

Bank Contract Deploy

The screenshot displays the Remix IDE interface for deploying a Solidity contract. The browser address bar shows the URL: `remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`.

Left Sidebar (DEPLOY & RUN TRANSACTIONS):

- ENVIRONMENT:** Injected Web3, Custom (5777) network.
- ACCOUNT:** 0x034...7902c (96.8635756 ETH).
- GAS LIMIT:** 3000000.
- VALUE:** 0 Wei.
- CONTRACT:** Bank - Bank.sol.
- Buttons:** Deploy, Publish to IPFS.
- Transactions recorded:** 1.
- Deployed Contracts:** (Empty list).

Central Code Editor (Bank.sol):

```
24
25
26 function transfer (uint amount, address acctToTransferTo) external{
27     require(account_balance[msg.sender]>=amount);
28     account_balance[msg.sender] -= amount;
29     account_balance[acctToTransferTo] += amount;
30 }
31
32 function bankBalance() external view returns(uint){return address(this).balance;}
33
34 function withdrawl(uint amount) external {
35     require (account_balance[msg.sender]>=amount);
36
37     account_balance[msg.sender] -= amount;
38     payable(msg.sender).transfer(amount);
39 }
40
41 receive () external payable {
42     account_balance[msg.sender] += msg.value;
43 }
```

Right Sidebar (CONTRACT DEPLOYMENT):

Account 2 → New Contract

<https://remix.ethereum.org>

CONTRACT DEPLOYMENT

DETAILS DATA

Estimated gas fee: 0.0072548 0.007255 ETH

Site suggested Max fee: 0.0072548 ETH

Total: 0.0072548 0.0072548 ETH

Amount + gas fee Max amount: 0.0072548 ETH

Buttons: Reject, Confirm

Console Log:

- creation of Bank pending...
- creation of Bank errored: MetaMask Tx Signature: User denied transaction signature.
- creation of Bank errored: MetaMask Tx Signature: User denied transaction signature.
- creation of Bank pending...

Send 20ETH

DEPLOY & RUN TRANSACTIONS

At Address

Load contract from Address

Transactions recorded 2

Deployed Contracts

BANK AT 0X1BA...EA529 (BLOCKCHAIN)

transfer

uint256 amount, address acc

withdrawal

uint256 amount

account_balan...

address

0: uint256: 20000000000000000000

bankBalance

0: uint256: 0

Low level interactions

CALLDATA

Transact

Bank.sol

```
24
25 function transfer (uint amount, address acctToTransferTo) external{
26     require(account_balance[msg.sender]>=amount);
27     account_balance[msg.sender] -= amount;
28     account_balance[acctToTransferTo] += amount;
29 }
30
31 function bankBalance() external view returns(uint){return address(this).balance;}
32
33 function withdrawl(uint amount) external {
34     require (account_balance[msg.sender]>=amount);
35
36     account_balance[msg.sender] -= amount;
37     payable(msg.sender).transfer(amount);
38 }
39 receive () external payable {
40     account_balance[msg.sender] += msg.value;
41 }
42
43 }
```

listen on network

Search with transaction hash or address

call to Bank.account_balance

[call] from: 0x034d9CD8e2d5e83bDAeD2851Ca3C131B0187902c to: Bank.account_balance(address) data: 0xb7b...7902c

Account 2

Account 2

SENDING

20

Estimated gas

0.00042 0.00042 ETH

fee

Max fee: 0.00042 ETH

Total

20.00042 20.00042 ETH

Amount + gas fee

Max amount: 20.00042 ETH

Reject

Confirm

Debug

Test the Balance

The screenshot shows a web browser window with the URL `remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.0+commit.c7dfd78e.js`. The browser's address bar and tabs are visible at the top. The main content area is the Remix IDE, which is divided into three main sections:

- Left Sidebar (DEPLOY & RUN TRANSACTIONS):** This panel contains options for deploying and running transactions. It includes a "Publish to IPFS" checkbox, a "Load contract from Address" button, and a list of "Deployed Contracts". The first contract listed is "BANK AT 0X1BA...EA529 (BLOCKCHAIN)". Below this, there are buttons for "transfer", "withdrawal", "account_balance", and "bankBalance". The "account_balance" button is currently selected, showing a transaction hash and a balance of 0.
- Main Editor:** This section displays the Solidity code for the "Bank.sol" contract. The code includes functions for `transfer`, `bankBalance`, `withdrawal`, and a `receive` function. The code is as follows:

```
24
25
26 function transfer(uint amount, address acctToTransferTo) external{
27     require(account_balance[msg.sender]>=amount);
28     account_balance[msg.sender] -= amount;
29     account_balance[acctToTransferTo] += amount;
30 }
31
32 function bankBalance() external view returns(uint){return address(this).balance;}
33
34 function withdrawal(uint amount) external {
35     require (account_balance[msg.sender]>=amount);
36     account_balance[msg.sender] -= amount;
37     payable(msg.sender).transfer(amount);
38 }
39
40 receive () external payable {
41     account_balance[msg.sender] += msg.value;
42 }
43 }
```
- Bottom Panel:** This panel shows the transaction log. It includes a search bar and a list of transactions. The first transaction is a call to `Bank.account_balance` with the following details:

```
CALL [call] from: 0x034d9CD8e2d5e83bDAeD2851Ca3C131B0187902c to: Bank.account_balance(address) data: 0xb7b...7902c
```

The bottom of the screen shows the macOS dock with various application icons, including Finder, Spotlight, Safari, Calendar, Notes, System Preferences, Google Chrome, WhatsApp, Telegram, YouTube, Visual Studio Code, Word, PowerPoint, and a trash can.