Ethers.js Cheat Sheet

#ethers#web3#javascript
ethers.js is a library that interact with Ethereum Blockchain.

It is a very useful library but the official documentation was a little hard to read for me so I would like to summarize it for easy reference. (Focusing on what will be used often.)
*They are arranged in alphabetical order.

Accounts

Gets a list of accounts
const accounts = await provider.listAccounts();

Example:
// Connect web3
const provider = new ethers.providers.Web3Provider(window.ethereum);

const accounts = await provider.listAccounts();
console.log(accounts[0]);

Balance

Gets a blanace of address
const balance = await provider.getBalance(`address`);

Example:
// Connect web3
const provider = new ethers.providers.Web3Provider(window.ethereum);

const address = "0x28d3...";

const balance = await provider.getBalance(address);

console.log(`The ${address} balance: ${balance.toString()}`);

Connect (MetaMask)

Connects to Ethereum with MetaMask
const provider = new ethers.provider.Web3Provider(window.ethereum);

Connect (RPC)

Connects to Ethereum with RPC
const provider = new ethers.provider.JsonRpcProvider(`url`);

`url` for example:

Platform	URL
Alchemy	`https://<network>.alchemyapi.io/v2/YOUR-API-KEY`
Infura	`https://<network>.infura.io/v3/YOUR-PROJECT-ID`

Contract

Create a contract instance by signer.

It does not work if the user does not have a wallet or is not connected.
const contract = new ethers.Contract(`address`, `abi`, `signer`);

Example:
import Artifact from './Contract.json';

// Connect web3
const provider = new ethers.providers.Web3Provider(window.ethereum);
const signer = provider.getSigner();

const contractAddress = "0x9fE4...";

```
const contract = new ethers.Contract(
   contractAddress,
   Artifact.abi,
   signer
);

// Call a state-change method
const userAddress = "0x28d3...";
const dai = ethers.utils.parseUnits("1.0", 18);
await contract.transfer(userAddress, dai);
```

Contract (Read-Only)

Create a contract instance by provider.

It can call Read-Only methods only. Instead, it also works if the user doesn't have a wallet or isn't connected.
const contract = new ethers.Contract(`address`, `abi`, `provider`);

Example:
```
import Artifact from './Contract.json';

// For example here, interact with Alchemy JSON-RPC
const provider = new
ethers.providers.JsonRpcProvider("https://eth-mainnet.alchemyapi.io/v2/<YOUR-API-KEY>");

const contractAddress = "0x9fE4...";

const contract = new ethers.Contract(
   contractAddress,
   Artifact.abi,
   provider
);

// Call a getter method
```

```
const contractName = await contract.name();
console.log(`Contract name is ${contractName}`);
```

Contract Event Listener

Listens events emitted in contract.

```
contract.on(`event`, `listener`);
```

Example:

```
contract.on("TransferedFrom", (from, to) => {
   console.log(`Token transfered from ${from} to ${to}`);
});

contract.on("Minted", (tokenId) => {
   console.log(`Token #${tokenId} minted`);
});
```

Convert (Ether -> Wei)

Returns `BigNumber`.

```
const wei = ethers.utils.parseEther(`ETH`);
```

Example:

```
const weiBigNumber = ethers.utils.parseEther("0.2");
const wei = weiBigNumber.toString();

console.log("wei: ", wei);
```

Convert (Wei -> Ether)

Returns `string`.

```
const ether = ethers.utils.formatEther(`wei`);
```

Example:
const address = "0x28d319067E209fa43Ef46bF54343Dae4CEDd3824";
const balanceBigNumber = await ethers.providers.getBalance(address);

const balance = ethers.utils.formatEther(balanceBigNumber.toString());
console.log(`user balance: ${balance} Ether`);

Install

npm install ethers

Import

for CommonJS
const { ethers } = require('ethers');

for ES Modules
import { ethers } from 'ethers';

Network & Chain ID

Gets a connecting network and chain ID.
const network = await provider.getNetwork();
const chainId = network.chainId;

Example:
// Connect web3
const provider = new ethers.providers.Web3Provider(window.ethereum);

const network = await provider.getNetwork();
const chainId = network.chainId;

Chain ID List for example:

Chain ID	**Network**
1	Mainnet
3	Ropsten
4	Rinkeby
5	Goerli
10	Optimism
42	Kovan
56	BSC
137	Polygon
42161	Arbitrum One
43114	Avalanche