How to integrate LENDF.ME

✓ How to

1. APPROVE

Before supplying asset to MoneyMarket contract, one should call approve function of the asset to the MoneyMarket contract, allow the contract to transfer user's balance under the permission of user. For example, "approve USDT":

contract(USDT).approve(0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea, (uint256)-1); //uint256 -1 represents maximal the user's balance, one can decide how much you want to approve to the contract.

2. SUPPLY

Supply asset to the MoneyMarket contract，for example "supply 100.01 USDT":

*contract(MoneyMarket).supply(0xdAC17F958D2ee523a2206206994597C13D831ec7, 100.01 * 10**6);* //USDT's balance is 6-digit precision integer, and only integer value is valid.

3. WITHDRAW

Withdraw asset from the MoneyMarket contract, for example "withdraw 100.01 USDT":

*contract(MoneyMarket).withdraw(0xdAC17F958D2ee523a2206206994597C13D831ec7, 100.01 * 10**6);*

4. GET SUPPLY BALANCE

To get supplied asset amount of account,

contract(MoneyMarket).getSupplyBalance(userAddress, 0xdAC17F958D2ee523a2206206994597C13D831ec7); //returns total amount of supplied asset(including interests).

5. BORROW

Borrow asset from the MoneyMarket contract，for example “borrow100.01 USDT”:

*contract(MoneyMarket).borrow(0xdAC17F958D2ee523a2206206994597C13D831ec7, 100.01 * 10**6);*

Notes:

a) Borrowed balance should be less than “supplied balance/collateralRatio” (collateralRatio = 125%, which is a public variable of MoneyMarket contract);

b) A fee will be charged when borrowing asset, calculate as “*borrow_amount * originationFee*”, and the fee is counting to total borrow balance immediately (originationFee = 0.0005, which is a public variable of MoneyMarket contract);

6. REPAYBORROW

Repay borrows to the MoneyMarket contract, for example “repay 100.01 USDT”:

*contract(MoneyMarket).repayborrow(0xdAC17F958D2ee523a2206206994597C13D831ec7, 100.01 * 10**6);*

7. GET REALTIME INTEREST RATE

First, get supplyRateMantissa/borrowRateMantissa from MoneyMarket contract,

contract(MoneyMarket).markets(0xdAC17F958D2ee523a2206206994597C13D831ec7);

Then,

*total_block_per_year= 3600 * 24 * 365 / 15;*

*supply APR = supplyRateMantissa * total_block_per_year / 10**18;* //supplyRateMantissa supplying rate in one block cycle, integer value with 18-digit precision;

*borrow APR = borrowRateMantissa * total_block_per_year / 10**18;* //borrowRateMantissa borrowing rate in one block cycle, integer value with 18-digit precision;

✓ MoneyMarket Interfaces:

Function Name	Inputs	Outputs	Remarks
supply(address, uint) returns (uint)	address asset: address of asset; uint amount: amount to supply;	uint error: 0 for success, not 0 for error code;	supply asset
withdraw(address, uint) returns (uint)	address asset: address of asset; uint requestedAmount: amount to withdraw (-1 represents max);	uint error: 0 for success, not 0 for error code;	withdraw supplying asset
borrow(address, uint) returns (uint)	address asset: address of asset; uint amount: amount to	uint error: 0 for success, not 0 for error code;	borrow asset

	borrow;		
repayBorrow(address, uint) returns (uint)	address asset: address of asset; uint amount: amount to repay (-1 represents max);	uint error: 0 for success, not 0 for error code;	repay borrowed asset
calculateAccountValues(address) returns (uint, uint, uint)	address account: address of account;	uint error: 0 for success, not 0 for error code; uint total_supplies: total balance of supplied asset; uint total_borrows: total balance of borrowed asset;	get balances of total borrowed and supplied denominated in WETH with 36-digit precision integer (WETH price is 18-digit precision integer)
getSupplyBalance(address, address) returns (uint)	address account: address of account; address asset: address of asset;	uint amount: amount of asset;	get supplied amount of asset, integer precision is depended on asset's decimal
getBorrowBalance(address, address) returns (uint)	address account: address of account; address asset: address of	uint amount: amount of asset;	get borrowed amount of asset, integer precision is depended on asset's

	asset;		decimal
markets(address) returns (bool, uint, address, uint, uint, uint, uint, uint, uint)	address asset: address of asset;	bool isSupported: if asset is activated in current market; uint blockNumber: current block number of Ethereum; address interestRateModel: interest mode address of asset; uint totalSupply: total amount of asset supplied; uint supplyRateMantissa: supply interest rate in one block cycle. uint supplyIndex: index of supply interest; uint totalBorrows: total amount of asset borrowed; uint borrowRateMantissa: borrow interest rate in one block cycle. uint borrowIndex: index of borrow interest;	market data for asset.

✓ Mainnet Contract Addresses:

Contract Name	Address
MoneyMarket	0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea
USDx	0xeb269732ab75a6fd61ea60b06fe994cd32a83549
USDT	0xdAC17F958D2ee523a2206206994597C13D831ec7
WETH	0xc02aaa39b223fe8d0a0e5c4f27ead9083c756cc2
imBTC	0x3212b29E33587A00FB1C83346f5dBFA69A458923
HBTC	0x0316eb71485b0ab14103307bf65a021042c6d380
PAX	0x8E870D67F660D95d5be530380D0eC0bd388289E1
TUSD	0x0000000000085d4780B73119b644AE5ecd22b376
USDC	0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48
WBTC	0x2260FAC5E5542a773Aa44fBCfeDf7C193bc2C599

✓ Rinkeby Contract Addresses:

Contract Name	Address

MoneyMarket	0xdcfd113789ef683f676435fff90b953a0cc76044
USDx	0xd96cc7f80c1cb595ebcdc072531e1799b3a2436e
USDT	0xaa74b62f737bba1d2e520f9ec38fc23b6e6817df
WETH	0x7a967421410019044aa829746d65575325082e99
imBTC	0x5dc95a046020880b93f15902540dbfe86489fdda
HBTC	0xcf07906CbCF9824D0caE475E8F958d48AcF1014C
PAX	0x722E6238335d89393A42e2cA316A5fb1b8B2EB55
TUSD	0xe72a3181f69Eb21A19bd4Ce19Eb68FDb333d74c6
USDC	0x4dbcdf9b62e891a7cec5a2568c3f4faf9e8abe2b
WBTC	0x7B65B937A0f3764a7a5e29fD696C391233218E91

✓ Other Resources

1. **APIs**

   https://github.com/Lendfme/docs/blob/master/API/Lendfme_API.md

2. **JSON ABI**

   https://etherscan.io/address/0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea#code

3. **Source Code**

   https://etherscan.io/address/0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea#code

4. **All Interfaces**

   https://etherscan.io/address/0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea#readContract

   https://etherscan.io/address/0x0eEe3E3828A45f7601D5F54bF49bB01d1A9dF5ea#writeContract