Go语言调用智能合约

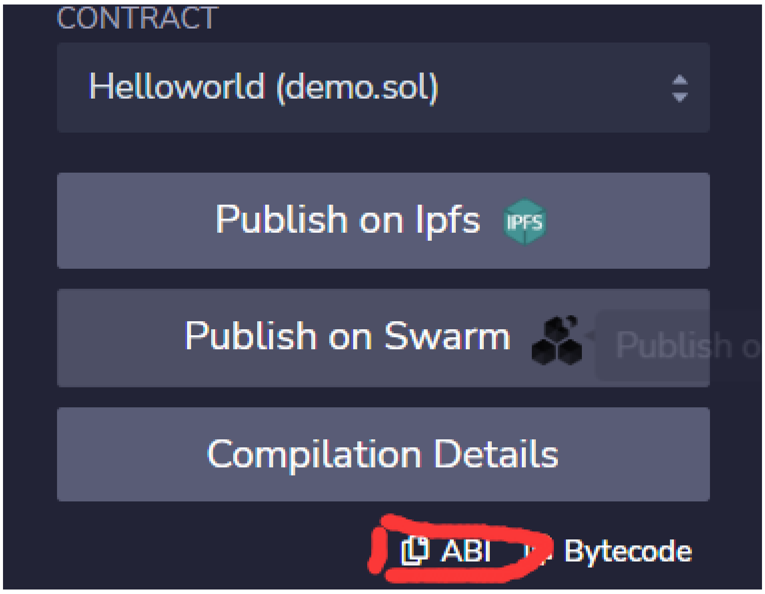
一、编写智能合约

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Helloworld {

    function HelloMyworld() public view returns(string memory mul) {
        string memory word = "hello wrold";
        return word;
    }
}
```

二、使用abigen工具生成go文件



abigen --abi xx.abi --pkg packagename --type structname --out xx.go

abi 文件在 remix 部署时可以得到

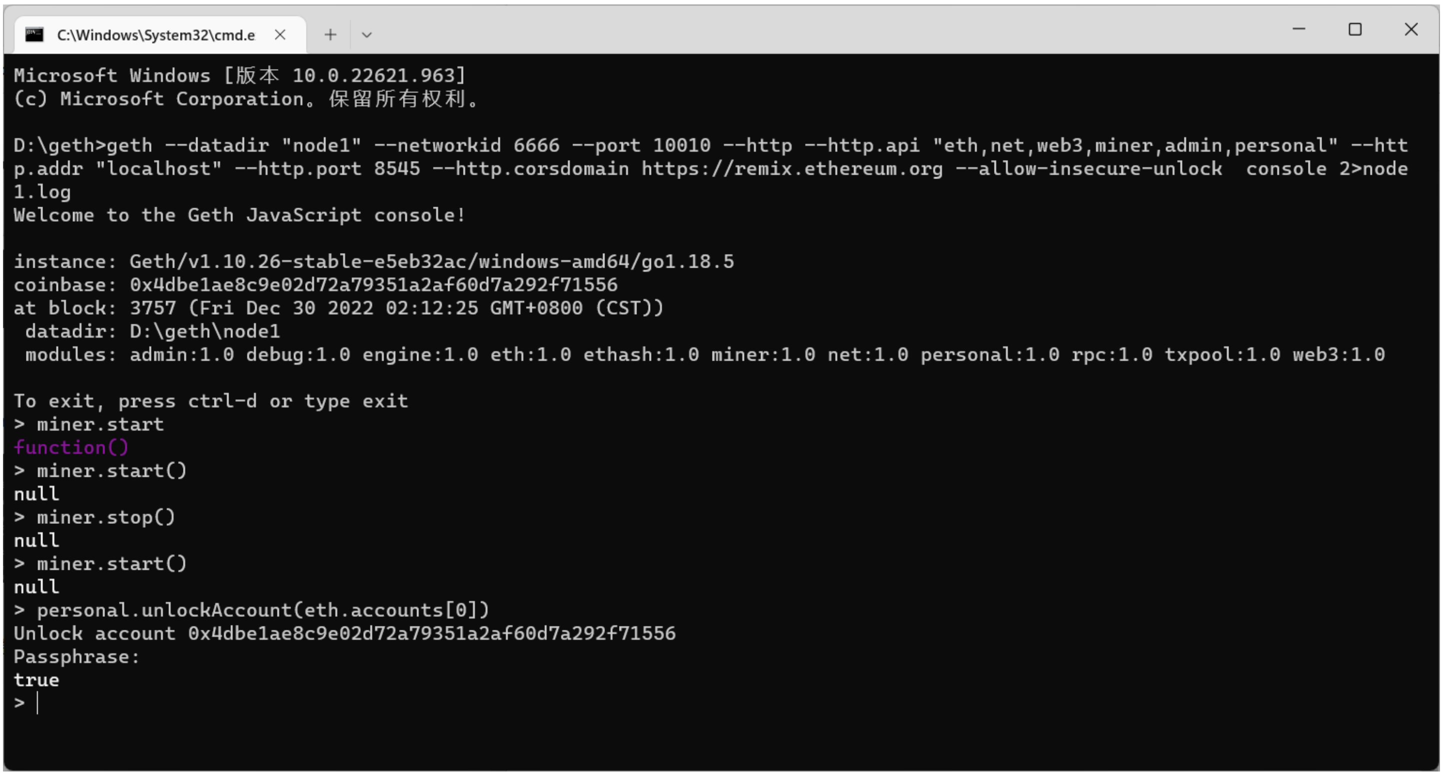
pkg 指定输出文件的包名，也就是package 名称

type 指定合约结构体名称

out 指定输出go文件名称

三、remix连接私有链

1.geth启动本地私有链



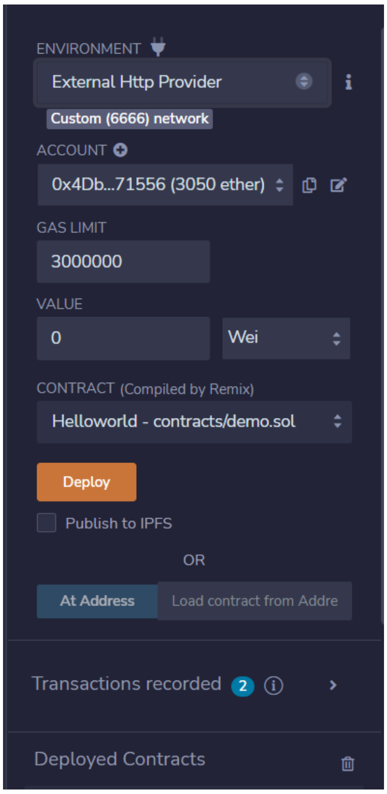
```
Microsoft Windows [版本 10.0.22621.963]
(c) Microsoft Corporation。保留所有权利。

D:\geth>geth --datadir "node1" --networkid 6666 --port 10010 --http --http.api "eth,net,web3,miner,admin,personal" --htt
p.addr "localhost" --http.port 8545 --http.corsdomain https://remix.ethereum.org --allow-insecure-unlock  console 2>node
1.log
Welcome to the Geth JavaScript console!

instance: Geth/v1.10.26-stable-e5eb32ac/windows-amd64/go1.18.5
coinbase: 0x4dbe1ae8c9e02d72a79351a2af60d7a292f71556
at block: 3757 (Fri Dec 30 2022 02:12:25 GMT+0800 (CST))
 datadir: D:\geth\node1
 modules: admin:1.0 debug:1.0 engine:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
> miner.start
function()
> miner.start()
null
> miner.stop()
null
> miner.start()
null
> personal.unlockAccount(eth.accounts[0])
Unlock account 0x4dbe1ae8c9e02d72a79351a2af60d7a292f71556
Passphrase:
true
> |
```

2.remix连接私有链



ENVIRONMENT
External Http Provider
Custom (6666) network

ACCOUNT
0x4Db...71556 (3050 ether)

GAS LIMIT
3000000

VALUE
0 Wei

CONTRACT (Compiled by Remix)
Helloworld - contracts/demo.sol

四、合约部署

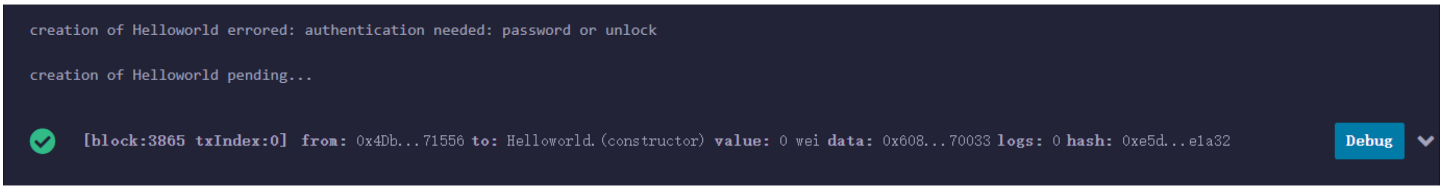
1.连接私有链：选择Web3 Provider，默认端口号时8545，如果geth启动的rpc端口号不一致，改成一致

2.部署合约

可能会出现下面几种错误，依此列出解决办法

1 .Error: authentication needed: password or unlock：这种报错解锁下账户即可

账户解锁：personal.unlockAccount(“账户地址”)



```
creation of Helloworld errored: authentication needed: password or unlock

creation of Helloworld pending...

[block:3865 txIndex:0] from: 0x4Db...71556 to: Helloworld.(constructor) value: 0 wei data: 0x608...70033 logs: 0 hash: 0xe5d...e1a32
```

五、go调用智能合约

```
package main

import (
    "fmt"
    "github.com/ethereum/go-ethereum/common"
    "github.com/ethereum/go-ethereum/contracts/checkpointoracle/contract"
    "github.com/ethereum/go-ethereum/ethclient"
)

func main() {

    // 连接rpc
    client,err := ethclient.Dial("http://127.0.0.1:8545")

    // 定义要操作合约的账户地址
    addr := "0x4dbe1ae8c9e02d72a79351a2af60d7a292f71556"
    // 将字符串地址转为common.Address
    common_addr := common.HexToAddress(addr)

    if err != nil {
        panic("连接以太坊合约出错")
    }

    // 创建合约对象
    //contract_obj,err11 := contract.NewTestAddress(common_addr,client)
    contract_obj,err11 := contract.NewCheckpointOracleCaller(common_addr,client)
    if err11 !=nil {
        panic("创建合约对象出错")
    }

    fmt.Println(contract_obj.GetAllAdmin) // Caller访问函数
    fmt.Println(contract_obj.GetLatestCheckpoint) // Transactor 有函数

}
```