

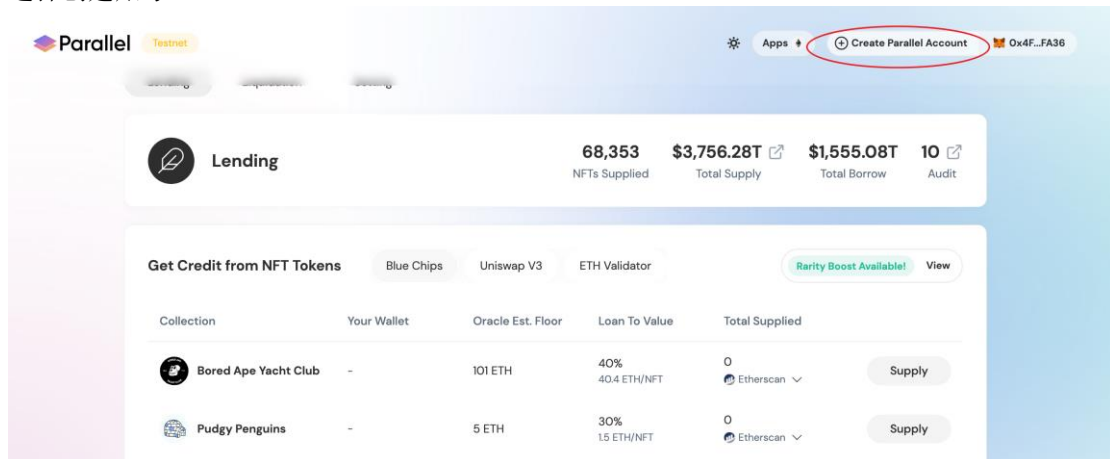
当我们参与一些小型或大型项目时，我们通常会用到 RPA 脚本来模拟元素点击实现交互，这是一种常见的项目交互方式。然而，这种交互方式在网页不‘流畅’的情况下，经常容易出错，导致本该发送的交易没有发送，并且不太容易监控交易是否完成。另一种常见的交互方式是直接通过脚本调用合约。这种方式在合约是开源的情况下尤其有效，因为我们可以很容易地通过下载合约的 abi 来调用合约函数。

接下来，我将以 Parallefi 为例，介绍如果合约未开源，我们应该如何调用合约。这将包括寻找合约的方法，如何使用脚本与合约进行交互，以及如何在实际项目中应用这些知识。我的目标是帮助你理解并掌握这种强大的交互工具，让你在未来的项目中能够更加自如地使用它。

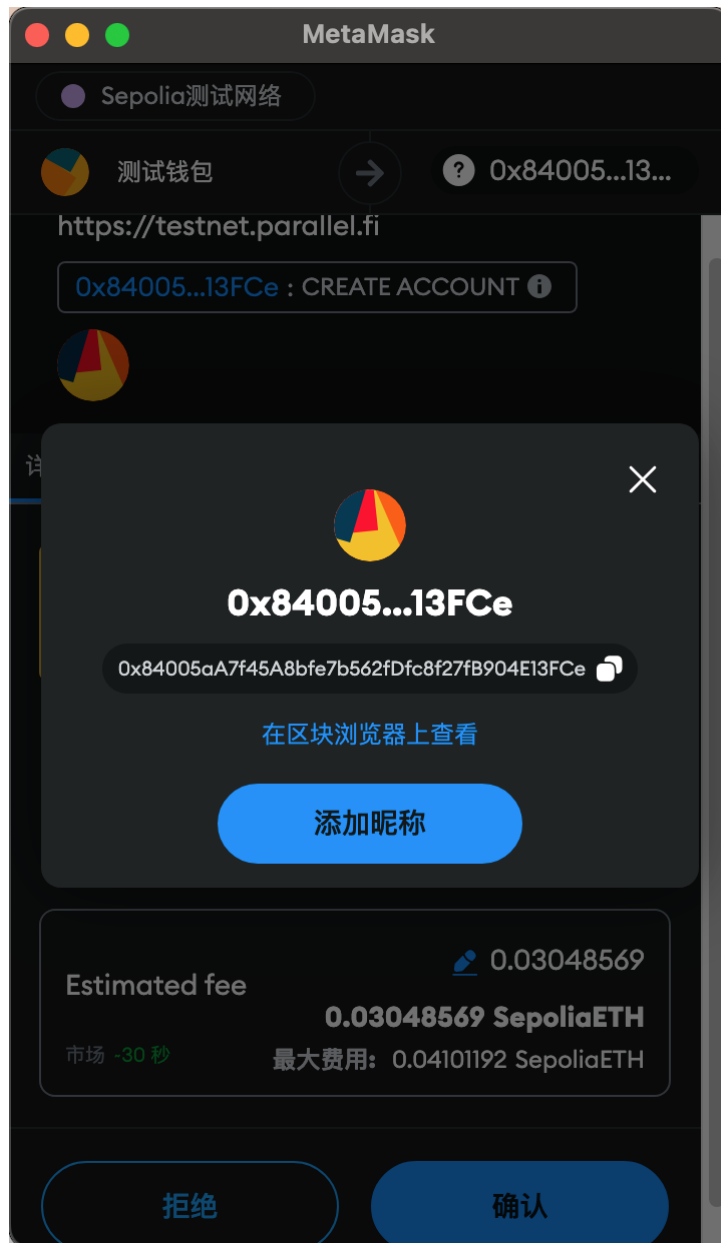
请注意，虽然这个教程以 Parallefi 为例，但这里的方法和技巧同样适用于任何其他的未开源合约。所以无论你是在哪个项目中，都可以参考这个教程来帮助你更好地理解和使用未开源合约。

Parallel 为‘全链’赛道的项目，但是其测试网非常卡顿。非要撸他的话。使用合约交互很适合这个项目。首先打开网站 <https://testnet.paralle.fi/sepolia/apps/paraspacespace/credit>

选择创建账号



小狐狸钱包会弹窗 让我们确认交易



我们可以点击在浏览器上查看该合约地址。

The screenshot shows a block explorer interface. The 'Overview' section displays 'ETH BALANCE' as '0 ETH'. The 'More Info' section shows 'CONTRACT CREATOR' as '0x1828185...23247e9eB' and 'Fund From' as '0x9d05ea...a4cc2612'. The 'Multichain Info' section is 'N/A'. The 'Transactions' section shows a list of transactions, all of which are 'Create Account' transactions. The table below shows the details of these transactions.

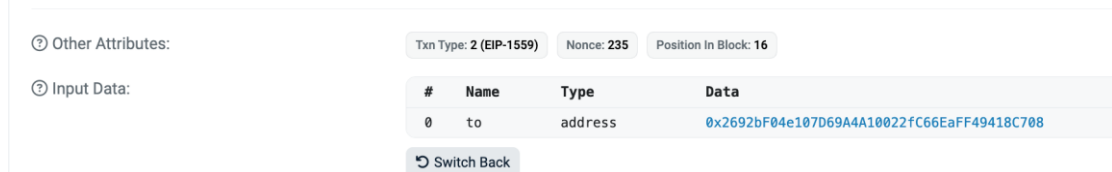
Transaction Hash	Method	Block	Age	From	To	Value	Txn Fee
0xf8568997a187d3...	Create Account	5886447	48 secs ago	0x6ce6401f...8bf806E8d	0x84005aA7...904E13FCe	0 ETH	0.02137801
0x4a830ec21bc7b0...	Create Account	5886437	2 mins ago	0x2dda7e6b...D354F4162	0x84005aA7...904E13FCe	0 ETH	0.02149872
0x8f14e420b931d2...	Create Account	5886437	2 mins ago	0xA1B08869...681C39996	0x84005aA7...904E13FCe	0 ETH	0.02149872
0x5afdf12df83c3d2...	Create Account	5886409	8 mins ago	0x478020dd...1BEa678A3	0x84005aA7...904E13FCe	0 ETH	0.02263328
0x1b36bd97f26385...	Create Account	5886404	9 mins ago	0xD350A702...a77961895	0x84005aA7...904E13FCe	0 ETH	0.02820872
0x4558f187e8da52f...	Create Account	5886403	9 mins ago	0x53461A89...Bf2c55746	0x84005aA7...904E13FCe	0 ETH	0.02163626

可以看到 该合约为开源，且我们调用的方法也是 Create Account

小狐狸弹出之后先简单的观察一下，我们调用的是 Mint 方法，传入的是一个地址类型。

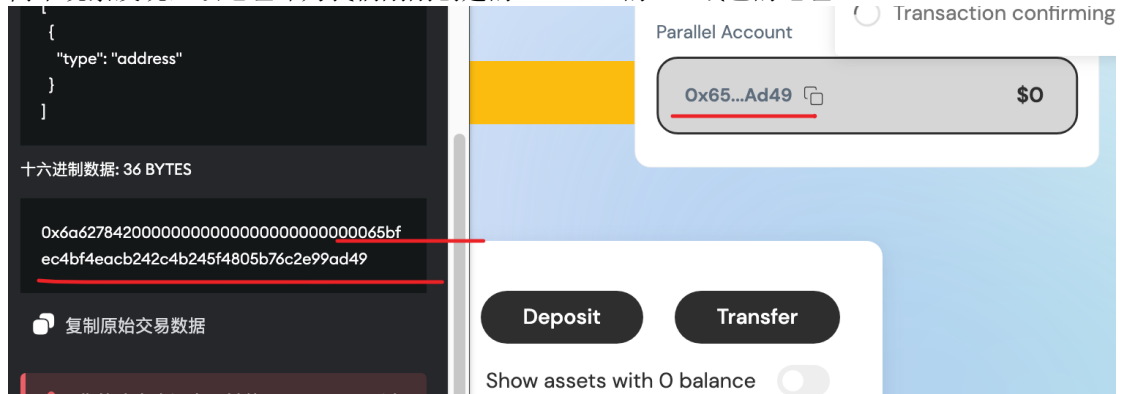


同样打开浏览器观察一下别人的交易



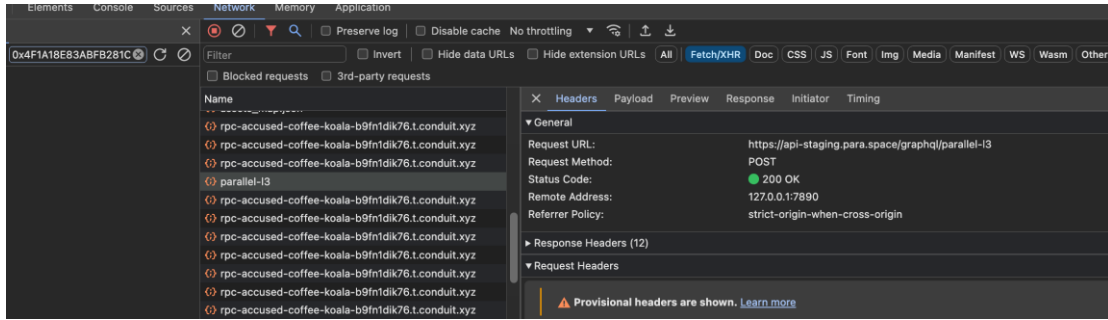
发现只需要传入一个地址即可。

简单观察发现，该地址即为我们刚刚创建的 Parallelfi 的 AA 钱包的地址。



那么我们接下来需要去找到一个接口去获得 AA 钱包的地址。即可构造并发送交易。

我们可以通过 F12 打开 Network 然后搜索我们的 AA 钱包的地址。获取到前端发起的 Http 请求。从响应体里面解析出地址



这里我们可以封装一个函数，输入小狐狸钱包地址，然后返回 AA 钱包的地址。直接贴上代码。

```
def get_AA_wallet_addr():
    global AA_wallet_addr
    headers = {
        'accept': '*/*',
        'accept-language': 'zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7',
        'authorization': '',
        'content-type': 'application/json',
        'origin': 'https://testnet.parallel.fi',
        'referer': 'https://testnet.parallel.fi/',
        'sec-ch-ua': '"Google Chrome";v="123", "Not:A-Brand";v="8", "Chromium";v="123"',
        'sec-ch-ua-mobile': '?0',
        'sec-ch-ua-platform': "macOS",
        'sec-fetch-dest': 'empty',
        'sec-fetch-mode': 'cors',
        'sec-fetch-site': 'cross-site',
        'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/123.0.0.0 Safari/537.36',
    }
    json_data = {
        'operationName': 'ParaAccounts',
        'variables': {
            'filter': {
                'EOAAccount': f'{{sender_address}}',
            },
        },
        'query': 'query ParaAccounts($filter: ParaAccountFilter) {\n  paraAccounts(filter: $filter)
{\n    paraAccount\n      EOAAccount\n      createdHash\n      createdBlockHeight\n      __typename\n    }\n  }'
    }
    response = requests.post('https://api-staging.para.space/graphql/parallel-l3', headers=headers,
json=json_data)
    data = json.loads(response.text)
    # 获取第一个账户地址
    AA_wallet_addr = data['data']['paraAccounts'][0]['paraAccount']
    print(" 获取到AA钱包地址: ",AA_wallet_addr)
```

我们继续构造并发送交易，即可领水成功。

以上就是交互未开源合约的一些思路。

只要是 EVM 系的合约，80%的合约都可以用这个方法进行交互。有部分合约的数据实在不好猜测，就没有办法。注意有些合约的交互中包含签名。而网页前端交互合约也需要签名构造请求，因此签名数据往往也能够从前端的 HTTP 请求中找到。

以上就是我的分享。