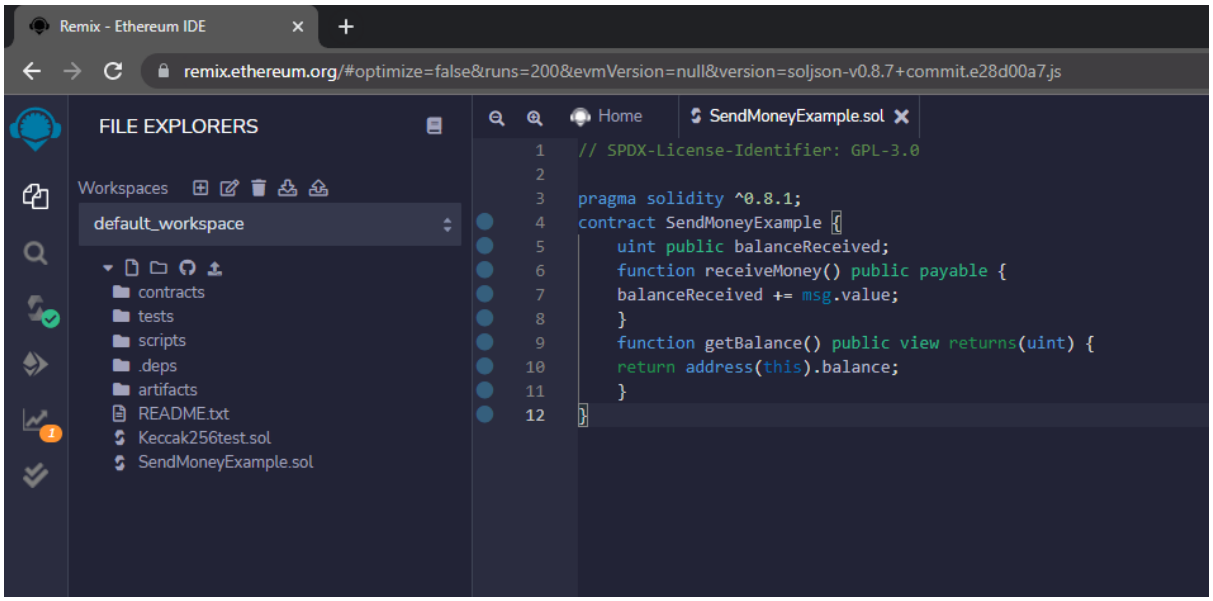
8.2 Smart Contract

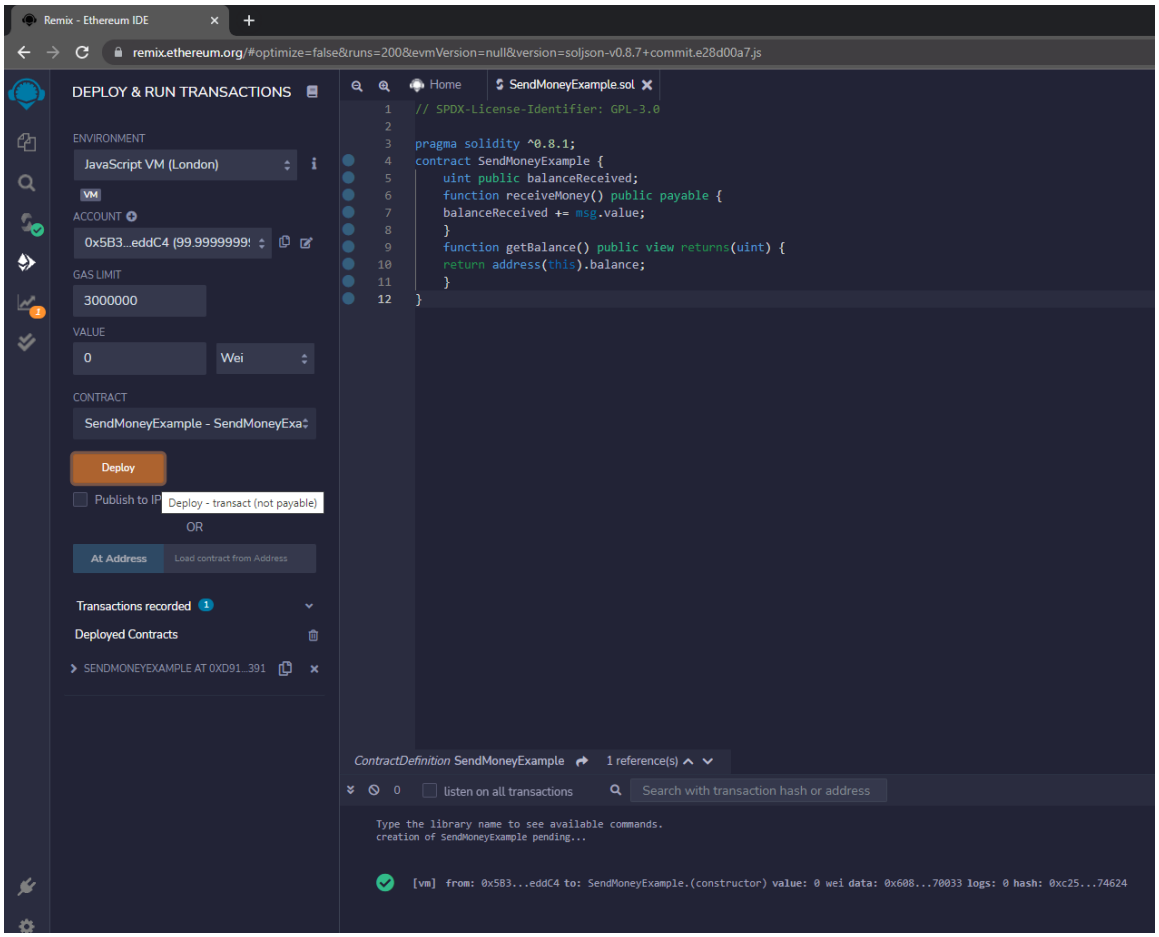
Let's start with a simple Smart Contract. Create a new file in Remix

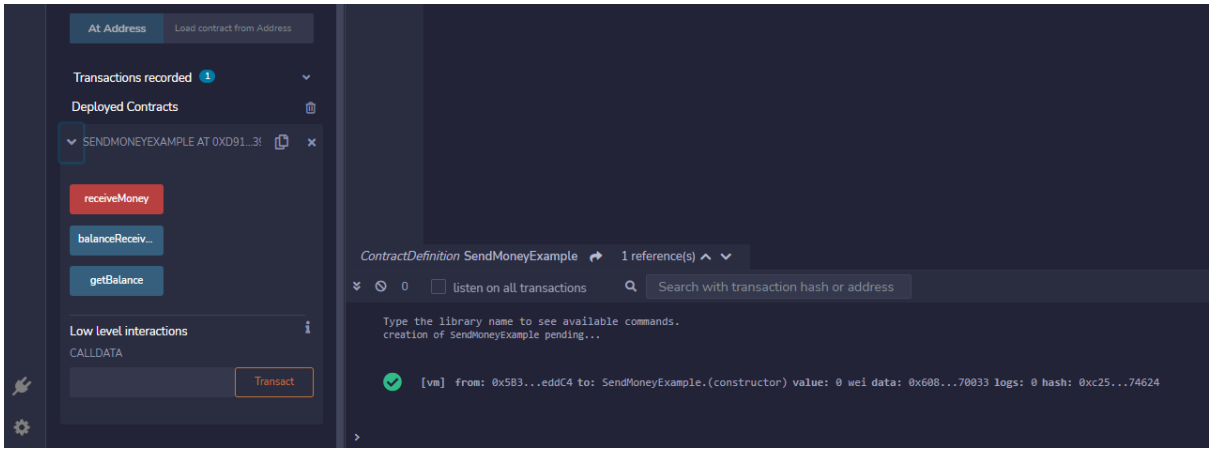


8.3 Deploy and Use the Smart Contract

8.3.1 Deploy the Smart Contract

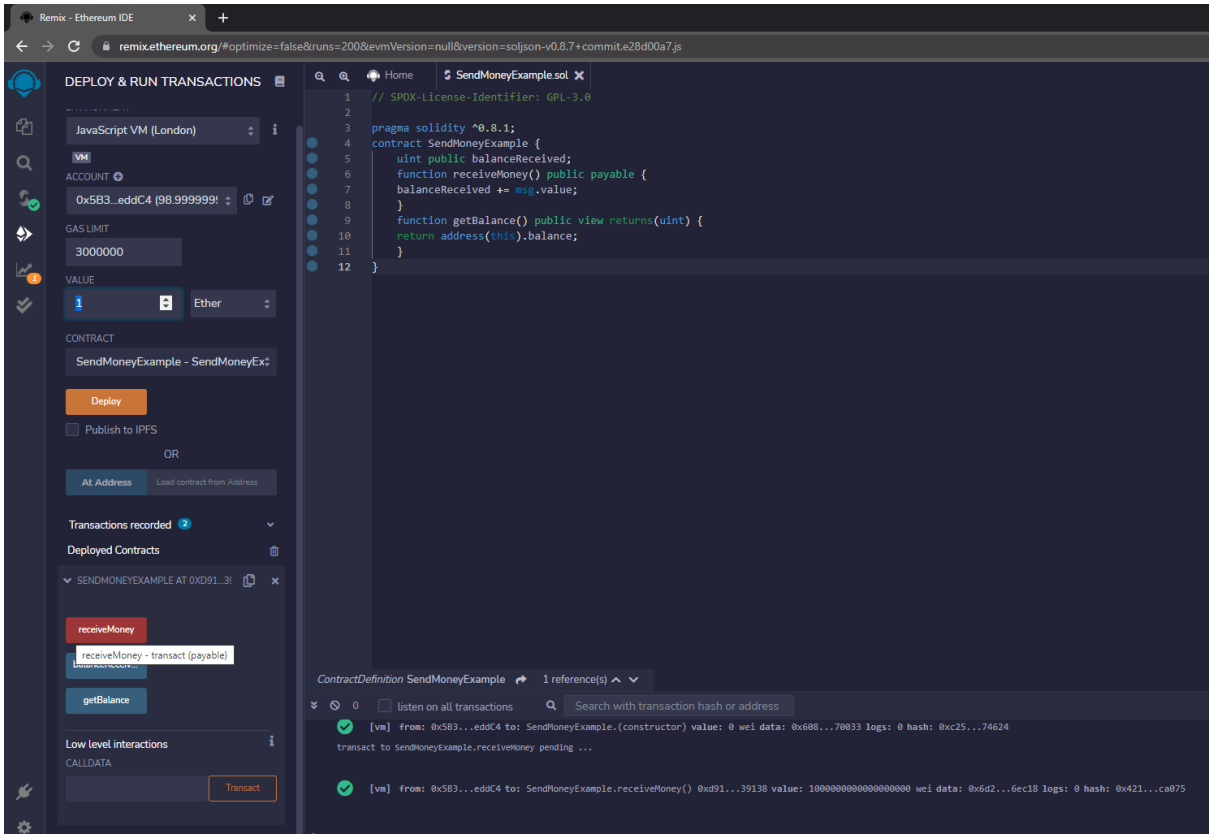
Head over to the Deploy and Run Transactions Plugin and deploy the Smart Contract into the JavaScript VM:





8.3.2 Send Ether To The Smart Contract

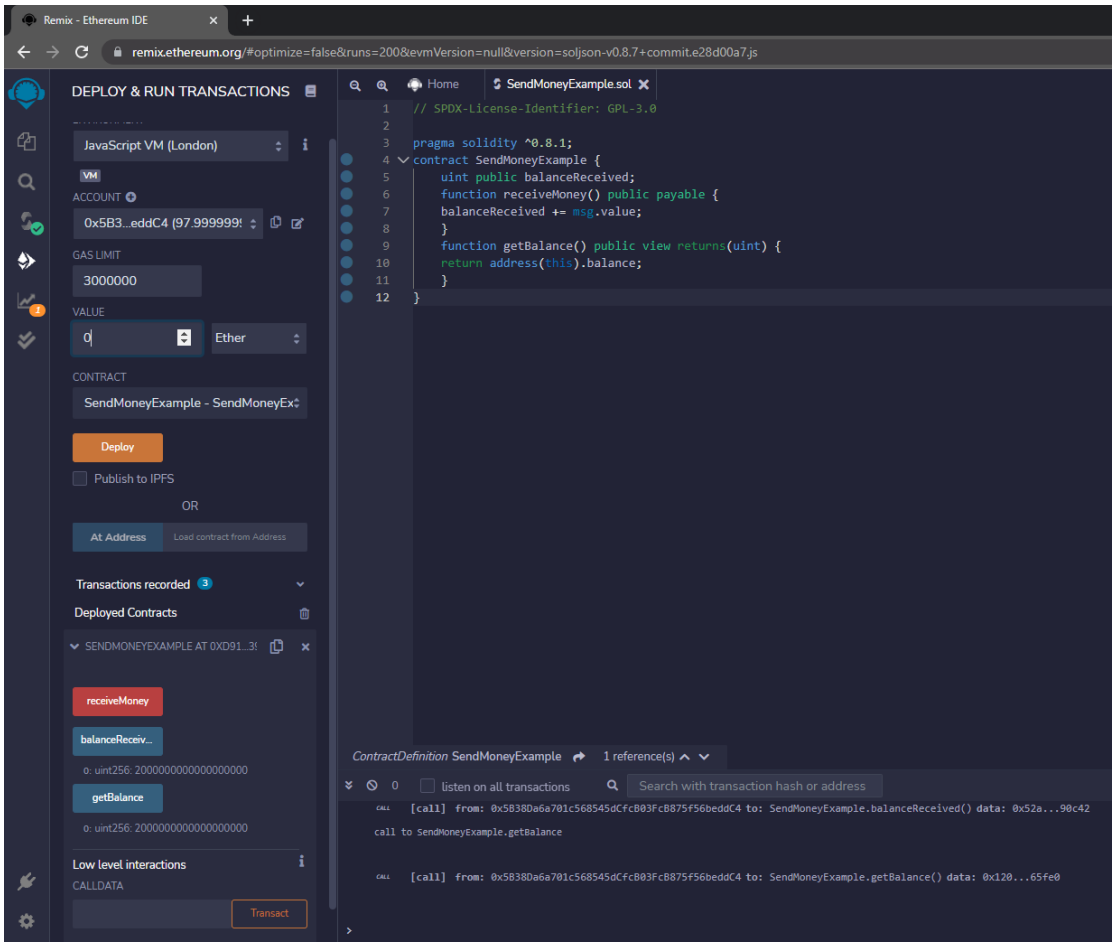
Now it is time to send some Ether to the Smart Contract! Scroll up to the "value" field and put "1" into the value input field and select "ether" from the dropdown:



Also observe the terminal , see that there was a new transaction sent to "the network" (although just a simulation in the browser, but it would be the same with a real blockchain).

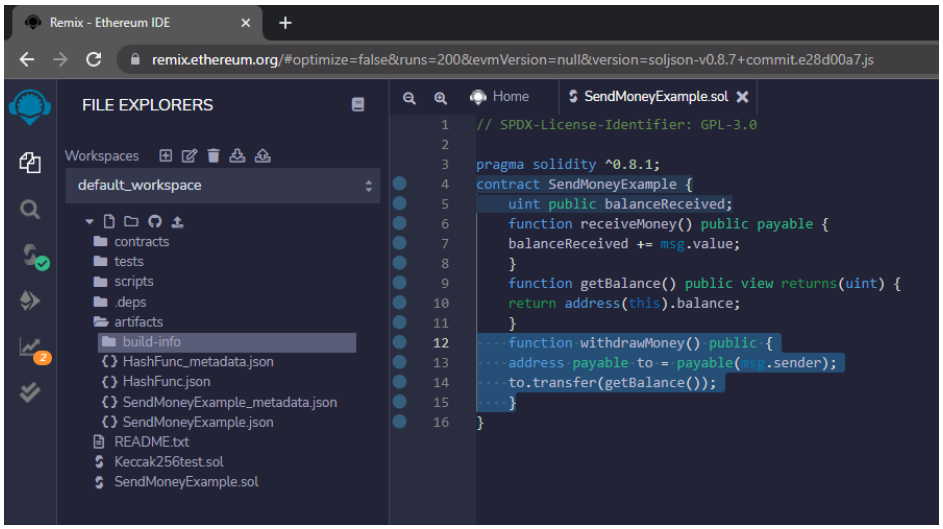
8.3.3 Check the Balance

Now we sent 1 Ether, or 10^18 Wei, to the Smart Contract. According to our code the variable balanceReceived and the function getBalance() should have the same value. And, indeed, they do:



8.4 Withdraw Ether From Smart Contract

8.4.1 Add a Withdraw Function

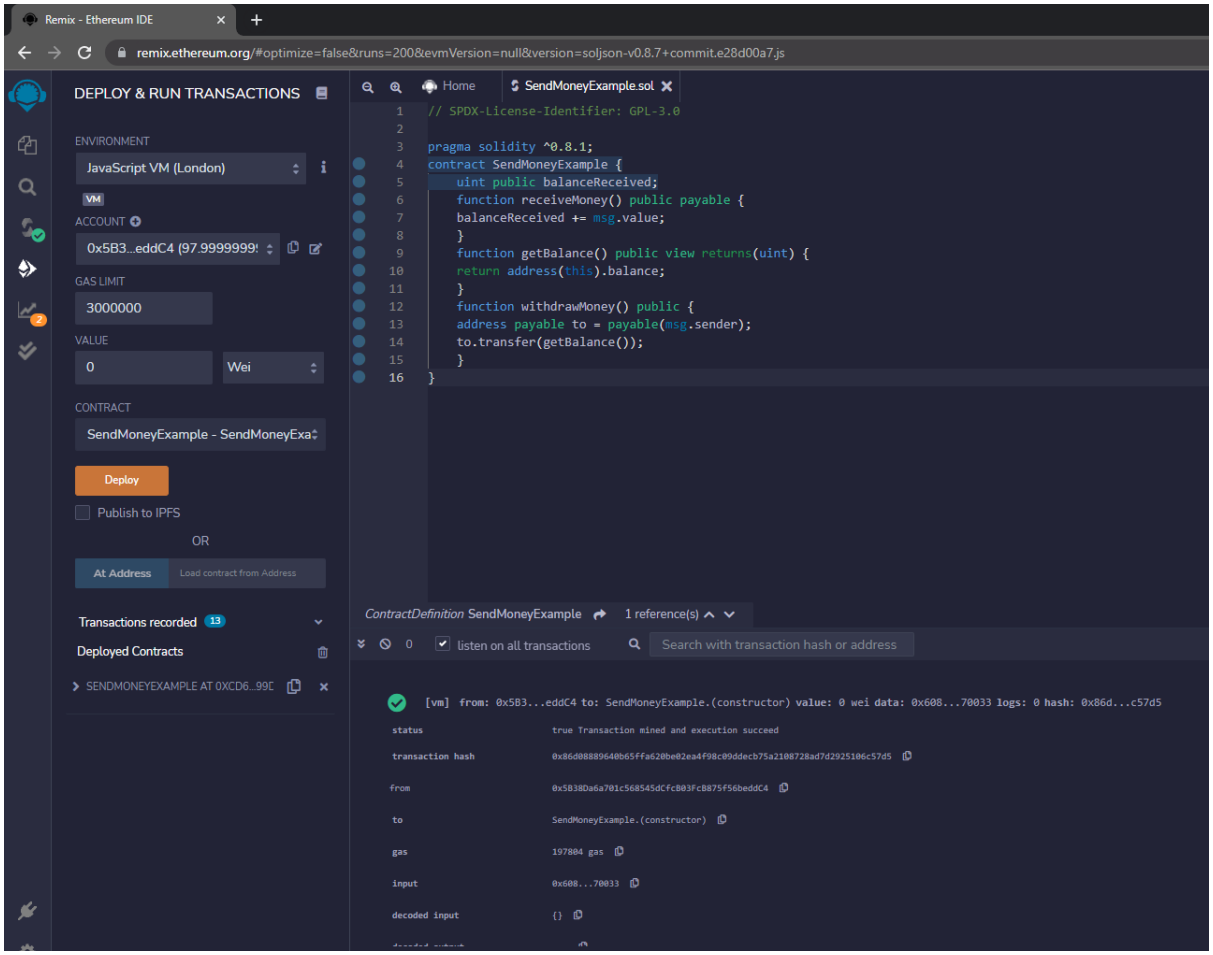


This function will send all funds stored in the Smart Contract to the person who calls the "withdrawMoney()" function.

8.4.2 Deploy the new Smart Contract

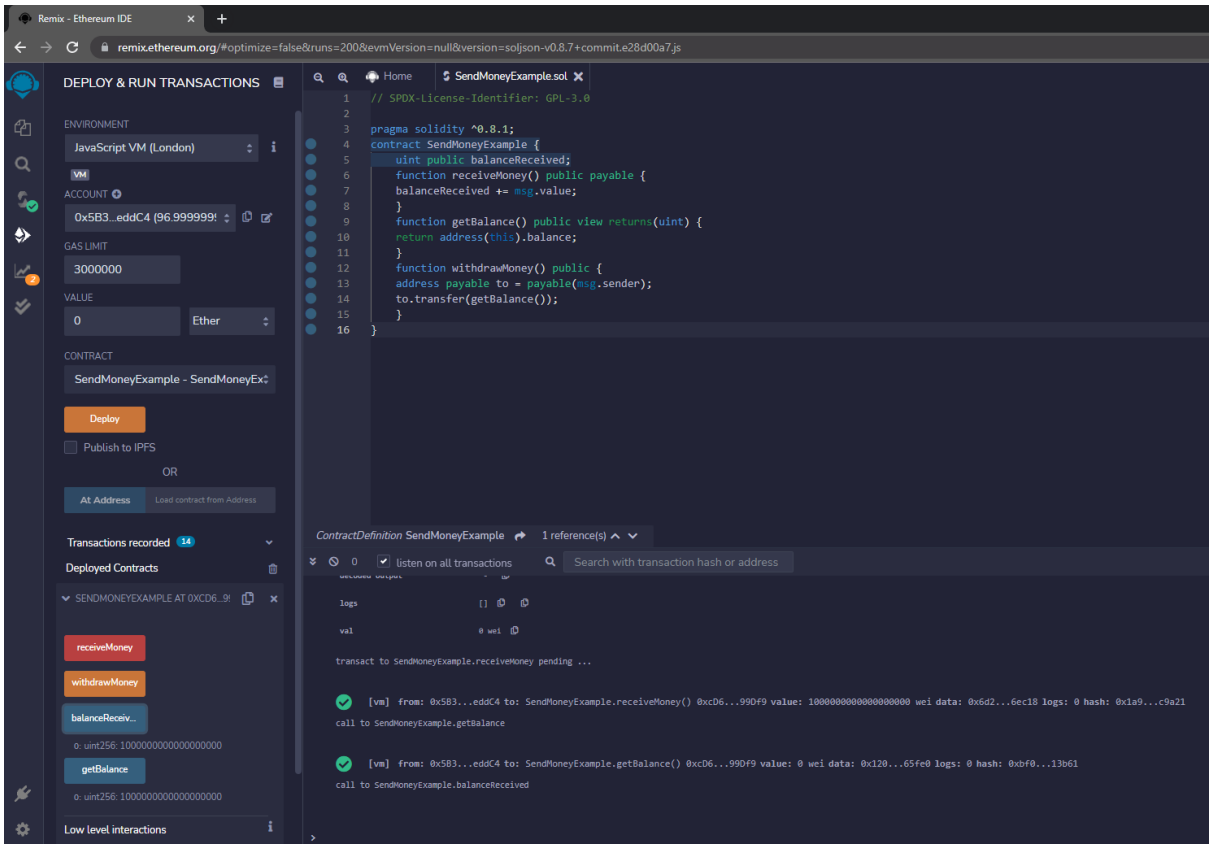
1. Deploy the new version and send again 1 Ether to the Smart Contract.

2. To avoid confusion I recommend you close the previous Instance, we won't need it anymore



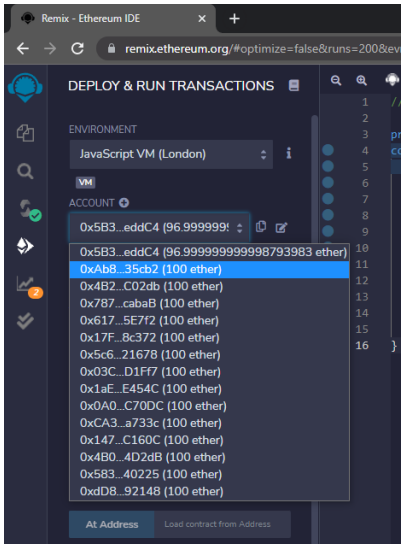
At the end you should end up with one active Instance of your Smart Contract. The same procedure as before:

1. Put in "1 Ether" into the value input box
2. hit "receiveMoney" in your new contract Instance

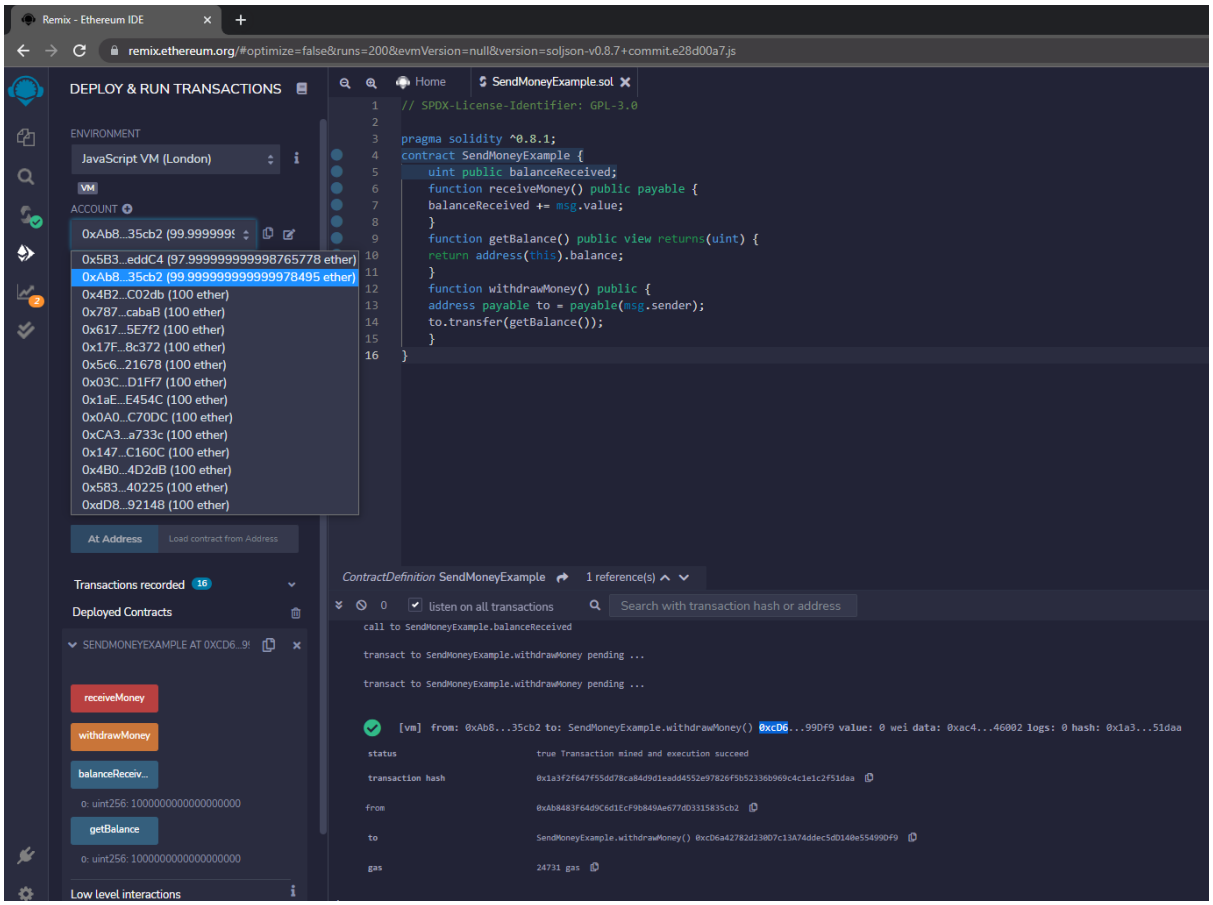


8.4.3 Withdraw Funds from the Smart Contract

Now it's time we use our new function! But to make things more exciting, we're going to withdraw to a different Account. Select the second Account from the Accounts dropdown:



Then hit the "withdrawMoney" button:

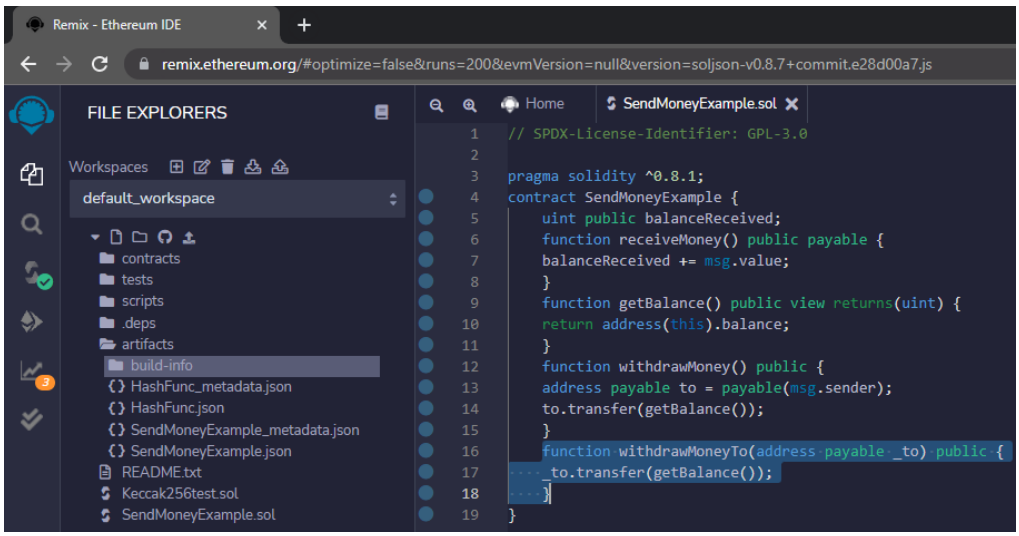


It's more than the previous 100 Ether! We got our 1 Ether through our Smart Contract into another Account!

8.5 Withdraw To Specific Account

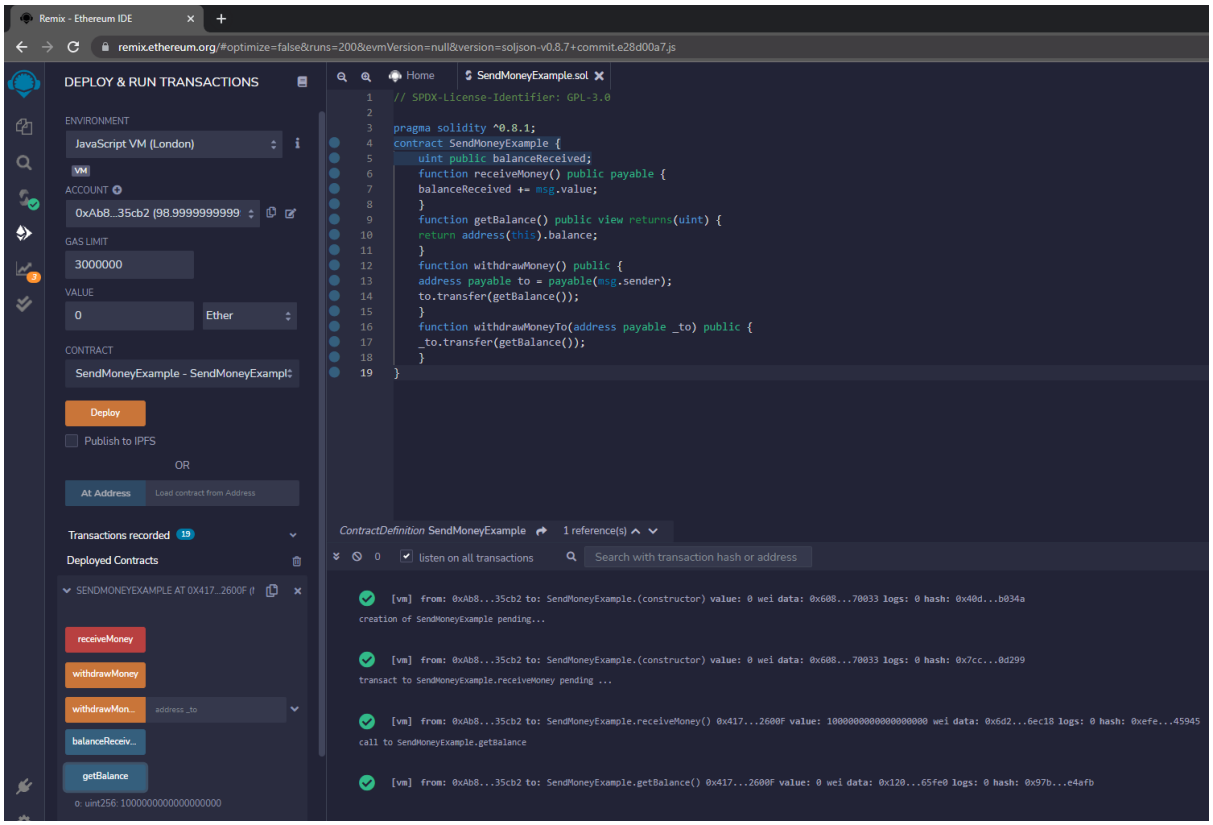
Previously we had our Smart Contract just blindly send the Ether to whoever called the Smart Contracts "withdrawMoney" function. Let's extend this a bit so that the Funds can be send to a specific Account.

As you can see, we can now specify an Address the money will be transferred to! Let's give this a try!



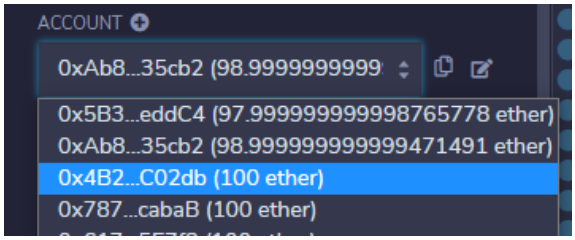
8.5.1 Redeploy our Smart Contract

Deploy the Smart Contract. Close the old Instance. Send 1 Ether to the Smart Contract (don't forget the value input field!). Make sure the Balance shows up correctly.

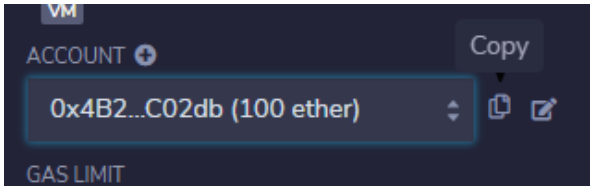


8.5.2 Test the "withdrawMoneyTo" function

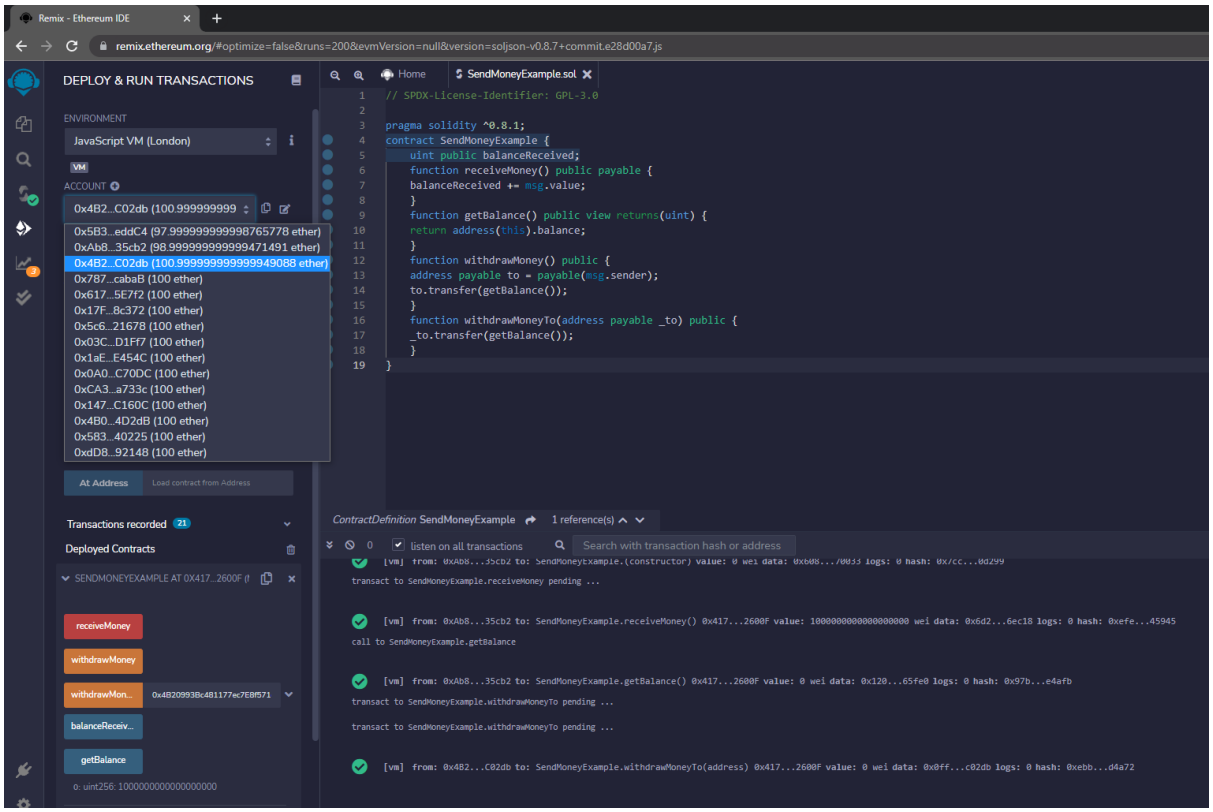
Select the third account from the dropdown



Hit the little "copy" icon:



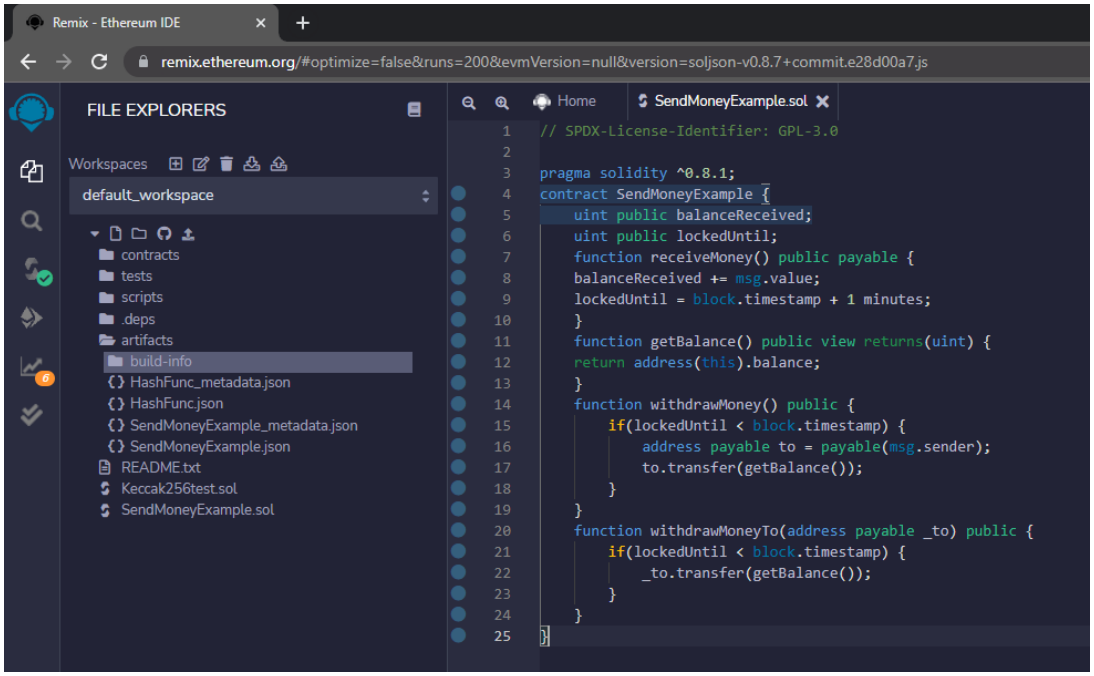
Switch back to the first Account. Paste the Account you copied into the input field next to "withdrawMoneyTo": Now open the Accounts dropdown. See the balance of your third Account?



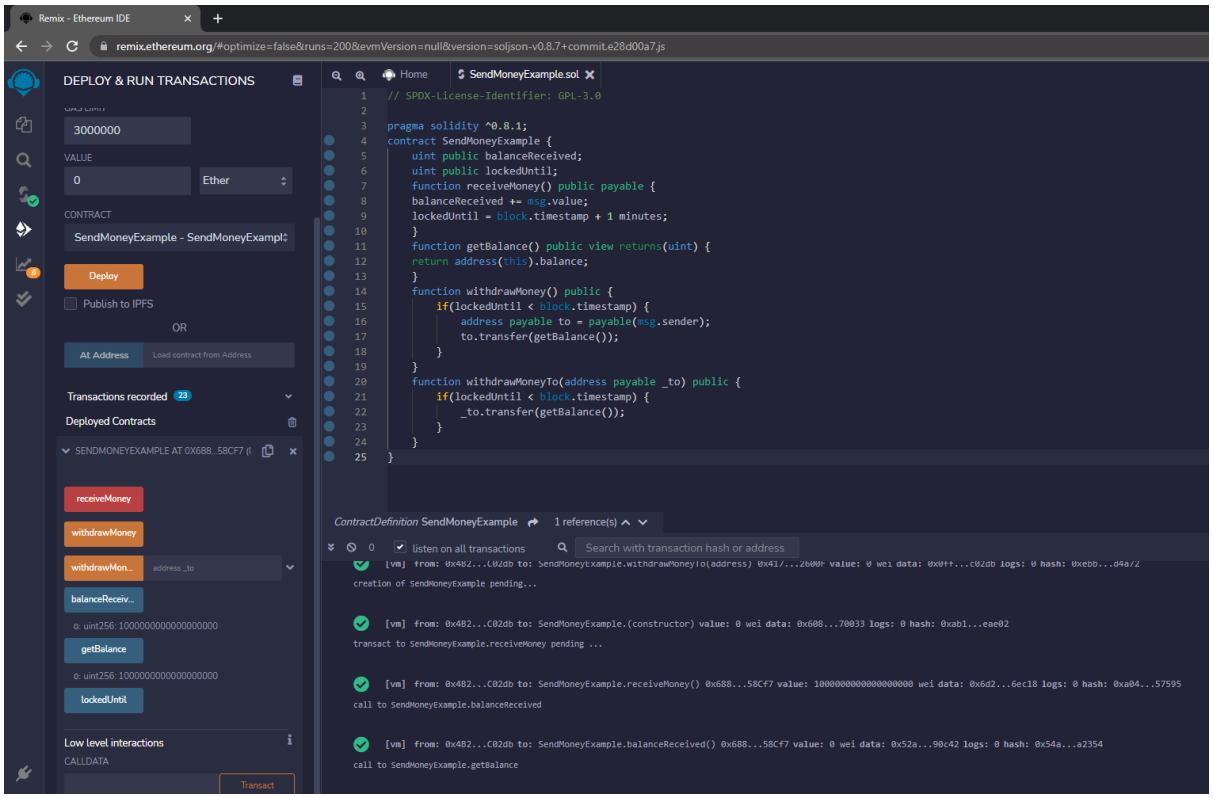
8.6 Withdrawal Locking

8.6.1 Extend the Smart Contract

What we need is to store the block.timestamp somewhere. There are several methods to go about this, I prefer to let the user know how long is it locked. So, instead of storing the deposit-timestamp, I will store the lockedUntil timestamp. Let's see what happens here:



8.6.2 Deploy and Test the Smart Contract



Click "withdrawMoney" - and nothing happens. The Balance stays the same until 1 Minute passed since you hit "receiveMoney".