

Transaction manager

The `transactionManager` service is used to track a transaction's status as it propagates through the blockchain.

Methods in `Dai.js` that start transactions are asynchronous, so they return promises. These promises can be passed as arguments to the transaction manager to set up callbacks when transactions change their status to `pending`, `mined`, `confirmed` or `error`.

...

```
Copy const txMgr = maker.service('transactionManager'); // instance of transactionManager
const open = maker.service('cdp').openCdp(); // open is a promise--note the absence of await
```

...

Pass the promise to `transactionManager.listen` with callbacks, as shown below.

...

```
Copy txMgr.listen(open, { pending: tx => { // do something when tx is pending }, mined: tx => { // do something when tx is mined },
confirmed: tx => { // do something when tx is confirmed }, error: tx => { // do something when tx fails } });
```

```
await txMgr.confirm(open); // 'confirmed' callback will fire after 5 blocks
```

...

Note that the `confirmed` event will not fire unless `transactionManager.confirm` is called. This async function waits a number of blocks (default 5) after the transaction has been mined to resolve. To change this globally, set the `confirmedBlockCount` attribute in `MakerOptions`. To change it for just one call, pass the number of blocks to wait as the second argument:

...

```
Copy await txMgr.confirm(open, 3);
```

...

Transaction Metadata

There are functions such as `lockEth()` which are composed of several internal transactions. These can be more accurately tracked by accessing `tx.metadata` in the callback which contains both the contract and the method the internal transactions were created from.

Transaction Object Methods

A `TransactionObject` also has a few methods to provide further details on the transaction:

- `hash`
- `transactionHash`
- `fees()`
- `amountOfEtherSpentOnGas`
- `timestamp()`
- `timestampOfWhenTransactionWasMined`
- `timestampSubmitted()`
- `timestampOfWhenTransactionWasSubmittedToTheNetwork`
-

...

```
Copy const lock = cdp.lockEth(1); txMgr.listen(lock, { pending: tx => { const { contract, method } = tx.metadata;
if (contract === 'WETH' && method === 'deposit') { console.log(tx.hash); // print hash for WETH.deposit } } })
```

...

[Previous](#) [Advanced](#) [Next](#) [DSProxy](#) Last updated 4 years ago On this page * [Transaction Metadata](#) * [Transaction Object Methods](#)

[Export as PDF](#)