbd12c0bf21fb4a848
PriceOracle: 0xfdc55f9ab320819b8d4a91f50d1e78809b09eb3d
PriceProxy: 0xe80151d6bc910c1edbb37d7ea897c6bb2428b16c

[注:]此版合约经过改动，为了适应 USDx 的合约。改动位于 Liquidator.sol(line:2796)：

```
function tokenAllowAll(address asset, address allowee) internal {

    EIP20Interface token = EIP20Interface(asset);

    if (token.allowance(address(this), allowee) != uint(-1))

        require(token.approve(allowee, uint(-1)),

"FAILED_LIQUIDATE_ASSET_ALLOWANCE_FAILED");

}
```

Liquidation

Step1: get account list

在 lendf.me 智能合约中，用户 account 及相关信息以 mapping 形式存储，无法直接获得。为此，我们提供了一个 URL 链接（https://api.lendf.me/api/v1/account?page=1），以获得所有账户信息。访问上述链接，返回以下格式的 json 数据：

```
{
    "accounts":[
        {
            "address":"0xcd63d37a4b28b55932611c7ab0c35ce63d729341",   //账户地址
            "total_supply_weth":90.239874928734,     //换算成 WETH 的抵押资产总量
            "total_borrow_weth":80.904234234,           //换算成 WETH 的借贷资产总量
            "shortfall_weth":3.223094823,                   //换算成 WETH 的资产缺口总量
            "borrow":[        //借贷资产明细
                {
                    "asset":"0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2",          //借贷资产合约地址
                    "amount":10.98739574          //借贷数量（含利息）
                },
                {
                    "asset":"0xeb269732ab75a6fd61ea60b06fe994cd32a83549",        //借贷资产合约地址
                    "amount":800.904234234          //借贷数量（含利息）
                },
               {
                  "asset":"0xdac17f958d2ee523a2206206994597c13d831ec7 ",        //借贷资产合约地址
                  "amount":0.00         //借贷数量（含利息）
                }
            ],
            "supply":[       //抵押资产明细
                {
                    "asset":"0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2",        //抵押资产合约地址
                    "amount":80.904234234        //抵押数量（含利息）
                },
                {
                    "asset":"0xeb269732ab75a6fd61ea60b06fe994cd32a83549",        //抵押资产合约地址
                    "amount":100.98739574        //抵押数量（含利息）
                },
               {
                  "asset":"0xdac17f958d2ee523a2206206994597c13d831ec7 ",        //借贷资产合约地址
                  "amount":0.00         //借贷数量（含利息）
                }
            ]
        }
    ],
    "request":{
        "block_number":8353527,     //当前区块高度
        "page_size":10,                    //当前页含有的条数(默认值:50)
        "current_page_number":1,   //当前数据页码
        "total_size":100                  //账户数据总条数
        "total_page_number":2,       //数据总页数
    }
}
```

account 中的数据对象将按照 shortfall_weth 值总大到小排列，排在第一位的是当前资产缺口最大的账户。当 shortfall_weth 的数量大于 0 时，说明该账户可以被清算。此时，需要根据该账户的抵押和借贷情况确定如何实施清算，以及是否能够通过清算获利。

Step2: call the contract function liquidateBorrow() to liquidate an account

liquidateBorrow()函数在 Liquidator.sol 合约中，定义如下：

```
function liquidateBorrow(
        address targetAccount,        //要清算的目标账户
        address assetBorrow,        //目标账户借贷资产合约地址
        address assetCollateral,       //目标账户抵押资产合约地址
        uint requestedAmountClose       //请求清算的数量。若为-1，则按照可清算最大值处理
        )
```

方式 1：在 remix(https://remix.ethereum.org)中，实例化 Liquidator.sol 合约并调用上述接口函数

方式 2：通过 web3 库(https://github.com/ethereum/web3.js/) 直接调用合约的接口函数，支持多种脚本语言

方式 3：访问 https://liquidate.lendf.me，在页面中实现目标账户清算，需要安装 chrome 浏览器并安装 MetaMask 钱包插件

Step3: check the liquidateBorrow() return value

liquidateBorrow()函数中包含许多条件判断。在一些基础条件不满足的情况下，函数调用会直接失败，而在另外一些情况下，虽然函数可以执行完成，但没有成功清算目标账户，此时会在合约函数的返回值中给出错误序号。

若函数执行完成并成功清算目标账户，liquidateBorrow()函数返回一个数 0。
若函数执行完成, 但没有清算目标账户, liquidateBorrow()函数返回三个数 uint(err), uint(info), 0。

err 枚举定义如下：

```
enum Error {
NO_ERROR,
OPAQUE_ERROR, 
UNAUTHORIZED,
INTEGER_OVERFLOW,
INTEGER_UNDERFLOW,
DIVISION_BY_ZERO,
BAD_INPUT,
TOKEN_INSUFFICIENT_ALLOWANCE,
TOKEN_INSUFFICIENT_BALANCE,
TOKEN_TRANSFER_FAILED,
MARKET_NOT_SUPPORTED,
SUPPLY_RATE_CALCULATION_FAILED,
BORROW_RATE_CALCULATION_FAILED,
TOKEN_INSUFFICIENT_CASH,
TOKEN_TRANSFER_OUT_FAILED,
INSUFFICIENT_LIQUIDITY,
INSUFFICIENT_BALANCE,
INVALID_COLLATERAL_RATIO,
MISSING_ASSET_PRICE,
EQUITY_INSUFFICIENT_BALANCE,
INVALID_CLOSE_AMOUNT_REQUESTED,
ASSET_NOT_PRICED,
INVALID_LIQUIDATION_DISCOUNT,
INVALID_COMBINED_RISK_PARAMETERS,
ZERO_ORACLE_ADDRESS,
CONTRACT_PAUSED
}
```

info 枚举定义如下：

```
enum FailureInfo {
ACCEPT_ADMIN_PENDING_ADMIN_CHECK,
BORROW_ACCOUNT_LIQUIDITY_CALCULATION_FAILED,
BORROW_ACCOUNT_SHORTFALL_PRESENT,
BORROW_ACCUMULATED_BALANCE_CALCULATION_FAILED,
BORROW_AMOUNT_LIQUIDITY_SHORTFALL,
BORROW_AMOUNT_VALUE_CALCULATION_FAILED,
BORROW_CONTRACT_PAUSED,
BORROW_MARKET_NOT_SUPPORTED,
BORROW_NEW_BORROW_INDEX_CALCULATION_FAILED,
BORROW_NEW_BORROW_RATE_CALCULATION_FAILED,
BORROW_NEW_SUPPLY_INDEX_CALCULATION_FAILED,
BORROW_NEW_SUPPLY_RATE_CALCULATION_FAILED,
BORROW_NEW_TOTAL_BALANCE_CALCULATION_FAILED,
BORROW_NEW_TOTAL_BORROW_CALCULATION_FAILED,
BORROW_NEW_TOTAL_CASH_CALCULATION_FAILED,
BORROW_ORIGINATION_FEE_CALCULATION_FAILED,
BORROW_TRANSFER_OUT_FAILED,
EQUITY_WITHDRAWAL_AMOUNT_VALIDATION,
EQUITY_WITHDRAWAL_CALCULATE_EQUITY,
EQUITY_WITHDRAWAL_MODEL_OWNER_CHECK,
EQUITY_WITHDRAWAL_TRANSFER_OUT_FAILED,
LIQUIDATE_ACCUMULATED_BORROW_BALANCE_CALCULATION_FAILED,
LIQUIDATE_ACCUMULATED_SUPPLY_BALANCE_CALCULATION_FAILED_BORROWER_COLLATERAL_AS
SET,
LIQUIDATE_ACCUMULATED_SUPPLY_BALANCE_CALCULATION_FAILED_LIQUIDATOR_COLLATERAL_AS
SET,
LIQUIDATE_AMOUNT_SEIZE_CALCULATION_FAILED,
LIQUIDATE_BORROW_DENOMINATED_COLLATERAL_CALCULATION_FAILED,
LIQUIDATE_CLOSE_AMOUNT_TOO_HIGH,
LIQUIDATE_CONTRACT_PAUSED,
LIQUIDATE_DISCOUNTED_REPAY_TO_EVEN_AMOUNT_CALCULATION_FAILED,
LIQUIDATE_NEW_BORROW_INDEX_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_BORROW_INDEX_CALCULATION_FAILED_COLLATERAL_ASSET,
LIQUIDATE_NEW_BORROW_RATE_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_SUPPLY_INDEX_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_SUPPLY_INDEX_CALCULATION_FAILED_COLLATERAL_ASSET,
LIQUIDATE_NEW_SUPPLY_RATE_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_TOTAL_BORROW_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_TOTAL_CASH_CALCULATION_FAILED_BORROWED_ASSET,
LIQUIDATE_NEW_TOTAL_SUPPLY_BALANCE_CALCULATION_FAILED_BORROWER_COLLATERAL_ASSET,
LIQUIDATE_NEW_TOTAL_SUPPLY_BALANCE_CALCULATION_FAILED_LIQUIDATOR_COLLATERAL_ASSET
,
LIQUIDATE_FETCH_ASSET_PRICE_FAILED,
LIQUIDATE_TRANSFER_IN_FAILED,
LIQUIDATE_TRANSFER_IN_NOT_POSSIBLE,
REPAY_BORROW_ACCUMULATED_BALANCE_CALCULATION_FAILED,
REPAY_BORROW_CONTRACT_PAUSED,
REPAY_BORROW_NEW_BORROW_INDEX_CALCULATION_FAILED,
REPAY_BORROW_NEW_BORROW_RATE_CALCULATION_FAILED,
REPAY_BORROW_NEW_SUPPLY_INDEX_CALCULATION_FAILED,
REPAY_BORROW_NEW_SUPPLY_RATE_CALCULATION_FAILED,
REPAY_BORROW_NEW_TOTAL_BALANCE_CALCULATION_FAILED,
REPAY_BORROW_NEW_TOTAL_BORROW_CALCULATION_FAILED,
REPAY_BORROW_NEW_TOTAL_CASH_CALCULATION_FAILED,
REPAY_BORROW_TRANSFER_IN_FAILED,
REPAY_BORROW_TRANSFER_IN_NOT_POSSIBLE,
SET_ASSET_PRICE_CHECK_ORACLE,
SET_MARKET_INTEREST_RATE_MODEL_OWNER_CHECK,
SET_ORACLE_OWNER_CHECK,
SET_ORIGINATION_FEE_OWNER_CHECK,
SET_PAUSED_OWNER_CHECK,
SET_PENDING_ADMIN_OWNER_CHECK,
SET_RISK_PARAMETERS_OWNER_CHECK,
SET_RISK_PARAMETERS_VALIDATION,
SUPPLY_ACCUMULATED_BALANCE_CALCULATION_FAILED,
SUPPLY_CONTRACT_PAUSED,
SUPPLY_MARKET_NOT_SUPPORTED,
SUPPLY_NEW_BORROW_INDEX_CALCULATION_FAILED,
SUPPLY_NEW_BORROW_RATE_CALCULATION_FAILED,
SUPPLY_NEW_SUPPLY_INDEX_CALCULATION_FAILED,
SUPPLY_NEW_SUPPLY_RATE_CALCULATION_FAILED,
SUPPLY_NEW_TOTAL_BALANCE_CALCULATION_FAILED,
SUPPLY_NEW_TOTAL_CASH_CALCULATION_FAILED,
SUPPLY_NEW_TOTAL_SUPPLY_CALCULATION_FAILED,
SUPPLY_TRANSFER_IN_FAILED,
SUPPLY_TRANSFER_IN_NOT_POSSIBLE,
SUPPORT_MARKET_FETCH_PRICE_FAILED,
SUPPORT_MARKET_OWNER_CHECK,
SUPPORT_MARKET_PRICE_CHECK,
SUSPEND_MARKET_OWNER_CHECK,
WITHDRAW_ACCOUNT_LIQUIDITY_CALCULATION_FAILED,
WITHDRAW_ACCOUNT_SHORTFALL_PRESENT,
WITHDRAW_ACCUMULATED_BALANCE_CALCULATION_FAILED,
WITHDRAW_AMOUNT_LIQUIDITY_SHORTFALL,
WITHDRAW_AMOUNT_VALUE_CALCULATION_FAILED,
WITHDRAW_CAPACITY_CALCULATION_FAILED,
WITHDRAW_CONTRACT_PAUSED,
WITHDRAW_NEW_BORROW_INDEX_CALCULATION_FAILED,
WITHDRAW_NEW_BORROW_RATE_CALCULATION_FAILED,
WITHDRAW_NEW_SUPPLY_INDEX_CALCULATION_FAILED,
WITHDRAW_NEW_SUPPLY_RATE_CALCULATION_FAILED,
WITHDRAW_NEW_TOTAL_BALANCE_CALCULATION_FAILED,
WITHDRAW_NEW_TOTAL_SUPPLY_CALCULATION_FAILED,
WITHDRAW_TRANSFER_OUT_FAILED,
WITHDRAW_TRANSFER_OUT_NOT_POSSIBLE
}
```