Getting started

Installation

Install the package with npm in your terminal:

npm install @makerdao/dai

Once it's installed, import the module into your project as shown below.

```
Copy importMakerfrom'@makerdao/dai'; // or constMaker=require('@makerdao/dai');
```

Multi-Collateral Dai support in Dai.js is implemented as aplugin . This may change in the future. The MCD Plugin is also available as annpm package and its source code can be found onGithub .

npm install @makerdao/dai-plugin-mcd

```
Copy import{ McdPlugin }from'@makerdao/dai-plugin-mcd'; // or const{McdPlugin}=require('@makerdao/dai-plugin-mcd');
```

(Note the.default at the end of the line when usingrequire .)

UMD

This library is also usable as aUMD module , which you can build withnpm run build:frontend .

```
Copy
```

Quick Examples

Look up information about a vault

This code usesgetCdpIds to look up a vault that was created in theOasis Borrow UI. Since this code is only reading data, not creating any transactions, it is not necessary to provide a private key or connect a wallet.

```
Copy // you provide these values constinfuraKey='your-infura-api-key'; constownerAddress='0xf00...';

constmaker=awaitMaker.create('http',{ plugins:[McdPlugin], url:https://mainnet.infura.io/v3{infuraKey} });

constmanager=maker.service('mcd:cdpManager');
constproxyAddress=maker.service('proxy').getProxyAddress(ownerAddress);
constdata=awaitmanager.getCdpIds(proxyAddress);// returns list of { id, ilk } objects
constvault=awaitmanager.getCdp(data[0].id);

console.log([ vault.collateralAmount,// amount of collateral tokens vault.collateralValue,// value in USD, using current price feed values vault.debtValue,// amount of Dai debt vault.collateralizationRatio,// collateralValue / debt vault.liquidationPrice// vault becomes unsafe at this price ].map(x=>x.toString());
```

Create a vault

The code below opens a Vault, locks ETH into it, and draws out Dai.

Since this code sends transactions, it requires an account that can sign transactions. The simplest way to do this is to provide aprivateKey configuration option as shown below, but you can also connect to Metamask or other browser-based providers, or connect to hardware wallets.

```
```

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

// you provide these values constinfuraKey='your-infura-api-key'; constmyPrivateKey='your-private-key';

constmaker=awaitMaker.create('http',{ plugins:[McdPlugin], url:https://mainnet.infura.io/v3{infuraKey}, privateKey:myPrivateKey });

// verify that the private key was read correctly console.log(maker.currentAddress());

// make sure the current account owns a proxy contract; // create it if needed. the proxy contract is used to // perform multiple operations in a single transaction awaitmaker.service('proxy').ensureProxy();

// use the "vault manager" service to work with vaults constmanager=maker.service('mcd:cdpManager');

// ETH-A is the name of the collateral type; in the future, // there could be multiple collateral types for a token with // different risk parameters constvault=awaitmanager.openLockAndDraw( 'ETH-A', ETH(50), DAI(1000) );

console.log(vault.id); console.log(vault.debtValue);// '1000.00 DAI'
```

In the next section, learn more about how to configure the Maker instance withMaker.create . Or jump to learning more aboutaccounts ,vaults ,proxies , andcurrency units .

Integration Examples

For larger examples of integrating this library, check out thisrepo and theDai react template .

Previous The Dai Javascript Library of the Maker ProtocolNext Configuration Last updated3 years ago On this page * Installation * Quick Examples * Look up information about a vault * Create a vault * Integration Examples

Export as PDF