合约交互（转移代币、委托代币转移、取款、订单和交易）

一、合约交互方法汇总

根据有无合约的 abi，与合约交互的方法有些不同。

1.合约的 abi 完全公开（自己写的合约或已经开源），那么可以使用 ethers 直接进行交互，无需赘述。

2.合约的 abi 不公开（有大致的模板，但本身未开源，721/1155等），但交互的方法是知道的。比如某个 nft 合约，符合 erc721 标准，有 safeTransfer 方法。那么只要从标准 721 合约中提取出相关的函数，重新组织一下 abi 传入 ethers 既可以同 1 一样进行交互。

3.如果合约的 abi 完全不公开，那么需要调用相关的函数，首先要从区块链浏览器上查看相关的交易信息。提取其中的 data 字段。然后获知方法的签名和参数的组织（前 4 个字节，或者在"0x12121212....."中的 12121212 这 8 个数字即是函数签名，后面是参数）。

原理简述

与合约的交互一般最后都是以 rpc 的形式进行。查看 eth 的 json rpc 文档，发现对于不改变链上状态的是调用 eth_call 方法，改变链上状态的是 eth_sendTransaction 和 eth_sendRawTransaction 方法。这些方法里面都有一个参数 data，既包含合约里面指定的方法，也包含相关的调用参数。所以只要仔细分析 data 数据，理论上就可以直接调用任意想要的方法。所以并不局限于 ethers.js 这个库。

其他问题

1.区块链浏览器上尚无任何交易，如何进行调用？

这时就只能进行反汇编，反编译等逆向工程，做之前需要衡量投入产出是否值得，然后祝君好运。

2.还有其他方法吗？

还真有，可以部署合约，然后调用 call。就是会多花费一些 gas

演示项目（基于hardhat）

https://github.com/TechPlanB/CallFunctionExample

二、代币合约

在Solidity编程中，代币合约是一种智能合约，用于创建和发行数字代币。代币合约通常使用ERC20标准，该标准定义了一组函数和事件，用于代币的转账、查询和分发等功能。在本篇文章中，我们将介绍代币合约的基本概念、功能和实现方式。

2.1 基本概念

代币合约是一个智能合约，用于追踪谁拥有多少代币。它实现了一组函数和事件，用于代币的转账、查询和分发等功能。ERC20标准是代币合约的常用标准，它定义了一组函数和事件，包括transfer、approve、balanceOf等。

2.2 功能

代币合约的主要功能包括：

1. 代币转账：通过调用transfer函数，将一定数量的代币从一个地址转移到另一个地址。
2. 代币查询：通过调用balanceOf函数，查询指定地址的代币余额。
3. 代币分发：通过调用approve函数，允许另一个地址使用一定数量的代币。
4. 代币冻结：通过调用freeze函数，冻结指定地址的代币。

2.3 实现方式

实现一个代币合约需要遵循ERC20标准，并实现以下函数和事件：

1. totalSupply：返回总代币供应量。
2. balanceOf(address _owner)：返回指定地址的代币余额。
3. transfer(address _to, uint256 _amount)：将一定数量的代币从一个地址转移到另一个地址。
4. approve(address _spender, uint256 _amount)：允许另一个地址使用一定数量的代币。
5. transferFrom(address _from, address _to, uint256 _amount)：从一个地址转移一定数量的代币到另一个地址。
6. freeze(address _owner)：冻结指定地址的代币。
7. unfreeze(address _owner)：解除指定地址的代币冻结状态。
8. updateApproval(address _owner, bool _approved)：更新指定地址的批准状态。
9. updateBalances(address[] memory _addresses, uint256[] memory _amounts)：批量更新指定地址的代币余额。
10. add(uint256 _amount)：增加总代币供应量。

11. subtract(uint256 _amount)：减少总代币供应量。
12. safeTransfer(address _to, uint256 _amount, bytes memory _data)：安全转移一定数量的代币到另一个地址，同时可以传递附加数据。
13. safeTransferFrom(address _from, address _to, uint256 _amount, bytes memory _data)：从一个地址安全转移一定数量的代币到另一个地址，同时可以传递附加数据。
14. batchTransfer(address[] memory _to, uint256[] memory _amounts, bytes memory _data)：批量安全转移一定数量的代币到多个地址，同时可以传递附加数据。
15. event Transfer(address indexed _from, address indexed _to, uint256 indexed _amount)：当发生代币转账时触发的事件。
16. event Approval(address indexed _owner, address indexed _approvedAddress, uint256 indexed _amount)：当批准另一个地址使用一定数量的代币时触发的事件。