Vault manager

The vault manager works with vaults that are owned by theCdpManager contract, which is also used by Oasis Borrow. This intermediary contract allows the use of incrementing integer IDs for vaults, familiar to users of Single-Collateral Sai, as well as other conveniences.

In the code, this is calledCdpManager .

```

Copy constmgr=maker.service('mcd:cdpManager');

```

Instance methods

The methods below are all asynchronous.

getCdpIds()

Return an array describing the vaults owned by the specified address. Note that if the vaults were created in Oasis Borrow, the address of the proxy contract should be used.

```

Copy constproxyAddress=awaitmaker.service('proxy').currentProxy(); constdata=awaitmgr.getCdpIds(proxyAddress); const{id,ilk}=data[0]; // e.g. id = 5, ilk = 'ETH-A'

```

getCdp()

Get an existing vault by its numerical ID. Returns aVault instance .

```

Copy constvault=awaitmgr.getCdp(111);

```

open()

Open a new vault with the specifiedcollateral type . Will create aproxy if one does not already exist. Returns aVault instance . Works with thetransaction manager .

```

Copy consttxMgr=maker.service('transactionManager'); constopen=awaitmgr.open('ETH-A'); txMgr.listen(open,{ pending:tx=>console.log('tx pending: '+tx.hash) }); constvault=awaitopen;

```

openLockAndDraw()

Open a new vault, then lock and/or draw in a single transaction. Will create aproxy if one does not already exist. Returns aVault instance .

```

Copy constvault=awaitmgr.openLockAndDraw( 'BAT-A', BAT(1000), DAI(100) );

```

Vault instances

In the code, these are calledManagedCdp .

Properties

A note on caching: When a vault instance is created, its data is pre-fetched from the blockchain, allowing the properties below to be read synchronously. This data is cached in the instance. To refresh this data, do the following:

```

Copy vault.reset(); awaitvault.prefetch();

```

collateralAmount

The amount of collateral tokens locked, as a currency unit .

collateralValue

The USD value of collateral locked, given the current price according to the price feed, as a currency unit .

debtValue

The amount of Dai drawn, as a currency unit .

liquidationPrice

The USD price of collateral at which the Vault becomes unsafe.

isSafe

Whether the Vault is currently safe or not.

Instance methods

All of the methods below are asynchronous and work with the transaction manager . Amount arguments should be currency units , e.g.:

```

Copy import{ ETH,DAI }from'@makerdao/dai-plugin-mcd';

awaitvault.lockAndDraw(ETH(2),DAI(20));

```

lockCollateral(amount)

Deposit the specified amount of collateral.

drawDai(amount)

Generate the specified amount of Dai.

lockAndDraw(lockAmount, drawAmount)

Deposit some collateral and generate some Dai in a single transaction.

wipeDai(amount)

Pay back the specified amount of Dai.

wipeAll()

Pay back all debt. This method ensures that dust amounts do not remain.

freeCollateral(amount)

Withdraw the specified amount of collateral.

wipeAndFree(wipeAmount, freeAmount)

Pay back some debt and withdraw some collateral in a single transaction.

wipeAllAndFree(freeAmount)

Pay back all debt, ensuring dust amounts do not remain, and withdraw a specified amount of collateral in a single

transaction.

give(address)

Transfer ownership of this vault toaddress . Note that if the new owner plans to use this vault with Oasis Borrow, it should be transferred to their proxy withgiveToProxy instead.

giveToProxy(address)

Look up the proxy contract owned byaddress and transfer ownership of this vault to that proxy.

Previous Plugins Next Collateral types Last updated3 years ago On this page *Instance methods * getCdpIds() * getCdp() * open() * openLockAndDraw() * Vault instances * Properties * Instance methods

Export as PDF