

Desarrollo Blockchain Ethereum con Solidity

Módulo 2 - Web3

Web3

Web3

- WEB3 es un **framework** especializado en la **interconexión** Frontend-Blockchain.
- Existen diferentes versiones del mismo.
- Es posible utilizar tanto la versión javascript (**web3js**) como la versión Python (**web3.py**).
- Provee una serie de métodos funciones y propiedades que simplifican la interacción con los contratos inteligentes liberados en la blockchain de Ethereum.
- Toda la documentación asociada se la puede encontrar pública en <https://web3js.readthedocs.io>



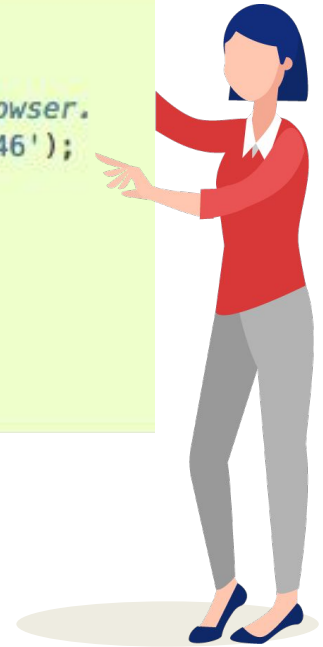
Introducción al framework

Web3 se compone de los siguientes **submódulos** dentro de los cuales se podrán llevar a cabo diferentes tareas de interconexión frontend-blockchain:

- **Eth:** el módulo eth permite interactuar con la red Ethereum.
- **Net:** el módulo net permite interactuar con las propiedades de la red.
- **Personal:** el módulo personal permite interactuar con las cuentas de Ethereum.
- **Ssh:** el módulo ssh permite interactuar con el protocolo SSH.
- **Bzz:** el módulo bzz permite interactuar con las swarm networks.

Introducción al framework

```
var Web3 = require('web3');  
  
// "Web3.providers.givenProvider" will be set if in an Ethereum supported browser.  
var web3 = new Web3(Web3.givenProvider || 'ws://some.local-or-remote.node:8546');  
  
> web3.eth  
> web3.shh  
> web3.bzz  
> web3.utils  
> web3.version
```



Introducción al framework

Web3.eth: Se compone, a su vez de los siguientes **submódulos** dentro de los cuales se podrán llevar a cabo diferentes tareas:

- **isMining:** Devuelve true si se está minando o false en caso contrario.
- **getAccounts:** devuelve una lista de cuentas controladas por el nodo.
- **getBalance:** devuelve el balance de una cuenta pudiendo especificar un bloque particular.
- **getTransaction:** devuelve información de la transacción cuyo hash se recibe por parámetro.
- **getBlockNumber:** devuelve el número del bloque más reciente.

```
web3.eth.isMining([callback])
```

```
web3.eth.getAccounts([callback])
```

```
web3.eth.getBalance(address [, defaultBlock] [, callback])
```

```
web3.eth.getTransaction(transactionHash [, callback])
```

```
web3.eth.getBlockNumber([callback])
```

Introducción al framework

Web3.eth - sendTransaction

Uno de los métodos más importantes del subconjunto web3.eth es el de **sendTransaction** a través del cual será posible enviar la transacción a la red. La función recibe un **transactionObject** que se compone de:

- **from:** la dirección (address) de quien envía la transacción.
- **to:** la dirección (address) de quien recibe la transacción.
- **value:** valor en Ether (expresado en WEI) que se envía junto con la transacción.
- **gas:** unidades de gas que se espera gastar en esta transacción.
- **gasPrice:** precio que el emisor está dispuesto a pagar por cada unidad de gas consumida.
- **data:** Contiene información específica de la llamada a la función.

```
// using the callback
web3.eth.sendTransaction({
  from: '0xde0B295669a9FD93d5F28D9Ec85E40f4cb697BAe',
  data: code // deploying a contract
}, function(error, hash){
  ...
});
```

Introducción al framework

Web3.eth.personal: se compone a su vez de los siguientes **submódulos** dentro de los cuales se podrán llevar a cabo diferentes tareas:

- **newAccount:** permitirá la creación de una cuenta dentro de Ethereum

```
web3.eth.personal.newAccount(password, [callback])
```

- **unlockAccount:** permite desbloquear una cuenta por un tiempo determinado

```
web3.eth.personal.unlockAccount(address, password, unlockDuration [, callback])
```

- **sign:** permite firmar data usando una cuenta específica

```
web3.eth.personal.sign(dataToSign, address, password [, callback])
```

- **ecRecover:** permite recuperar la dirección del firmante de una transacción

```
web3.eth.personal.ecRecover(dataThatWasSigned, signature [, callback])
```


Cuentas desbloqueadas

Para transaccionar más fácil vía web3 se puede desbloquear una cuenta en el módulo personal. Esto se recomienda para nodos privados pero no es una buena práctica en nodos públicos.

```
web3.eth.personal.unlockAccount("0x11f4d0A3c12e86B4b5F39B213F7E19D048276DAe", "test password!", 600)
  .then(console.log('Account unlocked!'));
  > "Account unlocked!"
```

¡Muchas gracias!

¡Sigamos trabajando!