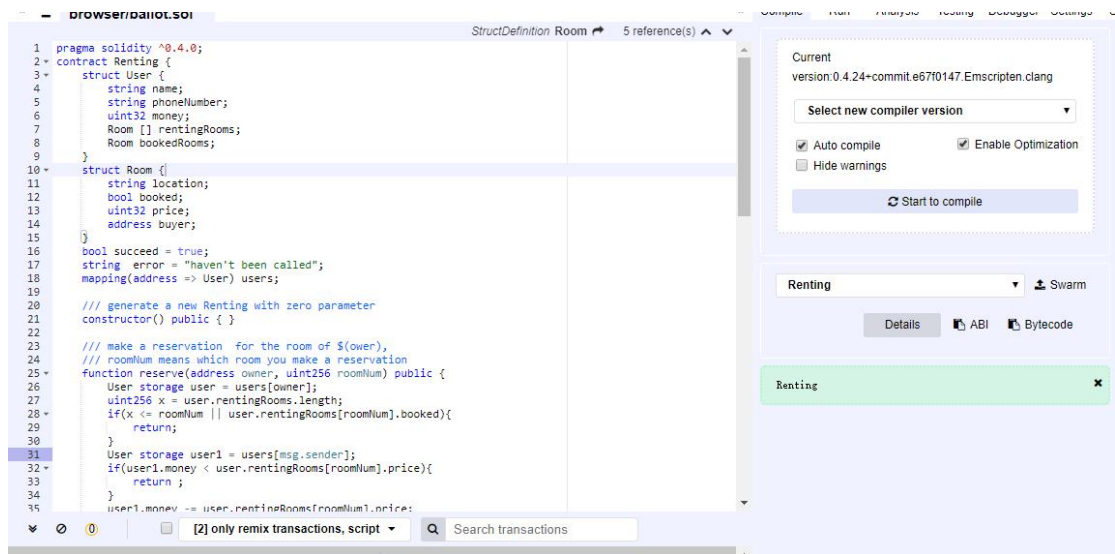
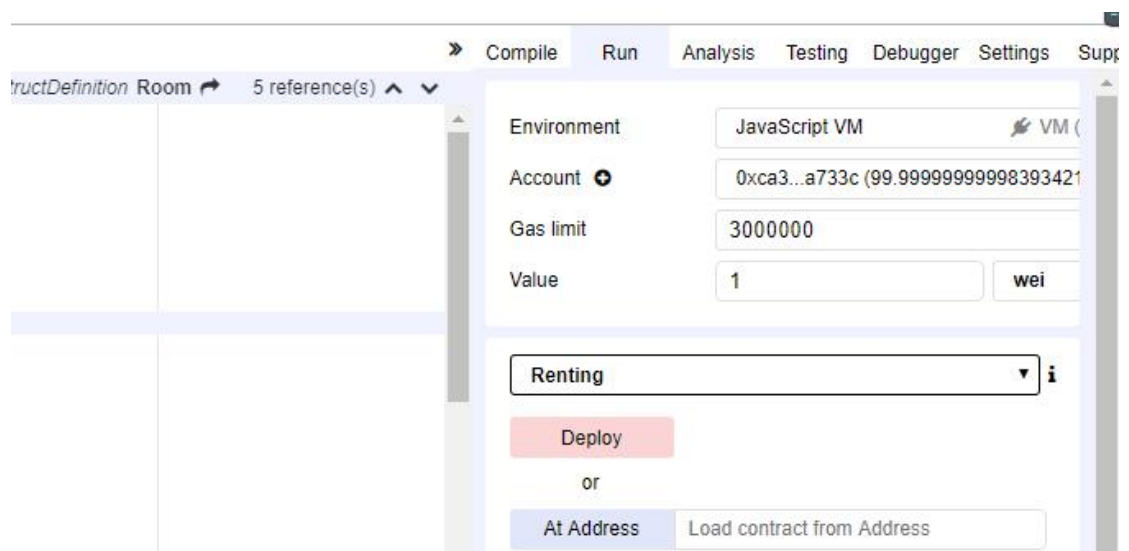


一、编写合约并调试

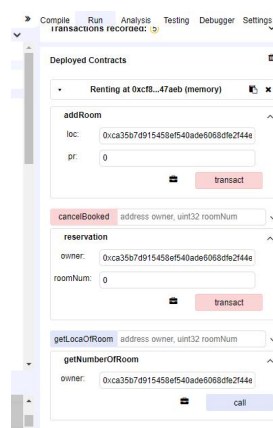
(1)、在 remix IDE 中编写智能合约，编辑完成后进行编译并且部署测试。



(2)、在 remix IDE 中进行部署测试



选中上图 run 点击 Deploy 进行部署;



向下移动滚轮，可以看到部署的合约，以及合约中的函数，在其中输入参数进行

```

1 pragma solidity ^0.4.0;
2 contract Renting {
3     struct User {
4         string name;
5         string phoneNumber;
6         uint32 money;
7         Room [] rentingRooms;
8         Room bookedRooms;
9     }

```

Environment

Account: 0xca3...a733c (99.9999999999839342)

Gas limit: 3000000

Value: 1 wei

2) only remix transactions, script

Search transactions

Debug

✓ [VM] from 0xca3...a733c to: Renting (constructor) value: 1 wei data: 0x00...0000 logs: 0 hash: 0xcfd...bec8b

status	0x1 Transaction mined and execution succeed	
transaction hash	0xcfd0ee7269d7746f418c1dabf2521d02d2f4b4d2856434b77387f3643abeec8b	
contract address	0xcab77b4b9bfb9b92a53572091e5798e570051ba8f	
from	0xca35b7d915458e540ade6068dfe2f44e8fa733c	
to	Renting (constructor)	
gas	3000000 gas	
transaction cost	862148 gas	
execution cost	610456 gas	
hash	0xcfd0ee7269d7746f418c1dabf2521d02d2f4b4d2856434b77387f3643abeec8b	
input	0x008...70029	
decoded input	{}	
decoded output	-	
logs	[]	
value	0 wei	

Transactions recorded: 5

Deployed Contracts

Renting at 0xca3...47aeb (memory)

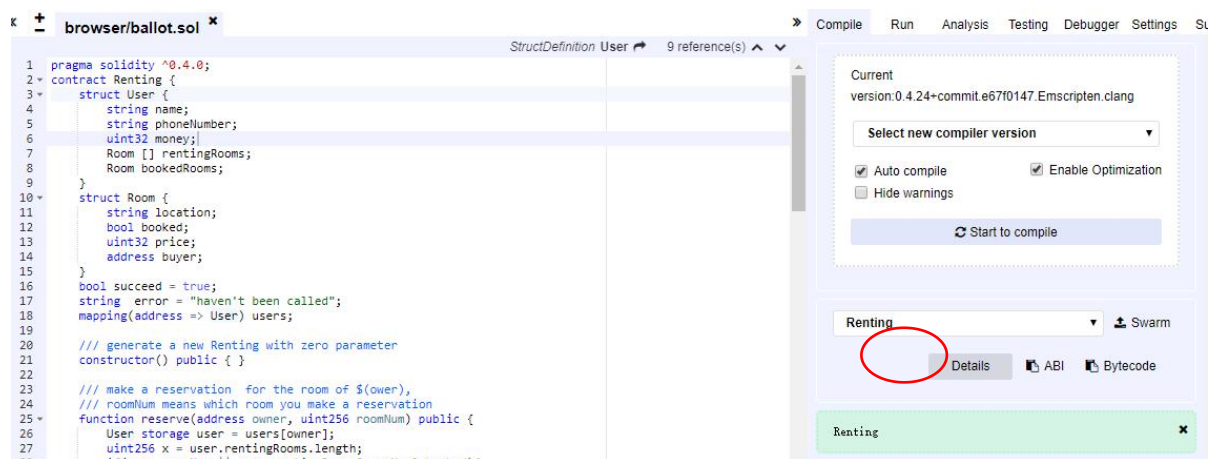
addRoom

loc: 0xca35b7d915458e540ade6068dfe2f44e

pr: 0

transact

(1)、点击下图中 Detail



The screenshot displays the Solidity IDE interface with three main panels:

- Source Code Panel (Left):** Shows the Solidity contract code. The `contract Rent` is defined with a `Room` struct containing `string`, `uint3`, and `Room` fields. A `Room` struct is also defined with `string`, `bool`, `uint3`, and `address` fields. A `bool succ` variable and a `string e` are declared. A `mapping(a)` is defined. A `function` is defined with parameters `User`, `uint2`, and `if(x)`. The `function` is marked as `public` and `payable`. The `function` is defined with parameters `User`, `uint2`, and `if(x)`.
- Bytecode Panel (Middle):** Displays the compiled bytecode. The `linkReferences` are empty. The `object` is a hex string. The `opcodes` are a list of assembly instructions. The `sourceMap` is a string.
- ABI Panel (Right):** Displays the ABI of the contract. The `ABI` is a list of functions and events.

The `contract Rent` is defined with a `Room` struct containing `string`, `uint3`, and `Room` fields. A `Room` struct is also defined with `string`, `bool`, `uint3`, and `address` fields. A `bool succ` variable and a `string e` are declared. A `mapping(a)` is defined. A `function` is defined with parameters `User`, `uint2`, and `if(x)`. The `function` is marked as `public` and `payable`. The `function` is defined with parameters `User`, `uint2`, and `if(x)`.

三、在 geth 客户端进行合约函数的调用

(1)、在 geth 客户端中输入获取到的合约实例，来查看合约中的函数

```
stateMutability: "view",
type: "function"
}, {
  constant: false,
  inputs: [...], [...]],
  name: "reserve",
  outputs: [],
  payable: false,
  stateMutability: "nonpayable",
  type: "function"
}, {
  inputs: [],
  payable: false,
  stateMutability: "nonpayable",
  type: "constructor"
}],
address: "0x4eabcc393ebe0356f7520a552f89fc9268083ec5",
transactionHash: "0xd4143472401bea911b4bab75600d62b23e34c7433735014e675cffff45d20cdf",
addRoom: function(),
allEvents: function(),
cancelBooked: function(),
getError: function(),
getLocaOfRoom: function(),
getNumberOfRoom: function(),
getPriceOfRoom: function(),
getSucceed: function(),
isBooked: function(),
reserve: function()
```

从上图中可以看到，我部署的这个合约的地址，以及合约中的一些函数。

(2)、调用一些函数

调用 `getNumberOfRoom()` 函数（[查询类函数](#)）来查看当前账户出租的房间个数：

```
> renting.getNumberOfRoom(eth.accounts[0])
0
```

调用 `addRoom()` 函数（[修改数据类的函数](#)）来给当前账户添加一个出租房间，需要将地理位置以及价格作为参数传入，要记得解锁账户：

```
> personal.unlockAccount(eth.accounts[0])
Unlock account 0xa9d4e2316cc4af531fddd44bb2c42631a460f5f3
Passphrase:
true
> renting.addRoom("xxxx", 0, {from: eth.accounts[0]})
INFO [11-26|20:42:43] Submitted transaction          fullhash=0xf96b9e310d5526402e441e3e0ec923e4d07a94d9a04df0f8d3b1eeaf321c8079 recipient=0x4eabcc393EBE0356F7520a552F89fC9268083ec5
"0xf96b9e310d5526402e441e3e0ec923e4d07a94d9a04df0f8d3b1eeaf321c8079"
> miner.start()
INFO [11-26|20:42:49] Updated mining threads          threads=0
INFO [11-26|20:42:49] Transaction pool price threshold updated price=18000000000
InNum101
[> 11-26|20:42:49] Starting mining operation
INFO [11-26|20:42:49] Commit new mining work          number=11 txs=1 uncles=0 elapsed=999µs
INFO [11-26|20:42:50] Successfully sealed new block   number=11 hash=0a38bf...12c33a
INFO [11-26|20:42:50] □□ mined potential block       number=11 hash=0a38bf...12c33a
INFO [11-26|20:42:50] Commit new mining work          number=12 txs=0 uncles=0 elapsed=500.8µs
INFO [11-26|20:42:50] Successfully sealed new block   number=12 hash=6e5270...0a9291
INFO [11-26|20:42:50] □□ mined potential block       number=12 hash=6e5270...0a9291
INFO [11-26|20:42:50] Commit new mining work          number=13 txs=0 uncles=0 elapsed=0s
> miner.stop()
true
> renting.getNumberOfRoom(eth.accounts[0])
```

调用之后还要进行挖矿才会生效，调用完成后，当前账户的出租房间数由 0 变为了

1。