Web3 Server-Side Signed Transaction

Introduction

- When you are calling web3 from a server application you do not have the option of having MetaMask pop-up to sign your transactions for you.
- You need to have the private key available so that you can sign the transactions before sending to the node.
- The node needs to see that you are the owner of the account. The node will decrypt your transaction using your public key.

Setting up for server-side transaction

Things you will need:

- Web3 object with node provider
- Private key
- ABI JSON interface
- Contract address

Transaction Data Object

```
//   construct the transaction data
  const txData = {
    nonce: web3.utils.toHex(txCount),
    gasLimit: web3.utils.toHex(250000),
    gasPrice: web3.utils.toHex(1000),
    from: addressFrom,
    to: contractAddress,
    data: encodedABI,
    value: web3.utils.toHex(web3.utils.toWei("1", 'ether'))
  }
```

Transaction Data Object

- **nonce**: Each Ethereum account has a field called nonce to keep track of the total number of transactions that account has executed.

```
web3.eth.getTransactionCount(Ethereum_Address)
```

- **data**: Let's see how this data field is calculated.

```
myContract.methods.myMethod([param1],[param2],..[param N]).encodeABI()
```

Introduction to Infura

Infura is a hosted Ethereum node cluster that lets your users run your application without requiring them to set up their own Ethereum node or wallet.

PRE-REQUISITE

- Browser with MetaMask Extension
  - including
    - mnemonic string
    - 3 or 4 account in Ropsten with minimum 2 ether
- ExpressJS
- Infura with Project ID
- Truffle

THANK YOU