

Using multiple accounts

Summary

Dai.js supports the use of multiple accounts (i.e. private keys) with a single Maker instance. Accounts can be specified in the options for `Maker.create` or with the `addAccount` method.

Call `useAccount` to switch to using an account by its name, or `useAccountWithAddress` to switch to using an account by its address, and subsequent calls will use that account as the transaction signer.

When the Maker instance is first created, it will use the account named `default` if it exists, or the first account in the list otherwise.

...

```
Copy const maker = await Maker.create({ url: 'http://localhost:2000', accounts: { other: { type: 'privateKey', key: someOtherKey }, default: { type: 'privateKey', key: myKey } } });
```

```
await maker.addAccount('yetAnother', { type: 'privateKey', key: thirdKey });
```

```
const cdp1 = await maker.openCdp(); // owned by "default"
```

```
maker.useAccount('other'); const cdp2 = await maker.openCdp(); // owned by "other"
```

```
maker.useAccount('yetAnother'); const cdp3 = await maker.openCdp(); // owned by "yetAnother"
```

```
await maker.addAccount({ type: 'privateKey', key: fourthAccount.key }); // the name argument is optional
maker.useAccountWithAddress(fourthAccount.address); const cdp4 = await maker.openCdp(); // owned by the fourth account
```

...

You can check the current account with `currentAccount` and `currentAddress` :

...

Copy

```
maker.currentAccount() { name: 'other', type: 'privateKey', address: '0xfff...' } maker.currentAddress() '0xfff...'
```

...

Account types

In addition to the `privateKey` account type, there are two other built-in types:

- `provider`
 - : Get the first account from the provider (e.g. the value from `getAccounts`
 -).
- `browser`
 - : Get the first account from the provider in the browser (e.g. MetaMask), even if the Maker instance is configured to use a different provider.
-

...

```
Copy const maker = await Maker.create({ url: 'http://localhost:2000', accounts: { // this will be the first account from the provider at // localhost:2000 first: { type: 'provider' },
```

```
// this will be the current account in MetaMask, and it // will send its transactions through MetaMask second: { type: 'browser' },
```

```
// this account will send its transactions through the // provider at localhost:2000 third: { type: 'privateKey', key: myPrivateKey } } })
```

...

Hardware wallets

Plugins can add additional account types. There are currently two such plugins for hardware wallet support:

- [dai-plugin-trezor-web](#)
- [dai-plugin-ledger-web](#)

•
...

```
Copy importTrezorPluginfrom'@makerdao/dai-plugin-trezor-web'; importLedgerPluginfrom'@makerdao/dai-plugin-ledger-web';
```

```
constmaker=awaitMaker.create({ plugins:[ TrezorPlugin, LedgerPlugin, ], accounts:{ // default derivation path is "44'/60'/0'/0/0" myTrezor:{type:'trezor',path:derivationPath1}, myLedger:{type:'ledger',path:derivationPath2} } });
```

...

Demo

Install the[multiple accounts demo app](#) to see this functionality in action.

[Previous Events](#) [Next Adding a new service](#) Last updated4 years ago On this page *[Summary](#) * [Account types](#) * [Hardware wallets](#) * [Demo](#)

[Export as PDF](#)