Configuration

Maker.create

You can configure the behavior of Dai.js by passing different arguments toMaker.create . The first argument is the name of a preset, and the second is an options object.

Presets

- 'browser'
-
    - Use this preset when using the library in a browser environment. It will attempt to connect usingwindow.ethereum
-
    - orwindow.web3
-
    - .
- *
- 'http'
-
    - Connect to a JSON-RPC node. Requiresurl
-
    - to be set in the options.
- *
- 'test'
-
    - Use a local node (e.g. Ganache) running athttp://127.0.0.1:2000
-
    - , and sign transactions using node-managed keys.
- *
-

```

Copy constmakerBrowser=awaitMaker.create('browser');

constmakerHttp=awaitMaker.create('http',{ url:'https://kovan.infura.io/v3/YOUR_INFURA_PROJECT_ID' });

constmakerTest=awaitMaker.create('test');

```

Options

- privateKey
-
    - Optional. The private key used to sign transactions. If this is omitted, the first account available from the Ethereum provider will be used. Only used with the'http'
-
    - preset.
-
    - If this is omitted and the provider does not have an unlocked account, themaker
-
    - object will start inread-only mode
-
    - .
- *
- url
-
    - The URL of the node to connect to. Only used with the'http'
-
    - preset.
- *
- web3.transactionSettings
-
    - Object containing transaction options to be applied to all transactions sent through the library.
-
    - Default value:{ gasLimit: 4000000 }
- *
- web3.confirmedBlockCount

-
    - Number of blocks to wait after a transaction has been mined when callingconfirm
-
    - . SeeTransactions
-
    - for further explanation.
-
    - Default value: 5
- *
- web3.inject
-
    - For advanced users. You can inject your own custom instance of a Web3 provider with this, instead of using the default HttpProvider.
- *
- log
-
    - Set this tofalse
-
    - to reduce the verbosity of logging.
- *
- autoAuthenticate
-
    - Set this tofalse
-
    - to create the Maker instance without connecting yet. If so, you must runawait maker.authenticate()
-
    - before using any other methods.
- *
-

```

Copy // It doesn't necessarily make sense to set all these // options at the same time (e.gurl and inject), // this is just meant to illustrate the shape of the // options object. constmaker=awaitMaker.create('http',{ privateKey:YOUR_PRIVATE_KEY,// '0xabc...' url:'http://some-ethereum-rpc-node.net', web3:{ statusTimerDelay:2000, confirmedBlockCount:8 transactionSettings: { gasPrice:12000000000 }, inject:someProviderInstance }, log:false, autoAuthenticate:false });

```

Instance methods

service()

- Returns:
- service object
-

Return a service instance that was included in this instance ofmaker .

```

Copy constaccountsService=maker.service('accounts');

```

Services

The MCD plugin defines several services for working with Multi-Collateral Dai. ReviewGetting started to see how to add the plugin.

- 'mcd:cdpManager'
- : for working with Vaults.
- 'mcd:cdpType'
- : for reading parameters and live data (totals and prices) for collateral types.
- 'mcd:savings'
- : for working with the Dai Savings Rate.
- 'mcd:systemData'
- : for reading system-wide parameters.
-

Read-only mode

As mentioned above, theMaker instance can be used in read-only mode, if you just want to read data from the blockchain without signing any transactions. Simply omit theprivateKey option.

You can start in read-only mode and stilladd an account with the ability to sign transactions later on.

Previous Getting started Next Plugins Last updated3 years ago On this page *Maker.create * Presets * Options * Instance methods * service() * Services * Read-only mode

Export as PDF