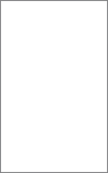
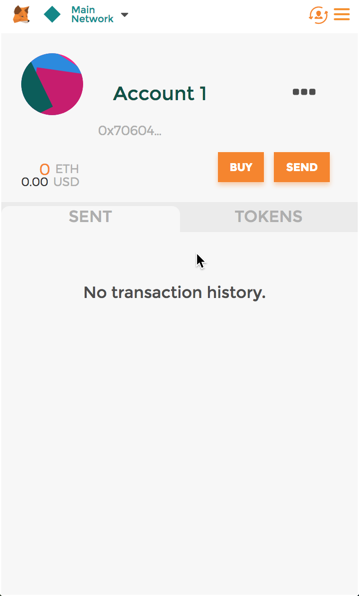
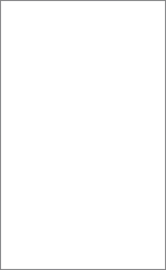
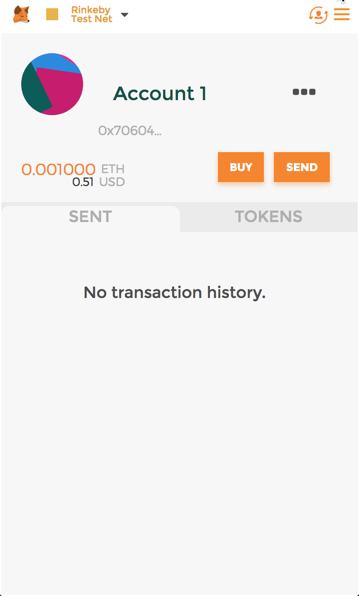
继续回到我们的学习。

在这一课的内容中，让我们一起来了解下ETH的交易流程。

首先打开Chrome浏览器，点击右上角的小狐狸图标。此时可能会让你输入自己的密码，以便访问账户。

登陆后会看到类似下面的界面。

0.001ETH,0.51USD，貌似比上次还多了点？

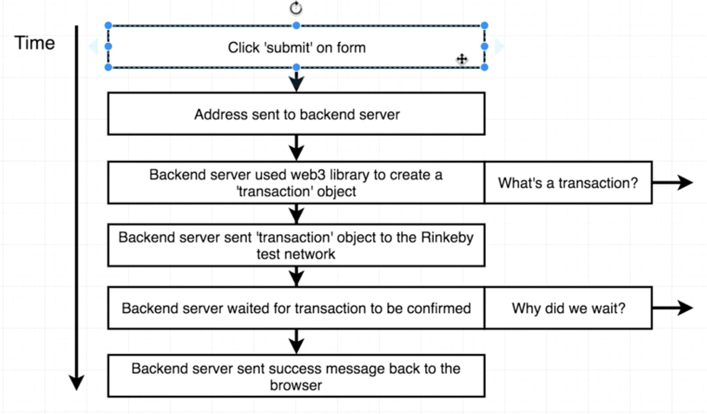
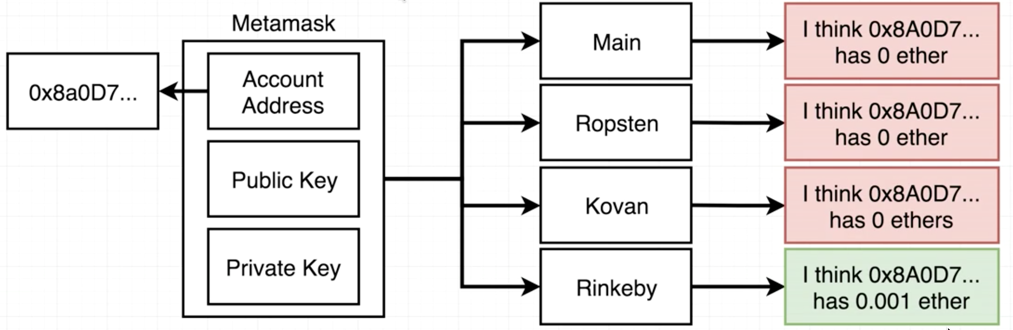


这是因为这里显示的金额是跟ETH的“行价”绑定的。不过先别激动，当我们从顶部的Rinkeby Test Network切换到主网，或是其它测试网络，会遗憾的看到：

好吧，也就是说这0.001ETH实际上是基于Rinkeby Test Network的，并不能作为主网的ETH来交易~

这是为什么呢？

打个形象的比方，大多数大型网络游戏在研发的过程之中都会进行内部测试和公开测试，而内部测试的账号很多时候到了公测或者正式开服的时候是要删档的。这里你可以把Rinkeby Test Network勉强看做是游戏的内测账号，那么不管你在这个内测账号上有了多么NB的成就，甚至是弄到了多少神器，都没办法成为游戏公测和正式开服时的账号上的东西。



或者换个比方，虽然ETH的四种网络对应的都是同一个账号，但是不同网络就像是同一个人在小学，初中，高中和大学四个阶段一样，在不同阶段的成绩是完全不同的，都需要从零开始累积。

好了，接下来让我们来看看，当我们在rinkeby-faucet.com网站中输入地址并点击Submit提交按钮后，究竟执行了哪些后台的操作呢？

1.用户在网页上点击”Submit”提交按钮

2.网页自动将用户输入的地址发送到后台服务器。

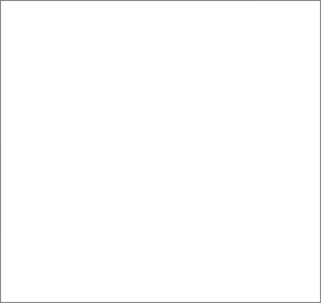
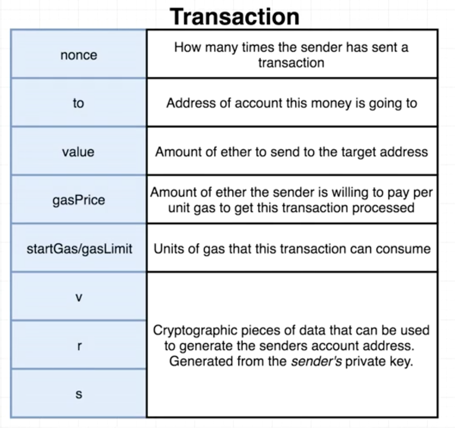
这里的后台服务器是一个node.js服务器，它的作用就是发送0.001ETH到用户所输入的账户地址。

3.后台服务器使用web3库创建了一个”transaction”交易对象。

还记得在之前的课程中所提到的吗？web3.js是开发者和ETH网络进行交互的重要开发工具，不仅仅是主网，也包括其它几种测试网络。

问题来了，到底什么是transaction交易对象呢?

我们可以把transaction看做一个交易记录，其中记录了一个账号向另外一个账户进行ETH交易的相关信息。



transaction对象中包含了多个属性，也就是上表中所显示的各个属性。

其中nonce代表发送方发送交易信息的次数。

to代表接收方的账户地址。

value表示要发送到接受方的ETH数量。

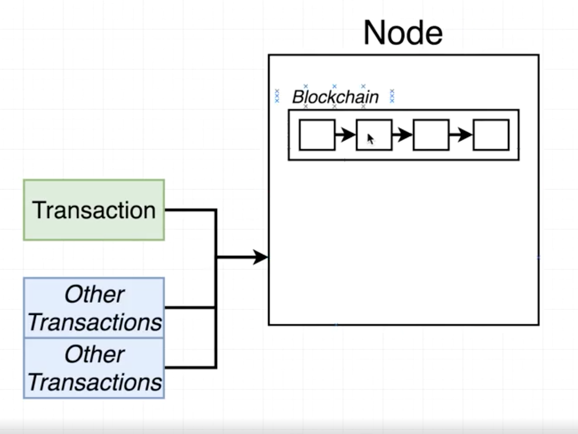
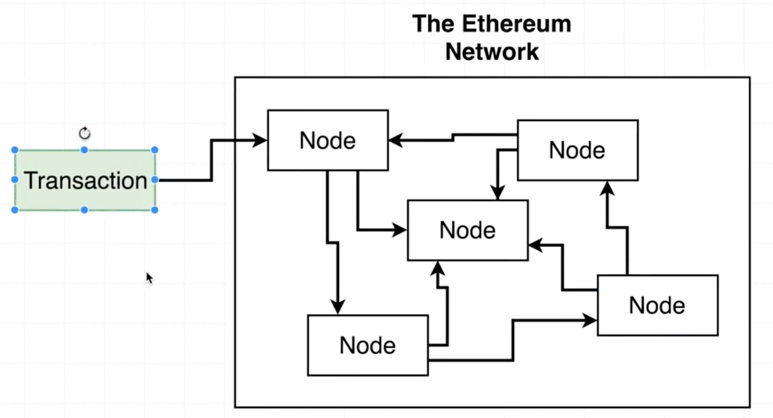
接下来的两个属性代表的是交易费用，这里先不详细解释，后面我们回过头来会再仔细看看。

最后的三个属性非常重要，它们用于生成发送方的账户地址，通常使用发送方的私钥生成。其背后的原理异常复杂，因此这里我们暂时不做详细的解释。需要特别说明的是，生成v,r,s属性的过程是单向的。也就是说我们可以使用发送方的私钥生成v,r,s，但是却无法通过v,r,s来反向生成发送方的私钥。

4.后台服务器将transaction对象发送到Rinkeby Test Network。

5.后台服务器等待交易的确认。

好了，在这个等待的过程中，究竟发生了什么呢？为什么我们等了这么长的时间来确认交易呢？让我们来一起看看。



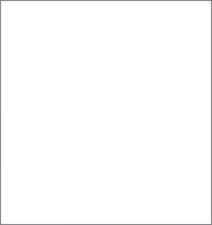
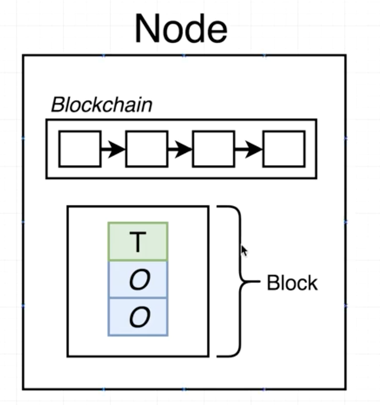
当我们把交易信息发送到ETH的网络时，它会首先被某个特定的节点所接收。

ETH网络中的任何一个节点中都包含了区块链的完整信息。

这里我们先把区块链看做某种特殊形态的数据结构，后面将更详细的说明它的原理。

当我们发起的交易传递到该节点的时候，实际上可能还有很多其它的交易也传递到该节点。

这里将我们的交易信息以T表示，其它的交易信息以O表示。这些交易信息共同组成了一个Block，此时节点会使用某种特殊的方式对Block进行验证。



而这个验证过程通常会需要大概30秒左右的时间，这段时间构成了我们主要的交易确认等待时间。

实际上，这个验证过程就是有些童鞋可能比较熟悉的mining(挖矿），是的，就是挖矿。接下来是该讲一下挖矿的原理了？

是的，让我们期待下一课的精彩内容，也是区块链技术的核心内容之一~