

一、概述

1. 本教程介绍

1. 使用hardhat部署到不同网络，自动verify代码
2. 单元测试编写
3. deploy脚本，deploy插件使用
4. 可升级合约编写，使用，页面关联

你对工具链的所有疑问都会在这里得到demo答案！

2. hardhat框架的好处

1. 在config文件中配置网络，部署在node，hardhat，其他指定的网络
2. 在.env中写密钥等信息
3. 可以自动部署，Verify代码
4. 可以编写单元测试

二、代码目录

参考链接：

1. 教程：<https://medium.com/@rahulsethram/the-new-solidity-dev-stack-buidler-ethers-waffle-typescript-tutorial-f07917de48ae>
2. hardhat：<https://learnblockchain.cn/docs/hardhat/guides/waffle-testing.html>

三、部署在localhost

安装：

```
1 | npm install
```

单元测试：

```
1 | npx hardhat test
```

启动本地内存区块链环境：

```
1 | npx hardhat node
```

1. 普通部署

方式1，使用hardhat-deploy包部署，具体编写查看 `deploy` 文件夹

```
1 | npx hardhat --network localhost deploy
```

方式2，使用脚本部署

```
1 | npx hardhat run --network localhost scripts/deploy.ts
```

测试一下，需要修改地址 `testV1.ts` 中的地址为刚刚部署的地址。

```
1 | npx hardhat run scripts/testV1.ts --network localhost
```

2. 可升级部署

- 以可升级方式部署合约

1. 执行命令：

```
1 | npx hardhat run --network localhost scripts/01-deploy_counter.ts
```

得到地址：0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9

- 调用后查看运行结果

1. 修改testV1.ts中的地址为：0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9

2. 执行命令：

```
1 | npx hardhat run --network localhost scripts/testV1.ts
```

- 修改代码后开始升级

1. 首先创建新文件：CounterV2Upgrade.sol，相比之下比之前的合约增加了 `changeOwner` 方法。

2. 修改 `02-upgradeCounterV2.t` 中的合约地址为：0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9。

3. 执行升级命令：

```
1 | npx hardhat run --network localhost scripts/02-upgradeCounterV2.ts
```

- 调用合约，确认结果

1. 修改 `testV2.ts` 中的地址为：0xCf7Ed3AccA5a467e9e704C703E8D87F634fB0Fc9

2. 执行命令，查看结果，结果应该加1，`manager` 也应该发生改变。

```
1 | npx hardhat run --network localhost scripts/testV2.ts
```

四、部署在kovan

1. 普通部署（略）

略，同上，只需要将localhost改为kovan即可。

```
1 | #部署在本地：
2 | npx hardhat --network localhost deploy
3 |
4 | #部署在kovan：
5 | npx hardhat --network kovan deploy
```

2. 可升级部署

先备份.openzeppelin下面的kovan.json文件（若有）

- 以可升级方式部署合约

1. 执行命令：

```
1 | npx hardhat run --network kovan scripts/01-deploy_counter.ts
```

2. 该操作会部署三个合约，得到三个地址（存在vscode工程下：.openzeppelin/kovan.json文件中）

1. **proxy**合约（**对外不变的**）：

1. 地址：0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22。
2. 默认verify，因为这个升级合约是openzeppelin官方提供的，很多人已经验证过了。

2. **proxyAdmin**合约：

1. 地址：0x5670ffB7167bc72d3B11e209133aCC73Fb9292be。
2. 默认verify，proxy的管理合约，用于修改代理与实现升级操作。

3. 我们的**Counter**合约：

1. 地址：0x06e166edB942fe7Dd2b04d13443BbB6e835aEd39。
2. 我们真正关心的业务逻辑。

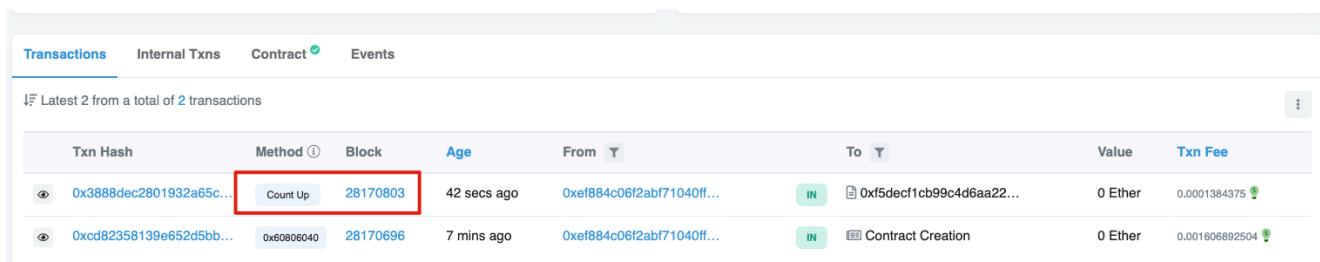
- 调用后查看运行结果

1. 修改 testV1.ts 中的地址为：0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22。

2. 执行命令：

```
1 | npx hardhat run --network kovan scripts/testV1.ts
```

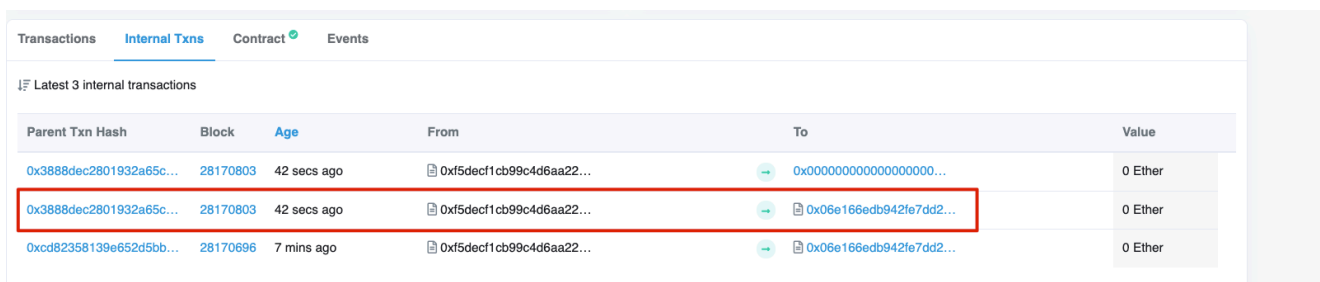
3. 通过浏览器，输入代理合约Proxy地址，我们可以看到已经成功调用了一次CountUp，数据变成：1，并且通过Internal Txns标签看到交互的合约是具体实现合约：



The screenshot shows a table of transactions. The first transaction is highlighted with a red box. It has a Txn Hash of 0x3888dec2801932a65c..., Method of Count Up, Block of 28170803, Age of 42 secs ago, From address of 0xef884c06f2abf71040ff..., To address of 0xf5decf1cb99c4d6aa22..., Value of 0 Ether, and Txn Fee of 0.0001384375.

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x3888dec2801932a65c...	Count Up	28170803	42 secs ago	0xef884c06f2abf71040ff...	0xf5decf1cb99c4d6aa22...	0 Ether	0.0001384375
0xcd82358139e652d5bb...	Contract Creation	28170696	7 mins ago	0xef884c06f2abf71040ff...	Contract Creation	0 Ether	0.001606892504

具体交互合约正是我们的Counter合约，如下图：

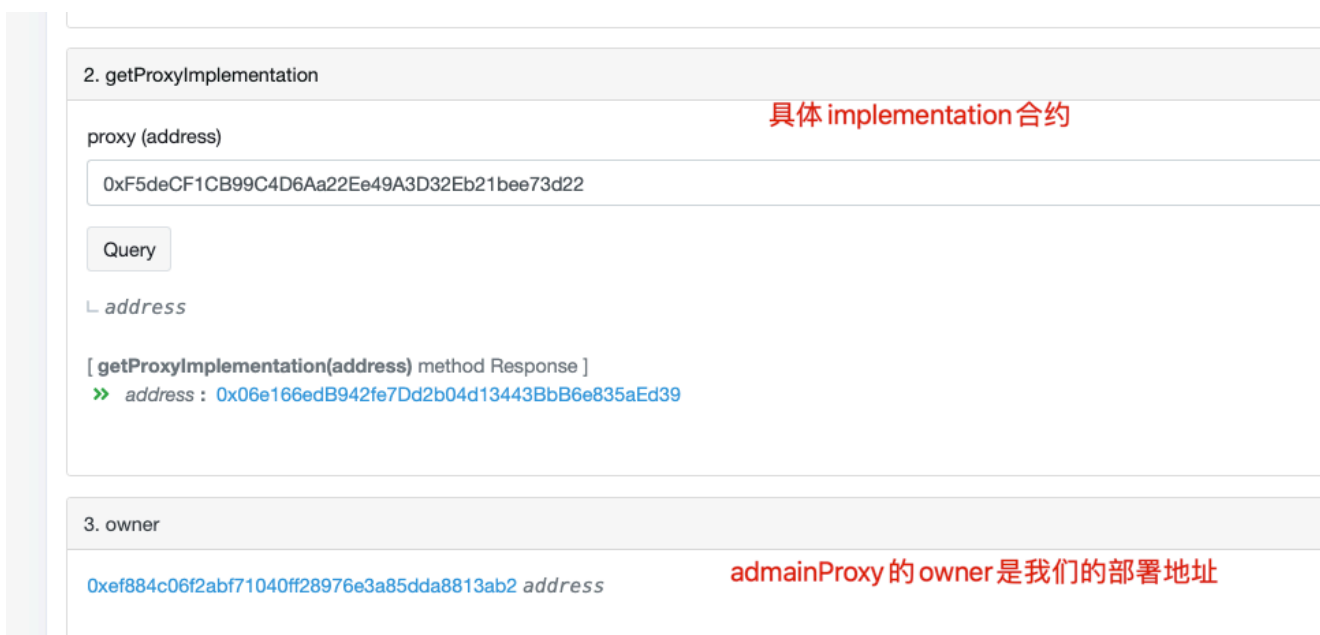


The screenshot shows a table of internal transactions. The second transaction is highlighted with a red box. It shows a call from 0xf5decf1cb99c4d6aa22... to 0x06e166edb942fe7dd2... with a value of 0 Ether.

Parent Txn Hash	Block	Age	From	To	Value
0x3888dec2801932a65c...	28170803	42 secs ago	0xf5decf1cb99c4d6aa22...	0x000000000000000000...	0 Ether
0x3888dec2801932a65c...	28170803	42 secs ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether
0xcd82358139e652d5bb...	28170696	7 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether

疑问：为什么上面还有个：0x0000交互？细节需要看代码。

4. 查看 proxyAdmin 合约，检查一下数据，在读方法中，如下图：输入proxy地址，可以得到我们的Counter合约地址：



The screenshot shows a web interface for the proxyAdmin contract. It has two sections: '2. getProxyImplementation' and '3. owner'. The first section has a text input for 'proxy (address)' with the value '0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22' and a 'Query' button. The second section shows the 'owner' address as '0xef884c06f2abf71040ff28976e3a85dda8813ab2'. Red annotations highlight the '具体 implementation 合约' and 'adminProxy 的 owner 是我们的部署地址'.

2. getProxyImplementation

proxy (address)

0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22

Query

[getProxyImplementation(address) method Response]

>> address : 0x06e166edB942fe7Dd2b04d13443BbB6e835aEd39

3. owner

0xef884c06f2abf71040ff28976e3a85dda8813ab2 address

在写方法，可以看到proxyAdmain的相关方法：更换管理员，更换实现合约

1. changeProxyAdmin

proxy (address)

更换管理员

proxy (address)

newAdmin (address)

newAdmin (address)

Write

2. renounceOwnership

3. transferOwnership

4. upgrade

proxy (address)

更换 proxy 所指向的 implementation 合约

proxy (address)

implementation (address)

implementation (address)

Write

4. 接下来我们verify我们的Counter合约代码：

```
1 #npx hardhat verify --network kovan <合约地址> constructor参数
2 npx hardhat verify --network kovan 0x06e166edB942fe7Dd2b04d13443BbB6e835aEd39
```

```
+ typescript-solidity-dev-starter-kit git:(master) x npx hardhat verify --network kovan 0x06e166edB942fe7Dd2b04d13443BbB6e835aEd39
INFURA_API_KEY: 02cd1e3c295c425597fa105999493baa
PRIVATE_KEY: 5a9e3002fb90a55186dd13d5fe0f9fd10cf9f2d30ef8f277e742606ee3b29c6d
MNEMONIC: rescue enforce dumb hammer waste stadium wave armed expose comfort surround lift
ETHERSCAN_API_KEY: 8D6RIGK9W71R1UE6HD81NE4AWINU5B53BG
Nothing to compile
No need to generate any newer typings.
Compiling 1 file with 0.8.0
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code. Please see https://spdx.org for more information.
--> contracts/Counter.sol

Successfully submitted source code for contract
contracts/Counter.sol:Counter at 0x06e166edB942fe7Dd2b04d13443BbB6e835aEd39
for verification on Etherscan. Waiting for verification result...
```

5. 为了在proxy下可以直接操作具体的实现逻辑，需要我们手动关联proxy和implementation的关系，操作为：在proxy页面点击：more option -> is this a proxy?

Transactions Internal Txns **Contract** Events

Code Read Contract Write Contract

Contract Source Code Verified (Similar Match)
Note: This contract matches the deployed ByteCode of the Source Code for Contract 0x658c0A0DF07F7Fdb2...

Contract Name: TransparentUpgradeableProxy Optimization Enabled: Yes with 200 runs

Compiler Version: v0.8.2+commit.661d1103 Other Settings: default evmVersion

Contract Source Code (Solidity Standard Json-Input format)

File 1 of 7: TransparentUpgradeableProxy.sol

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4 import "../ERC1967/ERC1967Proxy.sol";
5
6
7 /**
8  * @dev This contract implements a proxy that is upgradeable by an admin.
9  *
10  * To avoid https://medium.com/nomic-labs-blog/malicious-backdoors-in-ethereum-proxies-62629adf3357[proxy selector
11  * clashing], which can potentially be used in an attack, this contract uses the
12  * https://blog.openzeppelin.com/the-transparent-proxy-pattern/[transparent proxy pattern]. This pattern implies two
13  * things that go hand in hand:
14  *
```

More Options

- Is this a proxy? NEW
- Similar
- Sol2Uml NEW
- Submit Audit
- Compare NEW

点击: verify

Proxy Contract Verification NEW

Automated Proxy Implementation Verification

This page allows proxy contracts to be linked with their implementation ABIs and contract, thereby making it more intuitive for users to interact and know which contract their proxy address is connected with.

⚠ Disclaimer: The automated process uses programmatic heuristics which may not be accurate despite flagging this contract as a proxy contract. Please practice caution whenever interacting with contracts on Ethereum.

Please enter the Proxy Contract Address you would like to verify

0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22

Verify Reset

点击: save

Proxy Contract Verification Page

The proxy contract verification completed with the message:

The proxy's (0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22) implementation contract is found at:
0x06e166edb942fe7dd2b04d13443bbb6e835aed39

If this is correct, please click Save and the implementation address will be saved. If this is wrong, please click "Contact Us" and let us know.

Save Contact Us

此时, 回到proxy页面, 可以看到implementation对应的方法

Transactions Internal Txns **Contract** Events

Code Read Contract Write Contract **Read as Proxy** NEW Write as Proxy NEW ⓘ Search Source Code

Contract Source Code Verified (Similar Match)
Note: This contract matches the deployed ByteCode of the Source Code for Contract 0x658c0A0DF07F7Fdb2...

Contract Name: TransparentUpgradeableProxy Optimization Enabled: Yes with 200 runs
Compiler Version: v0.8.2+commit.661d1103 Other Settings: default evmVersion

Contract Source Code (Solidity Standard Json-Input format)
File 1 of 7 : TransparentUpgradeableProxy.sol

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.0;
4
5 import "../ERC1967/ERC1967Proxy.sol";
```

Code Read Contract Write Contract **Read as Proxy** NEW Write as Proxy NEW

ABI for the implementation contract at 0x06e166edb942fe7dd2b04d13443bbb6e835aed39, using the EIP-1967 Transparent Proxy pattern.

Connect to Web3

1. countDown

2. countUp

3. initialize

Powered by Etherscan

点击：countUp后，数据变成：2

Transactions Internal Txns **Contract** Events

Code Read Contract Write Contract **Read as Proxy** NEW Write as Proxy NEW

ABI for the implementation contract at 0x06e166edb942fe7dd2b04d13443bbb6e835aed39, using the EIP-1967 Transparent Proxy pattern.

Read Contract Information

1. getCount

2 uint256

- 修改代码后开始升级

1. 修改：（proxy地址）

```
1 02-upgradeCounterV2.ts中的地址为： 0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22
```

2. 执行：

```
1 npx hardhat run --network kovan scripts/02-upgradeCounterV2.ts
```

在upgradeCounterV2合约中，我们增加了一个 `changeOwner` 方法，其他内容未改变。

```

1 function changeOwner(address owner) public {
2     require(msg.sender == manager, "forbidden!");
3     manager = owner;
4 }

```

升级成功:

```

+ typescript-solidity-dev-starter-kit git:(master) x npx hardhat run --network kovan scripts/02-upgradeCounterV2.ts
INFURA_API_KEY: 02cd1e3c295c425597fa105999493baa
PRIVATE_KEY: 5a9e3002fb90a55186dd13d5fe0f9fd10cf9f2d30ef8f277e742606ee3b29c6d
MNEMONIC: rescue enforce dumb hammer waste stadium wave armed expose comfort surround lift
ETHERSCAN_API_KEY: 8D6RIGK9W71R1UE6HD81NE4AWINUSB53BG
No need to generate any newer typings.
INFURA_API_KEY: 02cd1e3c295c425597fa105999493baa
PRIVATE_KEY: 5a9e3002fb90a55186dd13d5fe0f9fd10cf9f2d30ef8f277e742606ee3b29c6d
MNEMONIC: rescue enforce dumb hammer waste stadium wave armed expose comfort surround lift
ETHERSCAN_API_KEY: 8D6RIGK9W71R1UE6HD81NE4AWINUSB53BG
Upgrading CounterV2Upgrade...
counterV2 upgraded: 0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22
+ typescript-solidity-dev-starter-kit git:(master) x

```

3. 到adminProxy中检查升级后的implementation地址: 0xC564f82cA3109F701420a212c345C4747Bc16b0f

2. getProxyImplementation

proxy (address)

0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22

Query

address

[getProxyImplementation(address) method Response]

>> address : 0xC564f82cA3109F701420a212c345C4747Bc16b0f

3. owner

新的implementation地址

4. 回到proxy中, 查看内部交易, 发现:

Contract 0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22

Contract Overview

Balance: 0 Ether

More Info

My Name Tag: Not Available

Creator: 0xf884c06f71040ff... at txn 0xcd82358139e652d5bb...

Transactions Internal Txns Contract Events

Latest 6 internal transactions

Parent Txn Hash	Block	Age	From	To	Value
0x2403d522a26c11fb51c...	28171209	2 mins ago	0x5670fb7167bc72d3b1...	0xf5decf1cb99c4d6aa22...	0 Ether
0xc5ec5737d61ec9338a...	28171160	5 mins ago	0xf5decf1cb99c4d6aa22...	0x0000000000000000...	0 Ether
0xc5ec5737d61ec9338a...	28171160	5 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether
0x3888dec2801932a65c...	28170803	29 mins ago	0xf5decf1cb99c4d6aa22...	0x0000000000000000...	0 Ether
0x3888dec2801932a65c...	28170803	29 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether
0xcd82358139e652d5bb...	28170696	36 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether

adminProxy 向 proxy 合约发送了一笔交易, 修改 proxy 的 implementation 地址

5. 对新发布的implementation合约进行verify代码:

```

1 npx hardhat verify --network kovan 0xC564f82cA3109F701420a212c345C4747Bc16b0f

```



```

→ typescript-solidity-dev-starter-kit git:(master) ✖ npx hardhat verify --network kovan 0xC564f82cA3109F701420a212c345C4747Bc16b0f
INFURA_API_KEY: 02cd1e3c295c425597fa105999493baa
PRIVATE_KEY: 5a9e3002fb90a55186dd13d5fe0f9fd10cf9f2d30ef8f277e742606ee3b29c6d
MNEMONIC: rescue enforce dumb hammer waste stadium wave armed expose comfort surround lift
ETHERSCAN_API_KEY: 8D6RIGK9W71R1UE6HD81NE4AWINUSB53BG
Nothing to compile
No need to generate any newer typings.
Compiling 1 file with 0.8.0
Warning: SPDX license identifier not provided in source file. Before publishing, consider adding a comment containing "SPDX-License-Identifier: <SPDX-License>" to each source file. Use "SPDX-License-Identifier: UNLICENSED" for non-open-source code. Please see https://spdx.org for more information.
--> contracts/CounterV2Upgrade.sol

Successfully submitted source code for contract
contracts/CounterV2Upgrade.sol:CounterV2Upgrade at 0xC564f82cA3109F701420a212c345C4747Bc16b0f
for verification on Etherscan. Waiting for verification result...

Successfully verified contract CounterV2Upgrade on Etherscan.
https://kovan.etherscan.io/address/0xC564f82cA3109F701420a212c345C4747Bc16b0f#code
→ typescript-solidity-dev-starter-kit git:(master) ✖

```

6. 查看proxy的写方法，我们的新方法中，看到新方法已经存在：

Transactions Internal Txns **Contract** Events

Code Read Contract Write Contract Read as Proxy **NEW** Write as Proxy **NEW**

ABI for the implementation contract at [0xc564f82ca3109f701420a212c345c4747bc16b0f](#), using the [EIP-1967 Transparent Proxy pattern](#).
Previously recorded to be on [0x06e166edb942fe7dd2b04d13443bbb6e835aed39](#).

Connected - Web3 [0xEf88...3ab2] [Expand all] [F]

1. changeOwner
2. countDown
3. countUp
4. initialize

Powered by [Etherscan.io](#). [Browse source](#)

读方法中，getCount依然为升级之前的值：2

Transactions Internal Txns **Contract** Events

Code Read Contract Write Contract **Read as Proxy NEW** Write as Proxy **NEW**

ABI for the implementation contract at [0xc564f82ca3109f701420a212c345c4747bc16b0f](#), using the [EIP-1967 Transparent Proxy pattern](#).
Previously recorded to be on [0x06e166edb942fe7dd2b04d13443bbb6e835aed39](#).

Read Contract Information

1. getCount
2 uint256
2. manager
[0xef884c06f2abf71040ff28976e3a85dda8813ab2](#) address

- 调用合约，确认结果

- 1. 修改 testV2.ts 中的地址为：0xF5deCF1CB99C4D6Aa22Ee49A3D32Eb21bee73d22
- 2. 执行命令

```
1 | npx hardhat run --network kovan scripts/testV2.ts
```

3. 效果

TransactionsInternal TxnsContractEvents

Latest 9 internal transactions

Parent Txn Hash	Block	Age	From	To	Value
0xf6b85d2ce3e00d2fd30...	28171403	2 mins ago	0xf5decf1cb99c4d6aa22...	0xc564f82ca3109f70142... 新合约: changeOwner	0 Ether
0x0619a998cd5201027d...	28171400	2 mins ago	0xf5decf1cb99c4d6aa22...	0x000000000000000000...	0 Ether
0x0619a998cd5201027d...	28171400	2 mins ago	0xf5decf1cb99c4d6aa22...	0xc564f82ca3109f70142... 新合约: countUp	0 Ether
0x2403d522a26c11fb51c...	28171209	15 mins ago	0x5670ffb7167bc72d3b1...	0xf5decf1cb99c4d6aa22...	0 Ether
0xc5ec5737d61ec9338a0...	28171160	18 mins ago	0xf5decf1cb99c4d6aa22...	0x000000000000000000...	0 Ether
0xc5ec5737d61ec9338a0...	28171160	18 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2... 旧合约: countUp	0 Ether
0x3888dec2801932a65c...	28170803	42 mins ago	0xf5decf1cb99c4d6aa22...	0x000000000000000000...	0 Ether
0x3888dec2801932a65c...	28170803	42 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether
0xcd82358139e652d5bb...	28170696	49 mins ago	0xf5decf1cb99c4d6aa22...	0x06e166edb942fe7dd2...	0 Ether

[Download CSV Export]

数值改变，owner改变：

CodeRead ContractWrite ContractRead as ProxyNEWWrite as ProxyNEW

ABI for the implementation contract at 0xc564f82ca3109f701420a212c345c4747bc16b0f, using the EIP-1967 Transparent Proxy pattern.
Previously recorded to be on 0x06e166edb942fe7dd2b04d13443bbb6e835aed39.

Read Contract Information

1. getCount

3 uint256

2. manager

0x6491d615b6db93154d6123e97751897cce524787 address