



Transaction

블록체인에서의 Transaction

- 거래의 기록 단위
 - 거래: 데이터
 - 기록: 데이터 저장
- 블록체인은 Transaction을 **Block** 단위의 **Chain** 형태로 저장하는 기술임
 - 즉, 거래 기록을 Blockchain 형태로 저장

Kaikas에서의 Transaction

- Kaikas에서는 트랜잭션을 만드는 메서드들을 제공한다.

방법 1) `klay_sendAsync()`

```
const transactionParameters = {
  gas: '0x2710',
  to: '0x0000000000000000000000000000000000000000000000000000000000000000',
  from: klaytn.selectedAddress,
  value: '0xff'
}

klaytn.sendAsync(
  {
    method: 'klay_sendTransaction',
    params: [transactionParameters],
    from: klaytn.selectedAddress
  },
```

```
callback
)
```

- `klay_sendAsync()` 메서드를 통해 `klay_sendTransaction()` 메서드를 실행
- 직접 메서드를 호출하지 않는 이유?

Sends a message to the Kaikas browser. Message format maps to the format of [Klaytn JSON-RPC API](#).

- `klay_sendAsync()` 메서드는 RPC API 형태로 래핑하는 과정을 수행해준다.

방법 2) Caver 라이브러리 사용 (more recommended)

```
// example of sending 1 KLAY

import Caver from 'caver-js' // or const Caver = require('caver-js')
const caver = new Caver(klaytn) // createing Caver instance
caver.klay
  .sendTransaction({
    type: 'VALUE_TRANSFER',
    from: klaytn.selectedAddress,
    to: '0x0000000000000000000000000000000000000000000000000000000000000000',
    value: caver.utils.toPeb('1', 'KLAY'),
    gas: 8000000
  })
  .once('transactionHash', transactionHash => {
    console.log('txHash', transactionHash)
  })
  .once('receipt', receipt => {
    console.log('receipt', receipt)
  })
  .once('error', error => {
    console.log('error', error)
  })
```

```
// injected caver, caver-js를 설치할 필요가 없

window.caver.sendTransaction({
  type: 'VALUE_TRANSFER',
  from: klaytn.selectedAddress,
  to: '0x0000000000000000000000000000000000000000000000000000000000000000',
  value: caver.utils.toPeb('1', 'KLAY'),
  gas: 8000000
})
```