Single-Collateral Sai

Installation

Single-Collateral Dai support in Dai.js is implemented as a plugin . The SCD plugin is also available as a npm package and its source code can be found on Github .

npm install @makerdao/dai-plugin-scd

```
```

Copy import{ ScdPlugin }from'@makerdao/dai-plugin-scd'; // or const{ScdPlugin}=require('@makerdao/dai-plugin-scd');

```
```

Quick example

The code below creates a CDP, locks ETH into it, and draws out Sai.

```
```

Copy importMakerfrom'@makerdao/dai'; import{ ScdPlugin }from'@makerdao/dai-plugin-scd';

asyncfunctionopenLockDraw() { constmaker=awaitMaker.create("http",{ plugins:[ScdPlugin], privateKey:YOUR_PRIVATE_KEY, url:'https://kovan.infura.io/v3/YOUR_INFURA_PROJECT_ID' });

awaitmaker.authenticate(); constcdpService=awaitmaker.service('cdp'); constcdp=awaitcdpService.openCdp();

awaitcdp.lockEth(0.25); awaitcdp.drawSai(50);

constdebt=awaitcdp.getDebtValue(); console.log(debt.toString);// '50.00 SAI' }

openLockDraw();

```
```

The services and objects below are used to work with Single-Collateral Sai.

- CDP Service
- Collateralized Debt Position
- System Status
- Token Conversion
-

openCdp()

- Returns:
- promise (resolves to new CDP object once mined)
-

openCdp() will create a new CDP, and then return the CDP object, which can be used to access other CDP functionality. The promise will resolve when the transaction is mined.

```
```

Copy constcdpService=awaitmaker.service('cdp'); constnewCdp=awaitcdpService.openCdp();

```
```

getCdp(int id)

- Returns:
- promise (resolves to CDP object)
-

getCdp(id) creates a CDP object for an existing CDP. The CDP object can then be used to interact with your CDP.

```
```

Copy constcdpService=awaitmaker.service('cdp'); constcdp=awaitcdpService.getCdp(614);

```
```

Once you have an instance of a CDP, you can use [CDP instance methods](#) to read its state and perform actions.

Last updated3 years ago On this page * [Installation](#) * [Quick example](#) * [openCdp()](#) * [getCdp(int id)](#)

[Export as PDF](#)