

加密僵尸：僵尸作战系统

可支付

`payable` 方法是让 Solidity 和以太坊变得如此酷的一部分 —— 它们是一种可以接收以太的特殊函数。

先放一下。当你在调用一个普通网站服务器上的API函数的时候，你无法用你的函数传送美元——你也不能传送比特币。

但是在以太坊中，因为钱（_以太_），数据（事务负载），以及合约代码本身都存在于以太坊。你可以在同时调用函数并付钱给另外一个合约。

```
1 | contract OnlineStore {
2 |     function buySomething() external payable {
3 |         // 检查以确定0.001以太发送出去来运行函数：
4 |         require(msg.value == 0.001 ether);
5 |         // 如果为真，一些用来向函数调用者发送数字内容的逻辑
6 |         transferThing(msg.sender);
7 |     }
8 | }
```

在这里，`msg.value` 是一种可以查看向合约发送了多少以太的方法，另外 `ether` 是一个内建单元。

这里发生的事是，一些人会从 `web3.js` 调用这个函数（从DApp的前端），像这样：

```
1 | // 假设 `OnlineStore` 在以太坊上指向你的合约：
2 | OnlineStore.buySomething().send(from: web3.eth.defaultAccount, value: web3.utils.toWei(0.001))
```

注意这个 `value` 字段，JavaScript 调用来指定发送多少(0.001)以太。如果把事务想象成一个信封，你发送到函数的参数就是信的内容。添加一个 `v` 信封里面放钱 —— 信件内容和钱同时发送给了接收者。

注意：如果一个函数没标记为`payable`，而你尝试利用上面的方法发送以太，函数将拒绝你的事务。

提现

```
1 | contract GetPaid is Ownable {
2 |     function withdraw() external onlyOwner {
3 |         owner.transfer(this.balance);
4 |     }
5 | }
```

你可以通过 `transfer` 函数向一个地址发送以太，然后 `this.balance` 将返回当前合约存储了多少以太。所以如果100个用户每人向我们支付1以太，`this.balance` 将是100以太。

你可以通过 `transfer` 向任何以太坊地址付钱。比如，你可以有一个函数在 `msg.sender` 超额付款的时候给他们退税：

```
1 | uint itemFee = 0.001 ether;
2 | msg.sender.transfer(msg.value - itemFee);
```

或者在一个有卖家和买家的合约中，你可以把卖家的地址存储起来，当有人买了它的东西的时候，把买家支付的钱发送给它 `seller.transfer(msg`

随机数

他是不安全的，但对于本游戏已经足够，这里不做阐述用`cekka256`产生随机数的方法

重构通用逻辑

用 `modifier` 来清理代码并避免重复编码。



LEVI_104

👍 0 🗨 0 ⭐ 0

“相关推荐”对你有帮助么？

-  非常没帮助
-  没帮助
-  一般
-  有帮助
-  非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照



LEVI_104

👍 0 🗨 0 ⭐ 0