

# Tutorial de Interoperar entre Moombean e Ethereum

**Detalhe importante:** Estaremos interoperando usando Moombean Alpha que é a rede de testes da Moombean. Onde já existe uma conta Moombean válida e uma carteira virtual com ativos Moombean(DEV) e uma carteira Virtual mantida para Tokens Ethereum, onde ambas são mantidas pela metamask. Todos os ativos da Conta Moombean(DEV) podem ser transferidos para a conta Ethereum, através de uma transação de conversão de valores, conforme abordado a baixo no tutorial.

1. Primeiramente é necessario instalar o node.js [clikando aqui](#)
2. Crie uma pasta para armazenar os arquivos que vamos utilizar. Para criar uma pasta pelo terminal do windows, use `mkdir nome_da_sua_pasta`
3. Abra o powershell e use o comando `cd Caminho_da_pasta_criada` para que o terminal acesse o local dentro da pasta que você criou
4. Agora use o comando `npm init --y` para inciar o ambiente node dentro da pasta que estamos.
5. Use o comando `npm install web3 solc@0.8.0` para instalar a biblioteca web3 que é um compilador de código solidity que te permite usar código solidity através do javascript.
6. crie um arquivo para verificação de balanço (ex: `balance.js`)
7. Crie a conexão com a web3 com o seguinte codigo no arquivo de balanço. (Básicamente um "import" de linguagens como python)

```
const { Web3 } = require('web3');  
const web3 = new Web3('https://rpc.api.moonbase.moonbe
```

8. Crie a função de verificação de balanço, substituindo os endereços

```
const addressFrom = 'INSERT_FROM_ADDRESS'; //endereço :  
const addressTo = 'INSERT_TO_ADDRESS'; // endereço rec
```

```

const balances = async () => {
  const balanceFrom = web3.utils.fromWei(
    await web3.eth.getBalance(addressFrom),
    'ether'
  );
  const balanceTo = web3.utils.fromWei(
    await web3.eth.getBalance(addressTo),
    'ether'
  );

  console.log(`The balance of ${addressFrom} is: ${balanceFrom}`);
  console.log(`The balance of ${addressTo} is: ${balanceTo}`);
};

balances();

```

## 9. Crie um arquivo para realizar as transferencias e inicie a conexão com a web3

```

const { Web3 } = require('web3');

const web3 = new Web3('https://rpc.api.moonbase.moonbase.org');

```

Agora crie a função de transferencia

```

const accountFrom = {
  privateKey: 'INSERT_YOUR_PRIVATE_KEY', //Substitua pela sua chave privada
  address: 'INSERT_PUBLIC_ADDRESS_OF_PK', //endereço da carteira
};

const addressTo = 'INSERT_TO_ADDRESS'; // endereço da carteira para a qual será enviada a transação

const send = async () => {
  console.log(
    `Attempting to send transaction from ${accountFrom.address} to ${addressTo}`
  );

  const createTransaction = await web3.eth.accounts.signTransaction(
    {
      gas: 21000,
      to: addressTo,
      value: web3.utils.toWei('1', 'ether'), //Valor da transação em ether
    },
    accountFrom.privateKey
  );
};

```

```

        gasPrice: await web3.eth.getGasPrice(),
        nonce: await web3.eth.getTransactionCount(accountFrom,
        },
        accountFrom.privateKey
    );

    const createReceipt = await web3.eth.sendSignedTransaction(
        createTransaction.rawTransaction
    );
    console.log(
        `Transaction successful with hash: ${createReceipt.hash}
    );
};

send();

```

#### 10. Volte ao powershell e execute os arquivos

```

node balance.js //Retorna o balanço atual das contas
node transaction.js // Retorna o hash da transação caso bem sucedida
node balance.js// Conferir se realmente ocorreu transferência

```

#### 11. Prontinho, sua transação foi realizada com sucesso!