# Web3js

1. **介绍**

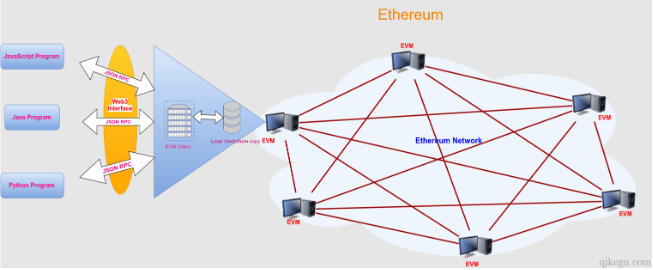
开发以太坊区块链应用程序，涉及到以下部分:

* **智能合约开发** – 使用solidity语言编写代码，部署到区块链。
* **网站或客户端开发** – 与区块链中的智能合约进行交互，读写数据。

在进行网站或客户端开发时，就需要用到[web3.js](https://github.com/ethereum/web3.js/)。web3.js库是一个javascript库，可以让你执行很多与区块链进行交互的任务，例如：

* 以太币转账
* 读写智能合约中的数据
* 创建智能合约

区块链网络是一个点对点的对等网络，网络由各个节点相互连接构成，智能合约的代码与数据存储在区块链中。使用web3.js可以向区块链网络中某个以太坊节点发出请求，与区块链网络进行交互。



web3.js使用JSON RPC与Ethereum区块链通信(JSON RPC即JSON远程调用)，对区块链网络进行数据读写。这有点像使用jQuery ajax库来读写web服务器上的数据。

1. **环境搭建**

* **安装node**
* **安装web3js**

**C:\Users\ADMINI~1\AppData\Local\Temp\1656659881(1).png**

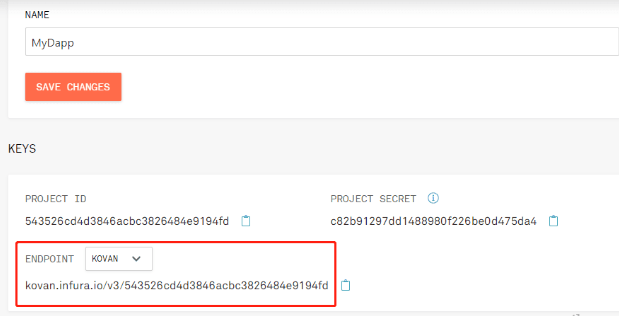
* **连接以太坊节点**

要访问以太坊网络，需要连接到Ethereum节点。

可以使用Geth或Parity运行自己的Ethereum节点。但这需要从区块链下载大量数据并保持同步，很麻烦。

比较方便的方法是，使用[Infura](https://infura.io/)访问Ethereum节点。Infura是一个免费提供Ethereum节点的服务。

在Infura上注册账号，创建项目，在项目详情页上可以查看API KEY：



1. **访问区块链网络**
2. **创建web3连接：**

在node控制台中加载Web3.js库:

**C:\Users\ADMINI~1\AppData\Local\Temp\1656660552(1).png**

接下来，使用Infura URL，创建Web3连接。声明Infura URL变量：

**C:\Users\ADMINI~1\AppData\Local\Temp\1656660587(1).png**

确保使用之前获得的实际Infura API密钥替换此处的YOUR\_INFURA\_API\_KEY。创建Web3连接:

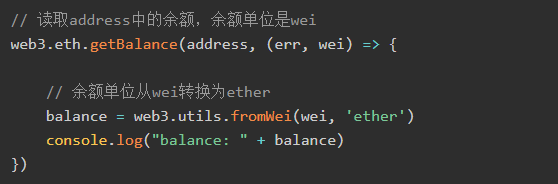
**C:\Users\ADMINI~1\AppData\Local\Temp\1656660639(1).png**

1. **读取余额：**

我们可以看一下这个帐户下有多少以太币

**C:\Users\ADMINI~1\AppData\Local\Temp\1656660828(1).png**

通过使用web3.eth.getBalance()检查帐户余额:

****

输出：

**C:\Users\ADMINI~1\AppData\Local\Temp\1656661005(1).png**

1. **智能合约对象**

智能合约对象是javascript中对智能合约的表示，可以使用智能合约对象访问智能合约。

对象可以使用web3.eth.Contract()函数获得，此函数需要2个参数: 智能合约ABI、智能合约地址。

ABI代表“Abstract Binary Interface/抽象二进制接口”，它是一个JSON数组，是面向以太坊虚拟机的可执行文件，类似于windows 平台上的二进制可执行文件。智能合约编译后生成ABI文件，可以在以太坊虚拟机上执行。

****

有了ABI和地址，就可以创建OmiseGo通证智能合约的Javascript对象：

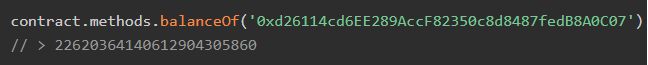
**C:\Users\ADMINI~1\AppData\Local\Temp\1656661812(1).png**

* **调用智能合约读函数**

智能合约对象的methods属性下，包含了对应智能合约的所有函数。要调用智能合约中的某个函数，例如myFunction()，可以使用contract.methods.myFunction()的方式调用。

**注意**：这里调用方法只能调用读操作函数，不能调用写操作函数，写操作函数会改变区块链状态，调用写函数被视为交易。

* 查看一下这个地址0xd26114cd6EE289AccF82350c8d8487fedB8A0C07的OmiseGo通证余额情况：



* 查看通证的名称：

