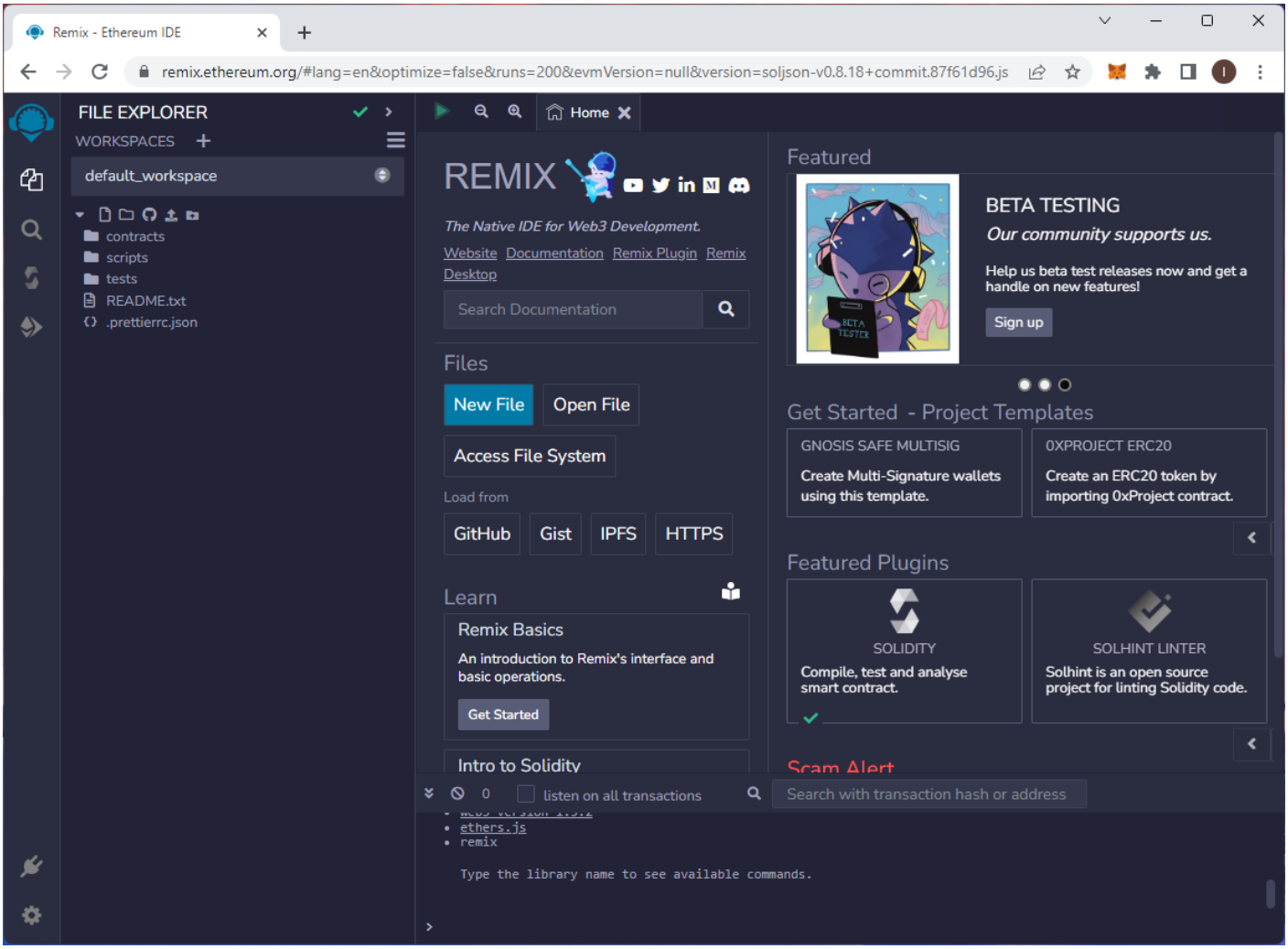
Development

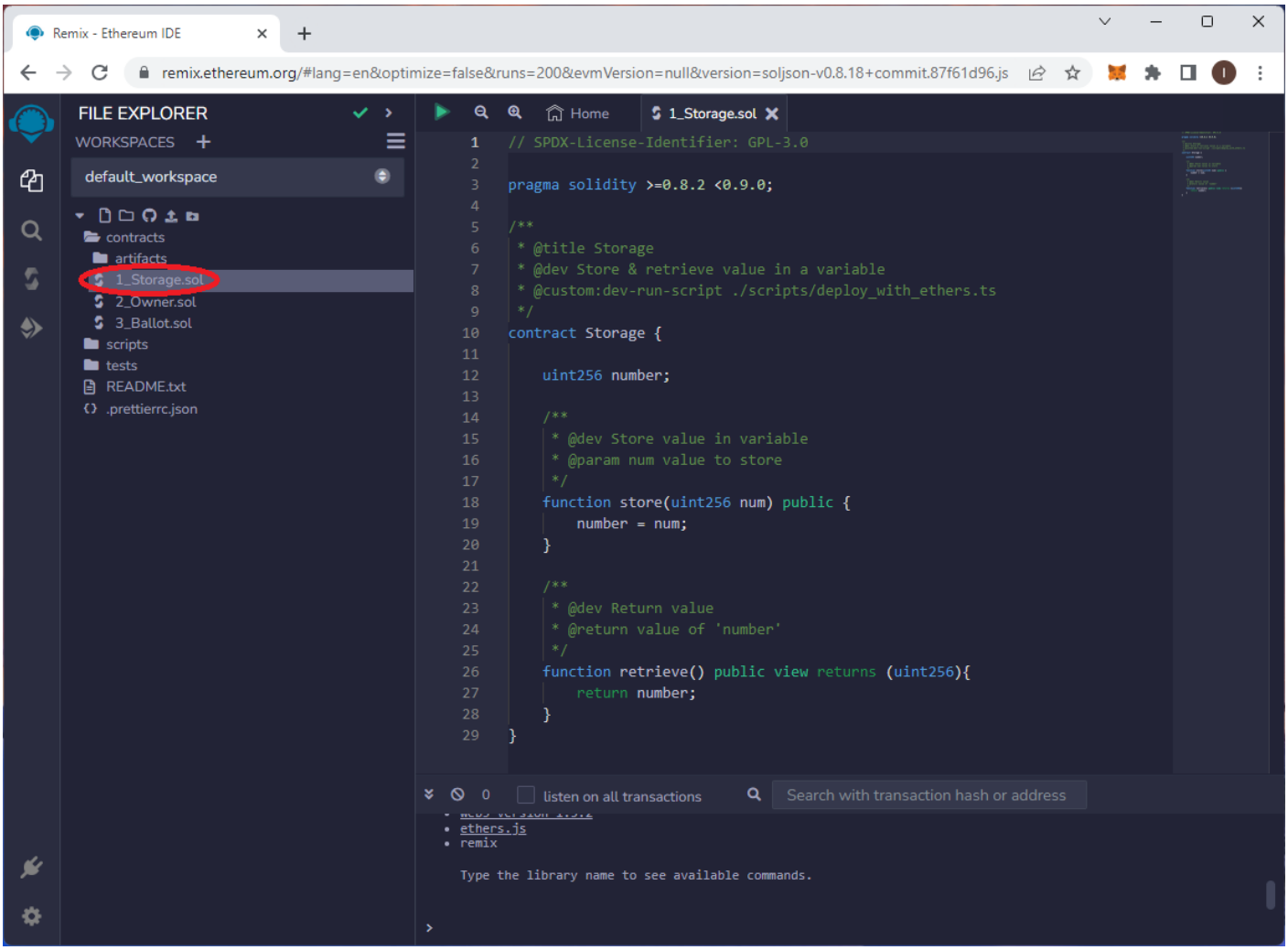
- Deploying a smart contract on BPX mainnet using Remix IDE

Deploying a smart contract on BPX mainnet using Remix IDE

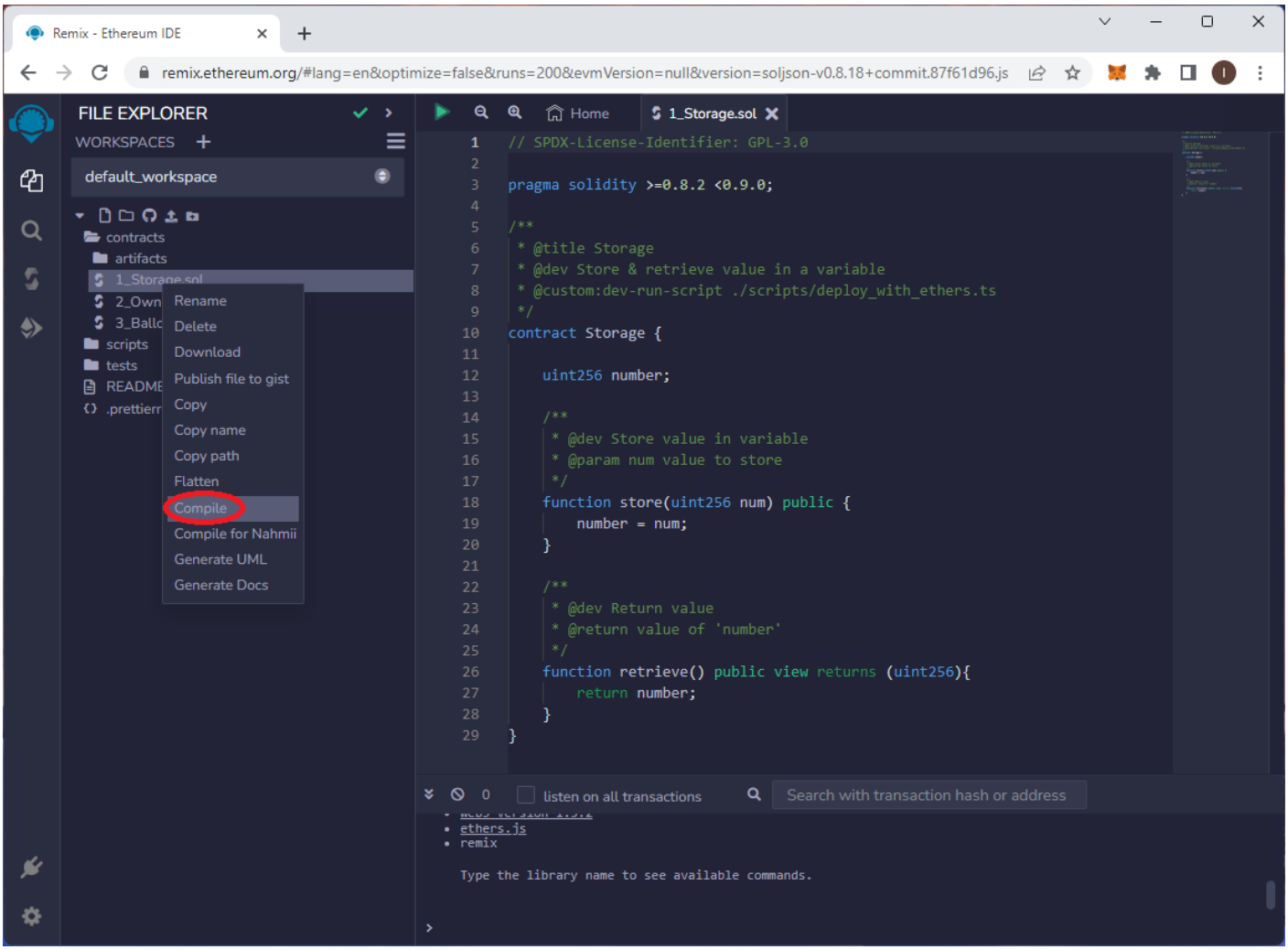
1. Open Remix IDE.



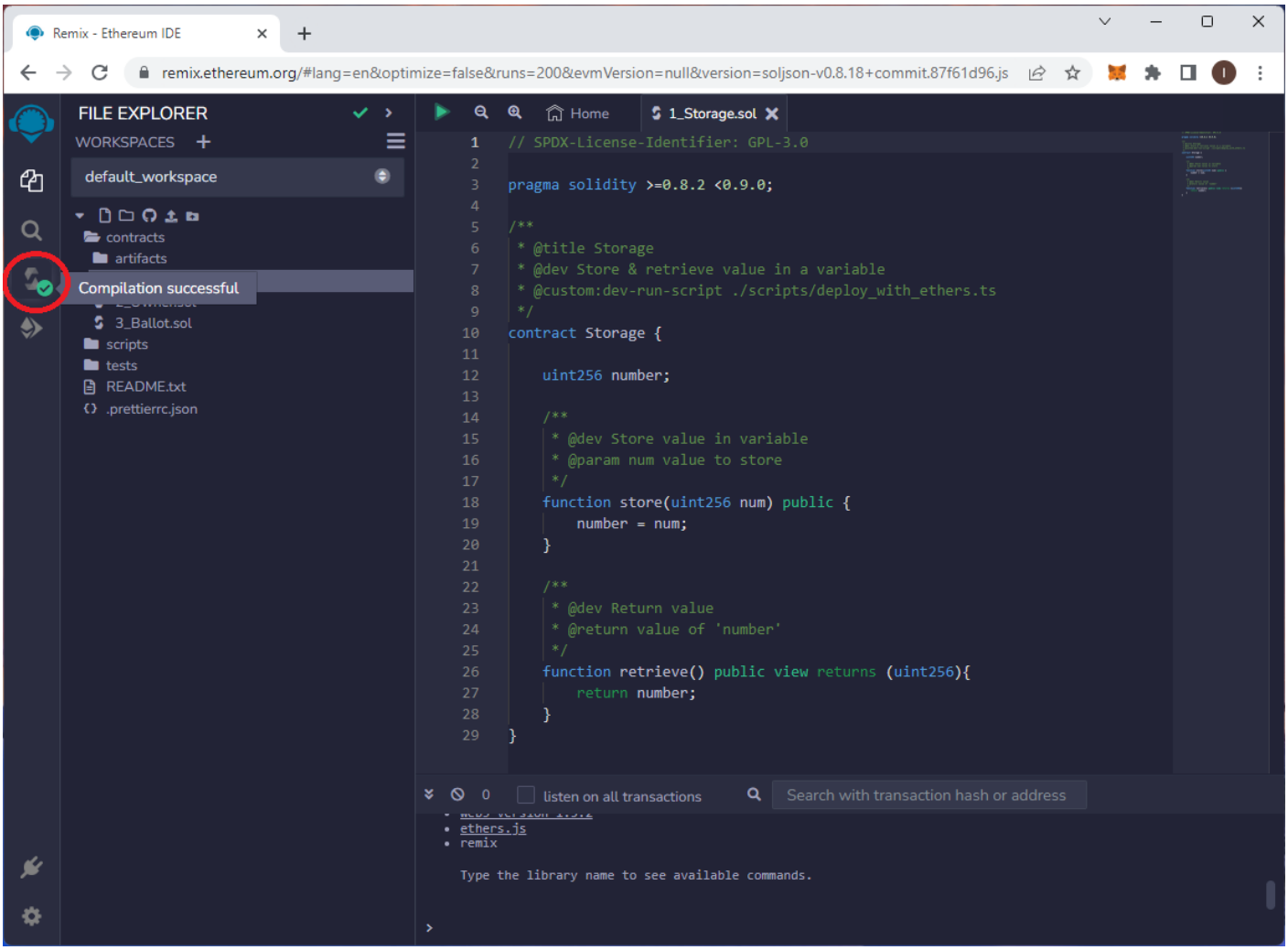
2. In our example, we will use one of the sample contracts from the IDE. This contract provides two functions: the first, `store`, allows you to save any number in the contract, while the second, `retrieve`, enables you to read the stored number. Open the `contracts/1_Storage.sol` file.



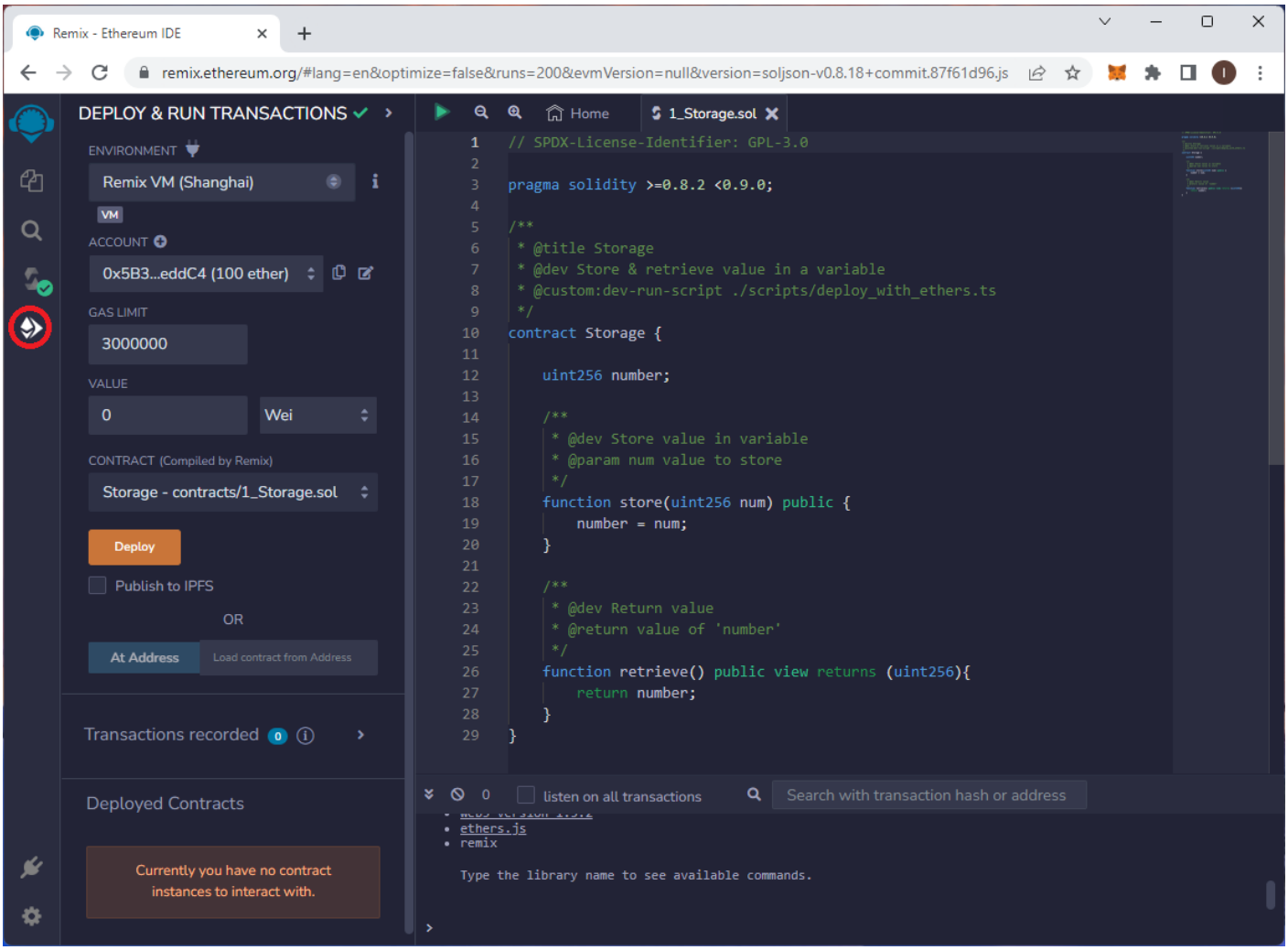
3. Compile the contract by right-clicking on the file name and selecting **Compile**.



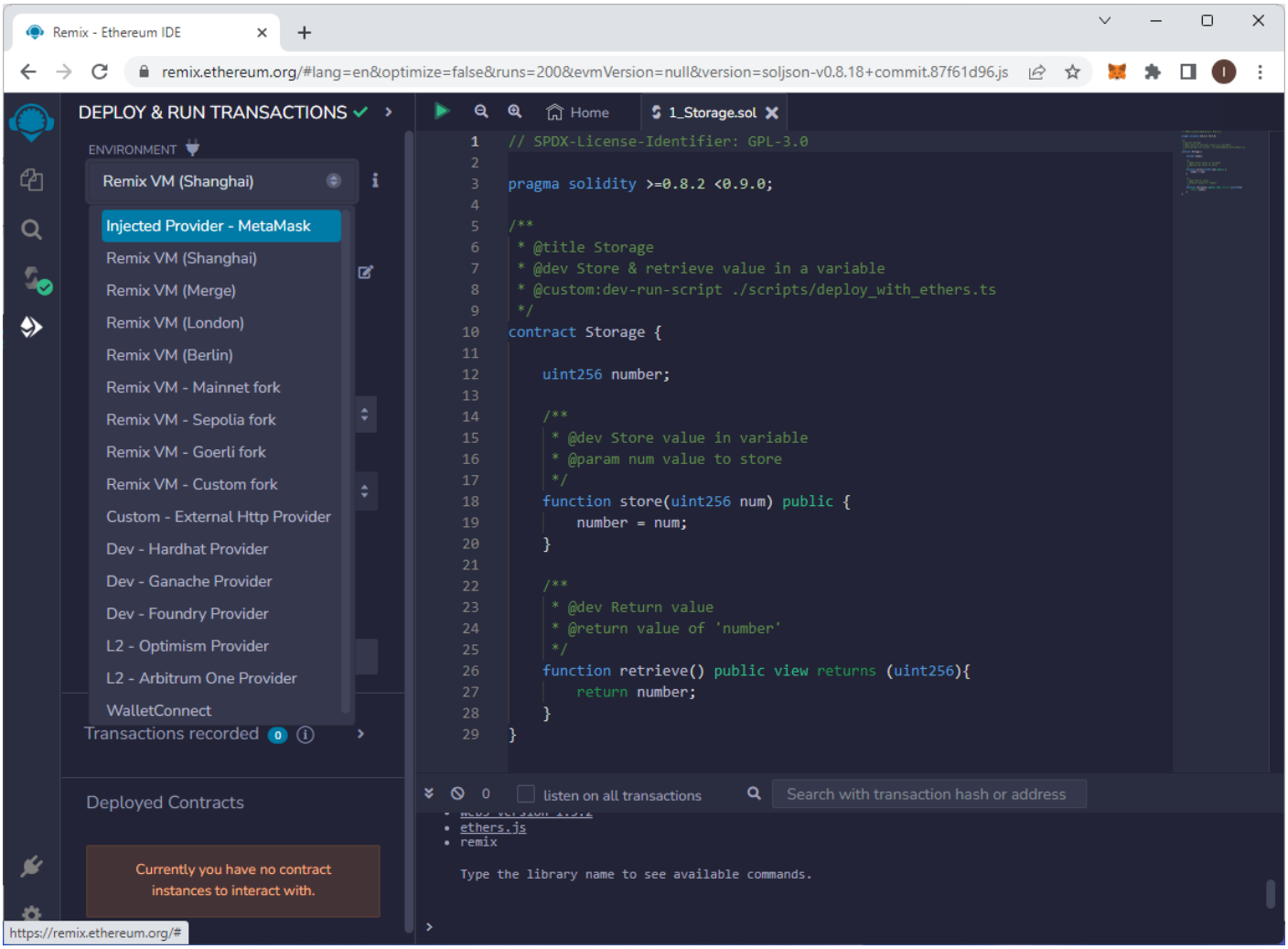
4. When the source code is error-free and the compilation is successful, you will see a green success icon.



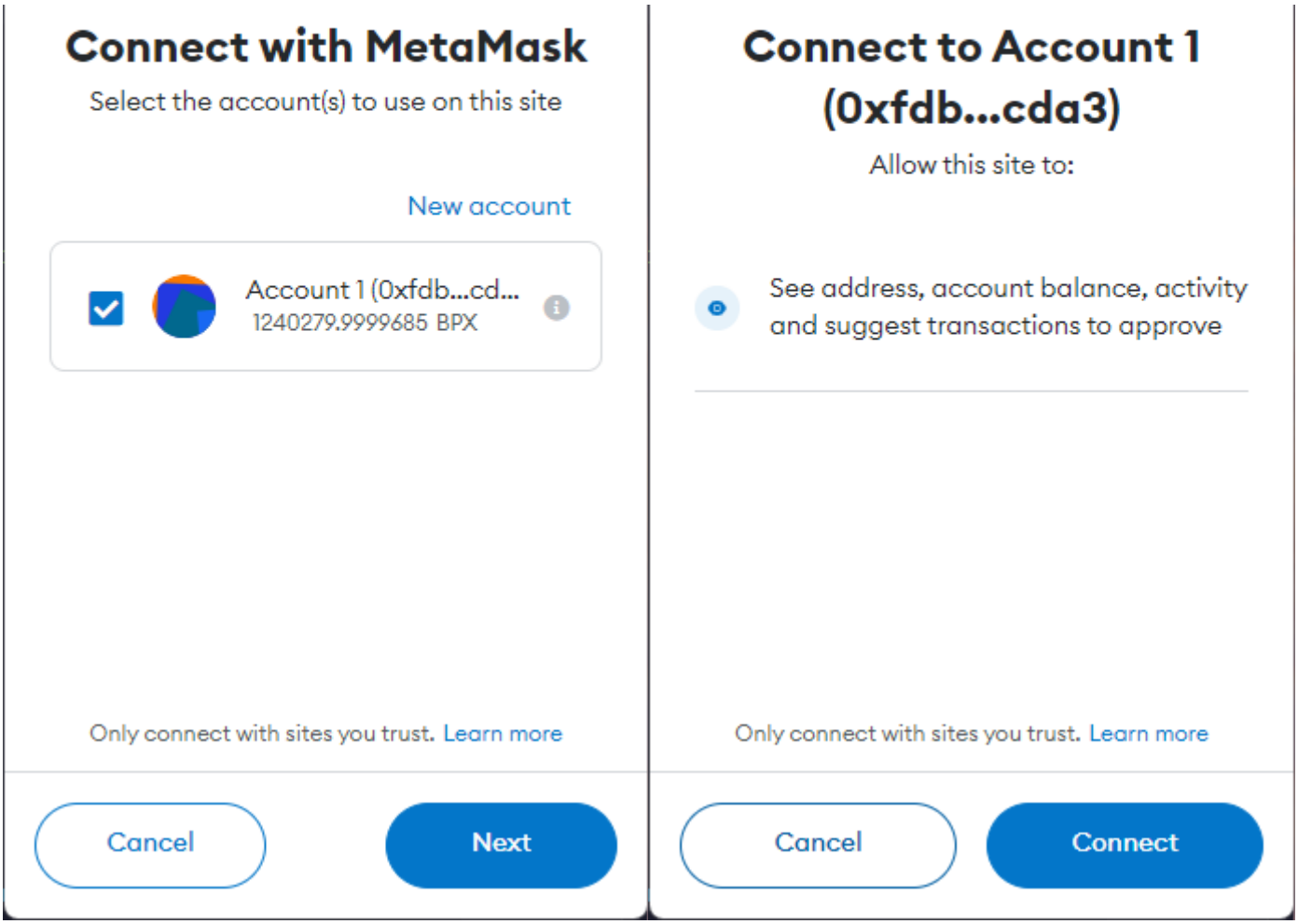
5. Go to the **Deploy** tab.



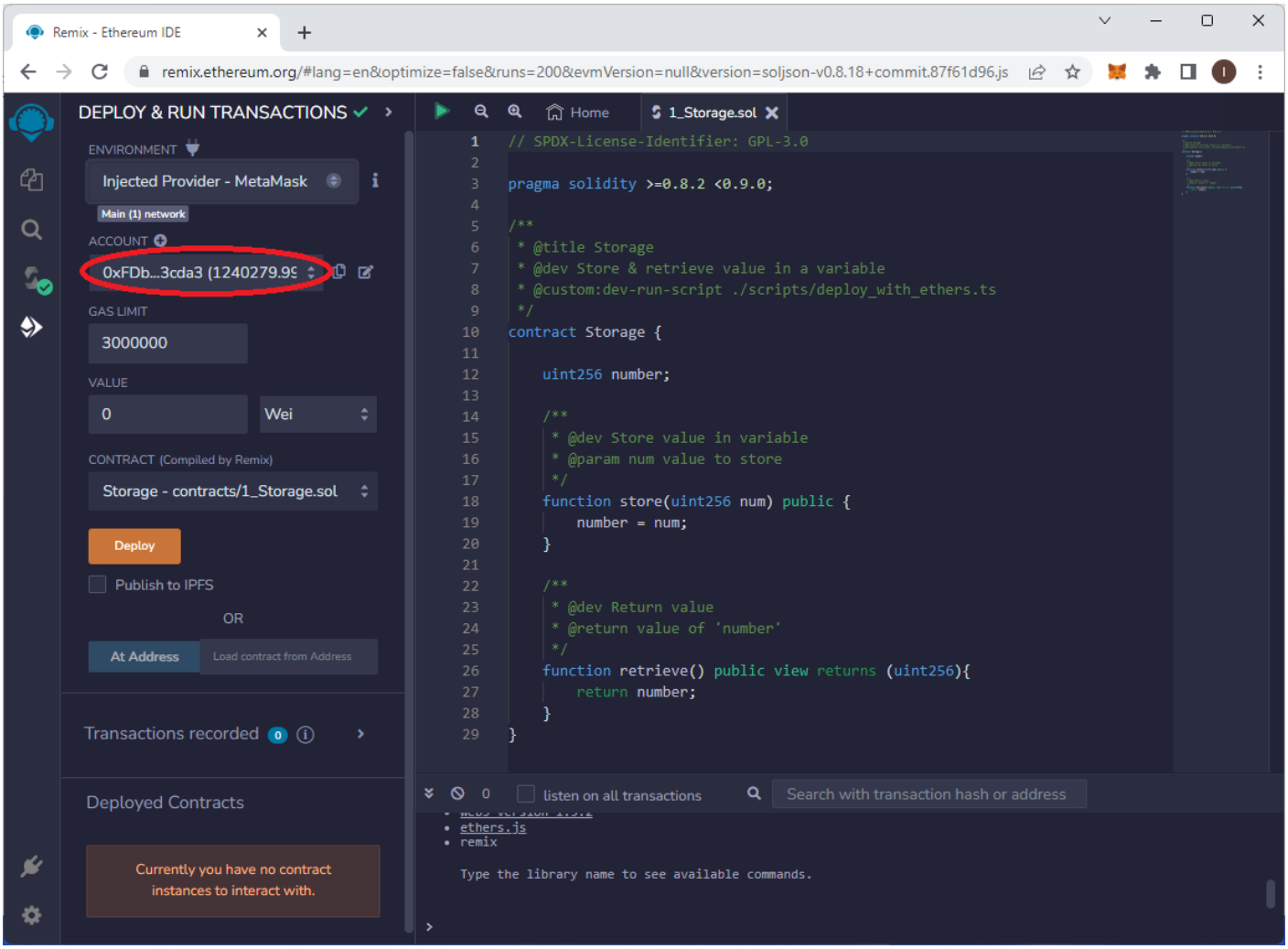
6. Expand the **Environment** list and select **Injected Provider - MetaMask**.



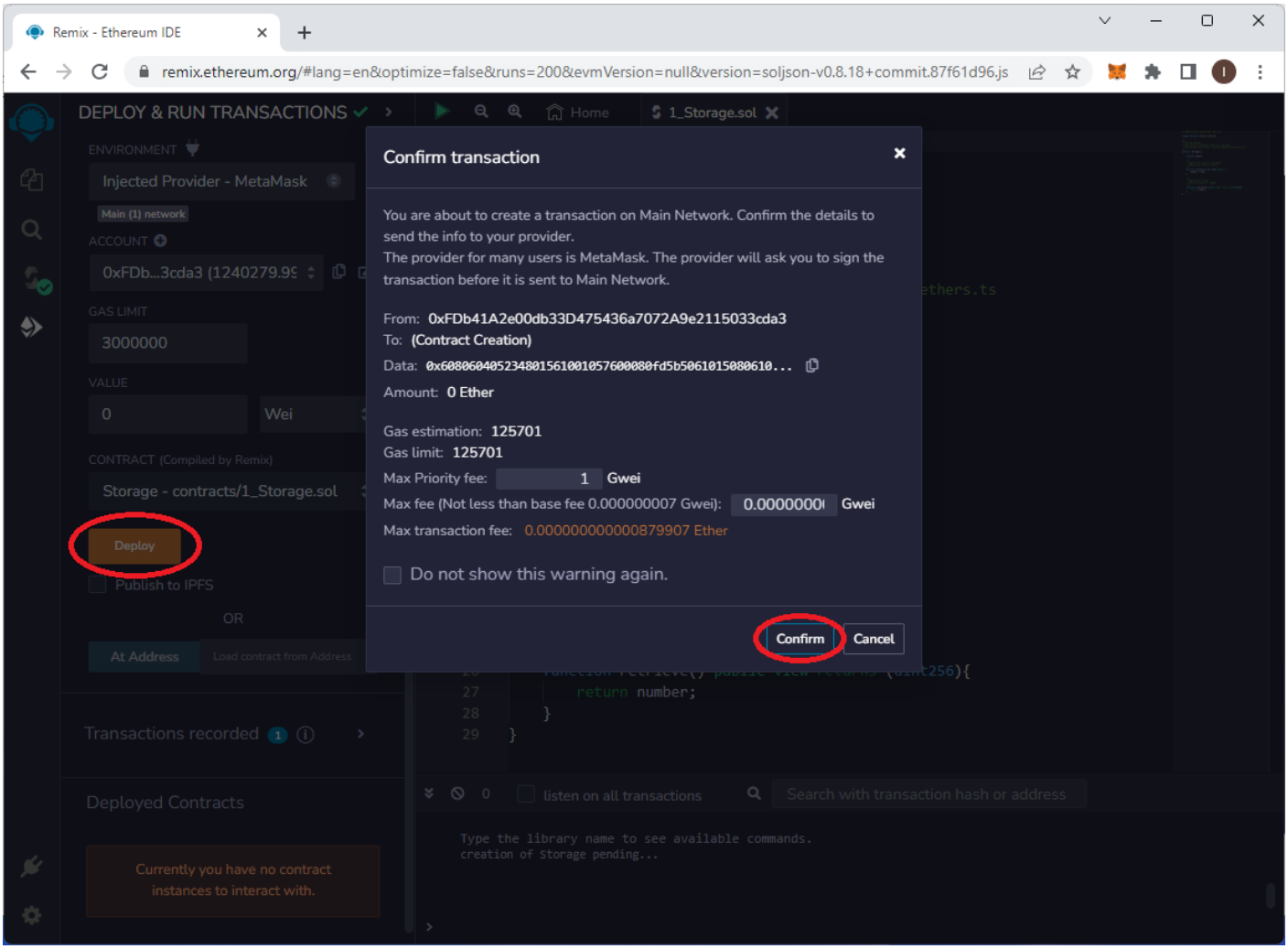
7. Now MetaMask will prompt you for permission to connect to Remix IDE. Agree to the connection.

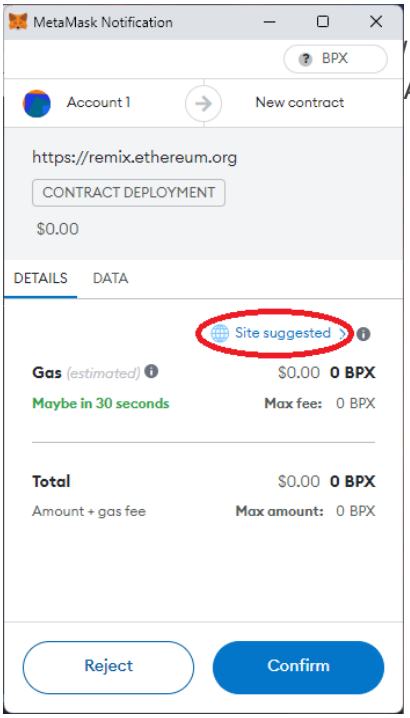


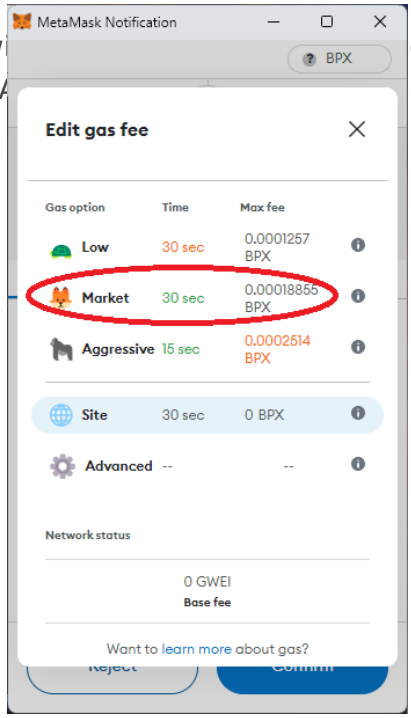
8. After successfully connecting, you should see your account address and its balance in the highlighted field.

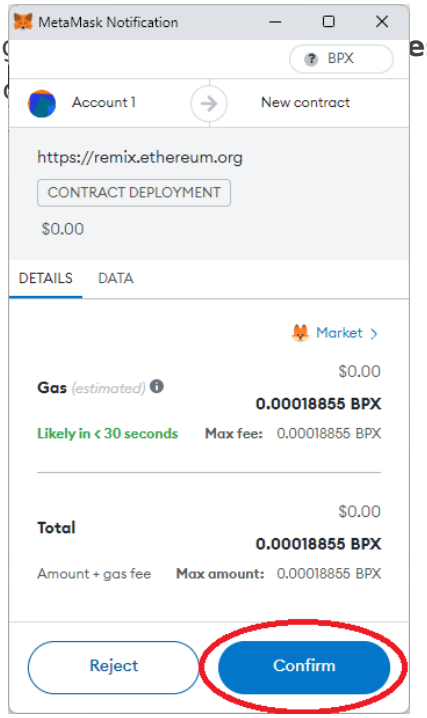


9. Click "**Deploy**" to begin deploying your contract. Confirm the suggested gas settings.


Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Provider - MetaMask
Main (1) network
ACCOUNT
0xFDb...3cda3 (1240279.99
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT (Compiled by Remix)
Storage - contracts/1_Storage.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded
Deployed Contracts
Currently you have no contract instances to interact with.
Home
1_Storage.sol
Confirm transaction
You are about to create a transaction on Main Network. Confirm the details to send the info to your provider.
The provider for many users is MetaMask. The provider will ask you to sign the transaction before it is sent to Main Network.
From: 0xFDb41A2e00db33D475436a7072A9e2115033cda3
To: (Contract Creation)
Data: 0x6080604052348015610010576000080fd5b5061015080610...
Amount: 0 Ether
Gas estimation: 125701
Gas limit: 125701
Max Priority fee: 1 Gwei
Max fee (Not less than base fee 0.000000007 Gwei): 0.0000000 Gwei
Max transaction fee: 0.000000000000879907 Ether
Do not show this warning again.
Confirm
Cancel
return number;
listen on all transactions
Search with transaction hash or address
Type the library name to see available commands.
creation of Storage pending...

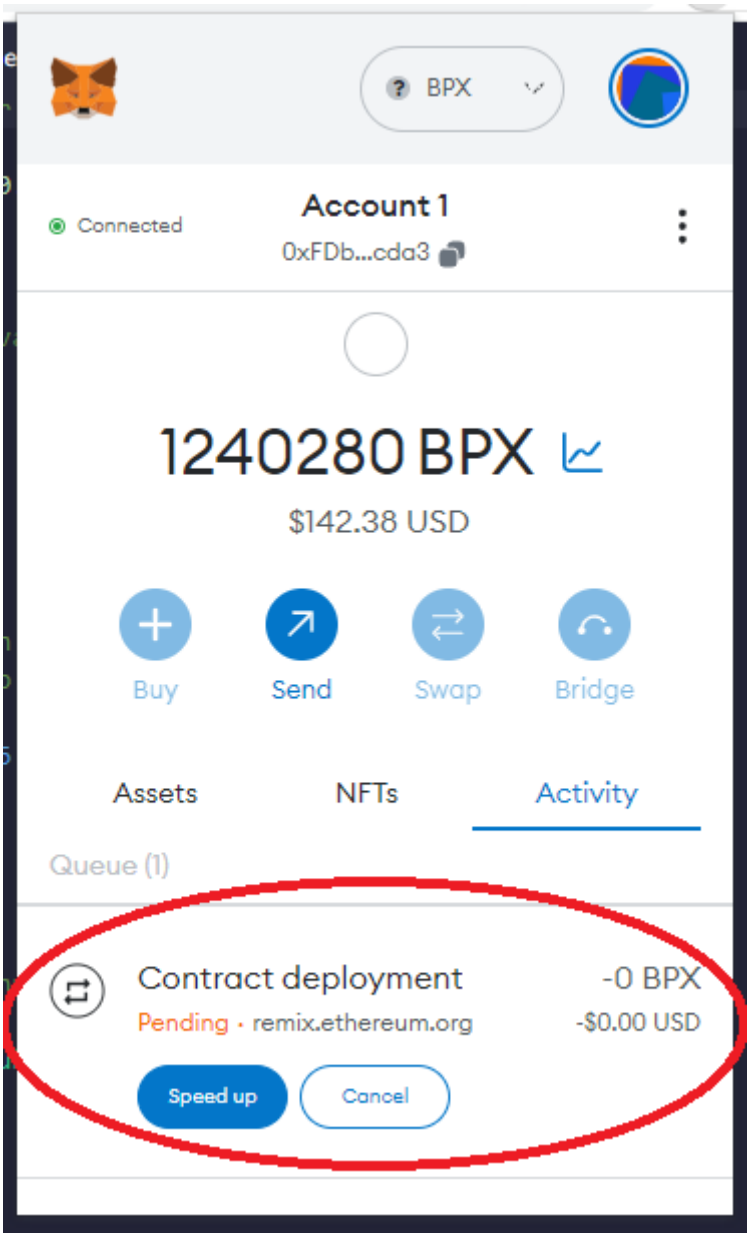

MetaMask Notification
BPX
Account 1
New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT
$0.00
DETAILS
DATA
Site suggested
Gas (estimated)
$0.00 0 BPX
Maybe in 30 seconds
Max fee: 0 BPX
Total
$0.00 0 BPX
Amount + gas fee
Max amount: 0 BPX
Reject
Confirm


MetaMask Notification
BPX
Edit gas fee
Gas option
Time
Max fee
Low
30 sec
0.0001257 BPX
Market
30 sec
0.00018855 BPX
Aggressive
15 sec
0.0002514 BPX
Site
30 sec
0 BPX
Advanced
Network status
0 GWEI
Base fee
Want to learn more about gas?

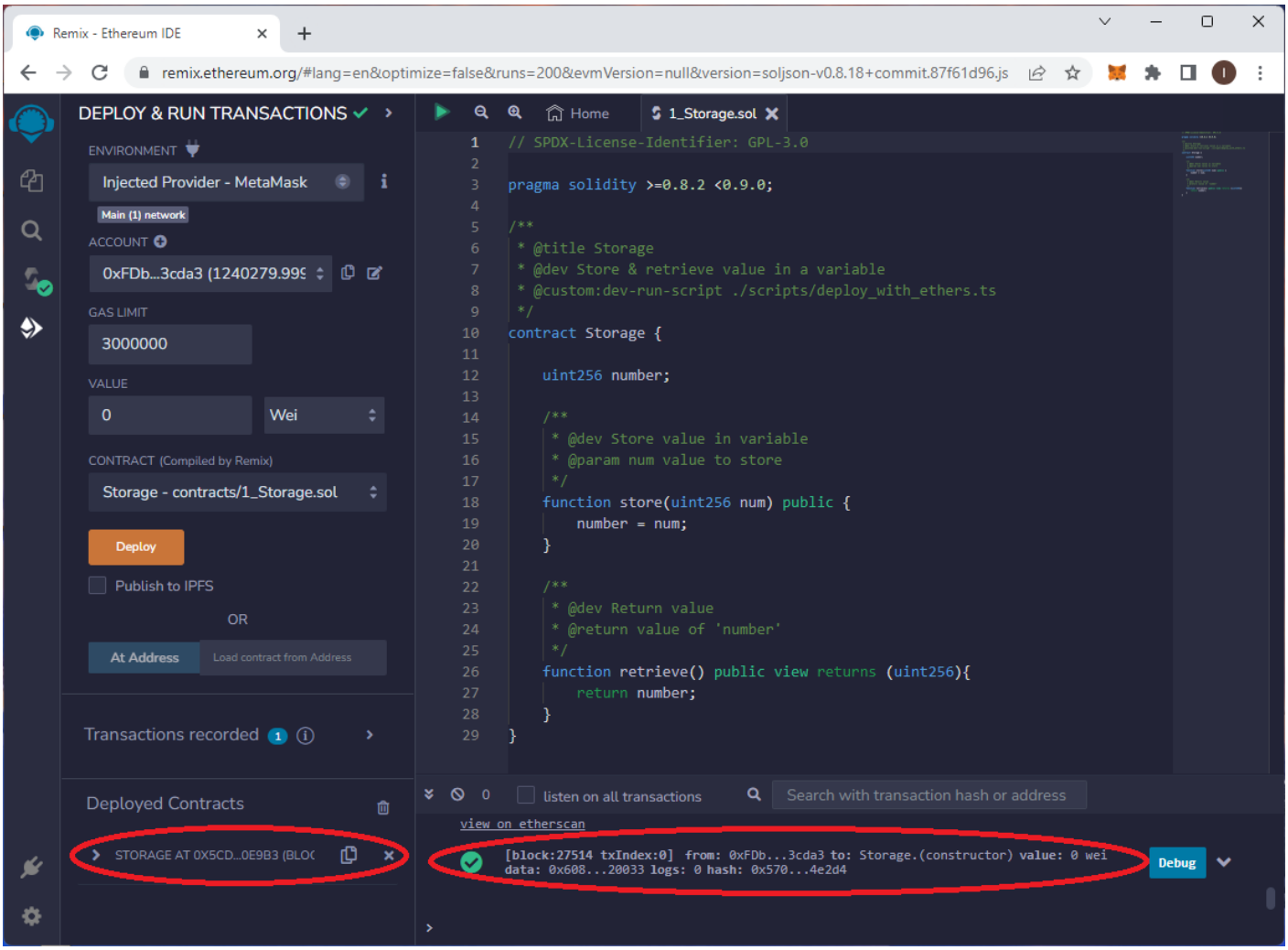

MetaMask Notification
BPX
Account 1
New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT
$0.00
DETAILS
DATA
Market
Gas (estimated)
$0.00
0.00018855 BPX
Likely in < 30 seconds
Max fee: 0.00018855 BPX
Total
$0.00
0.00018855 BPX
Amount + gas fee
Max amount: 0.00018855 BPX
Reject
Confirm

ested," then

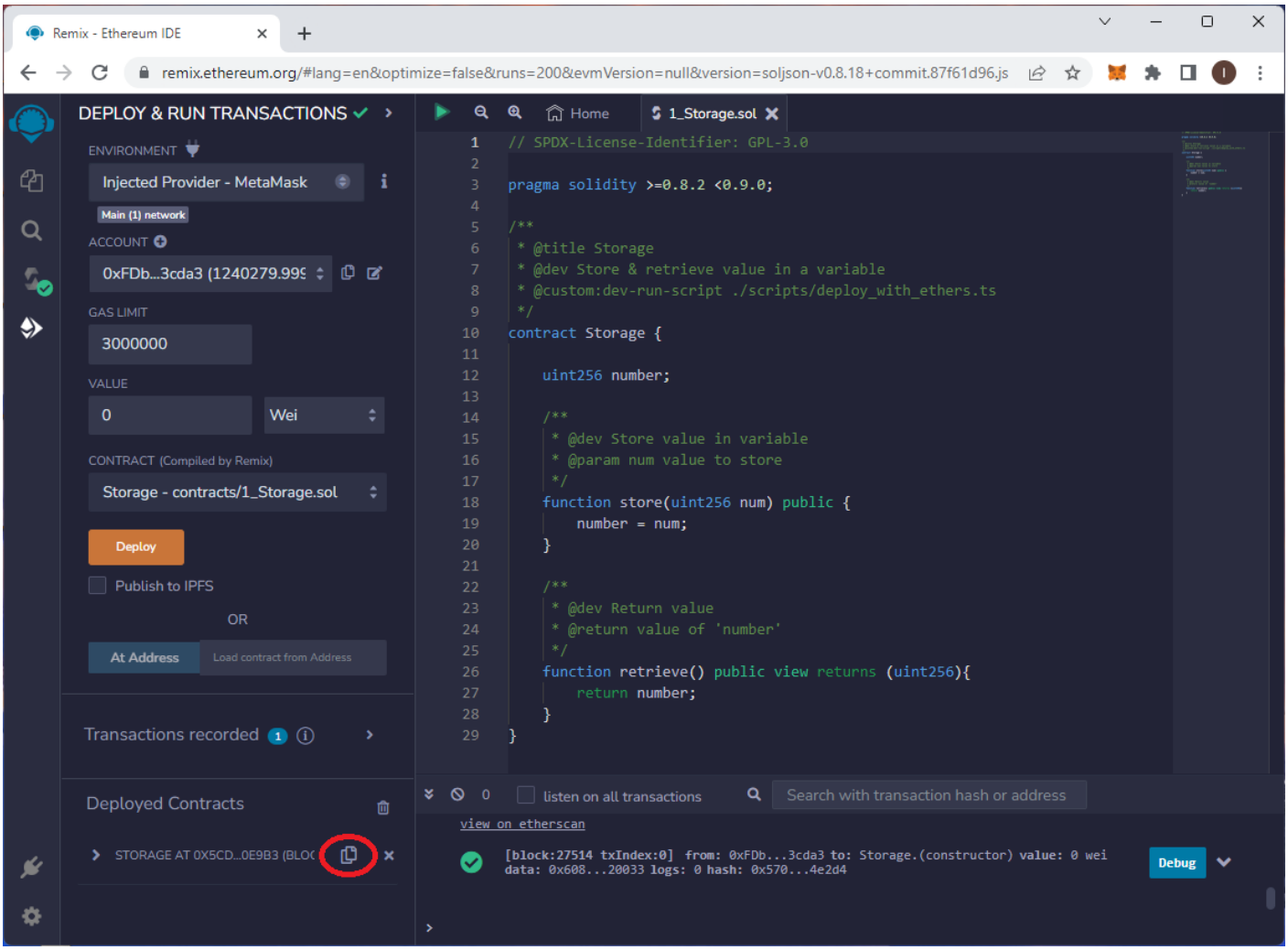
11. Your contract is now deploying. If you open the MetaMask window, you should see a new pending transaction.



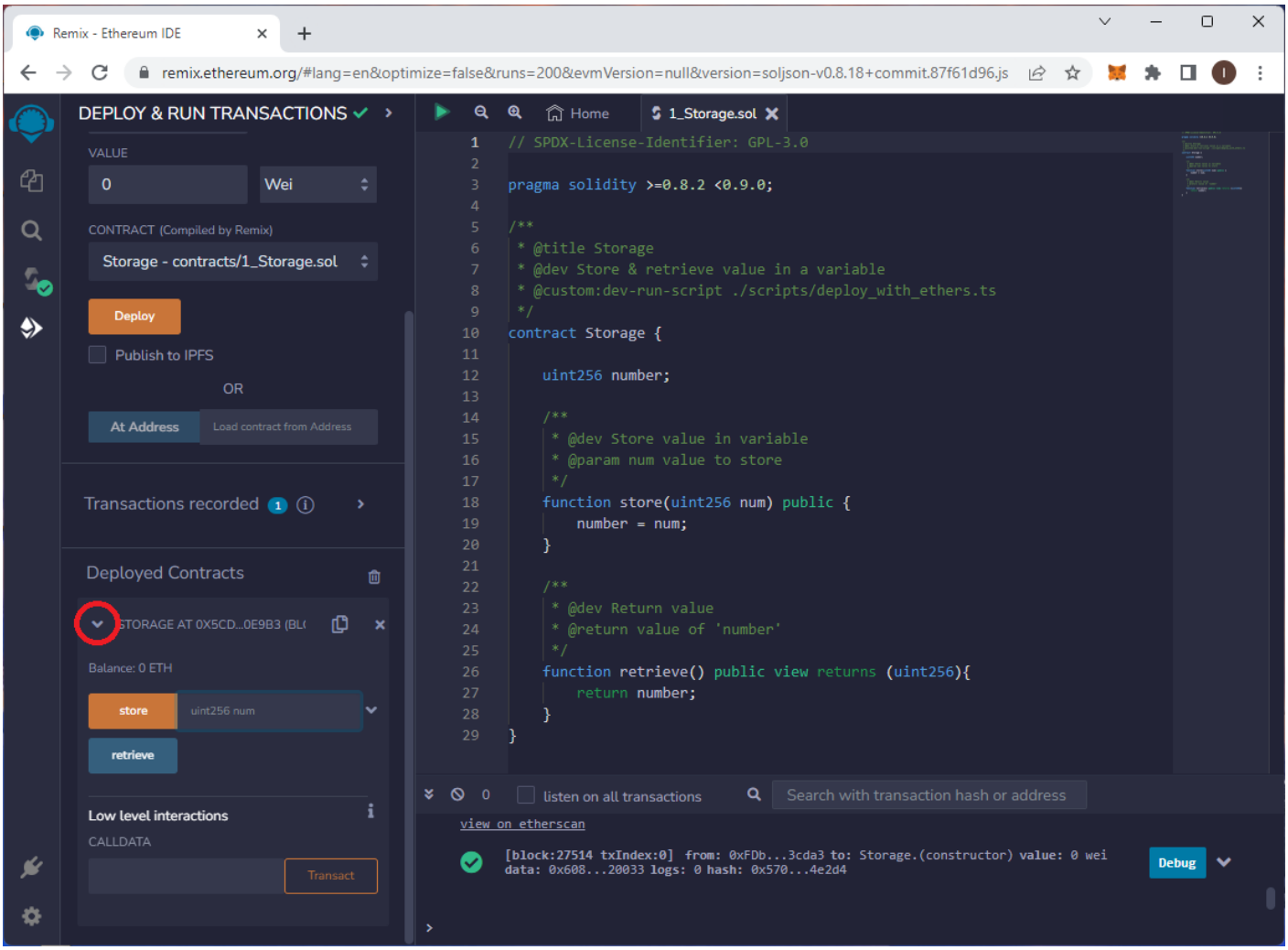
12. Once the transaction is confirmed by the blockchain, you will see its confirmation and a new entry in the Deployed Contracts section. Your contract has been successfully deployed.



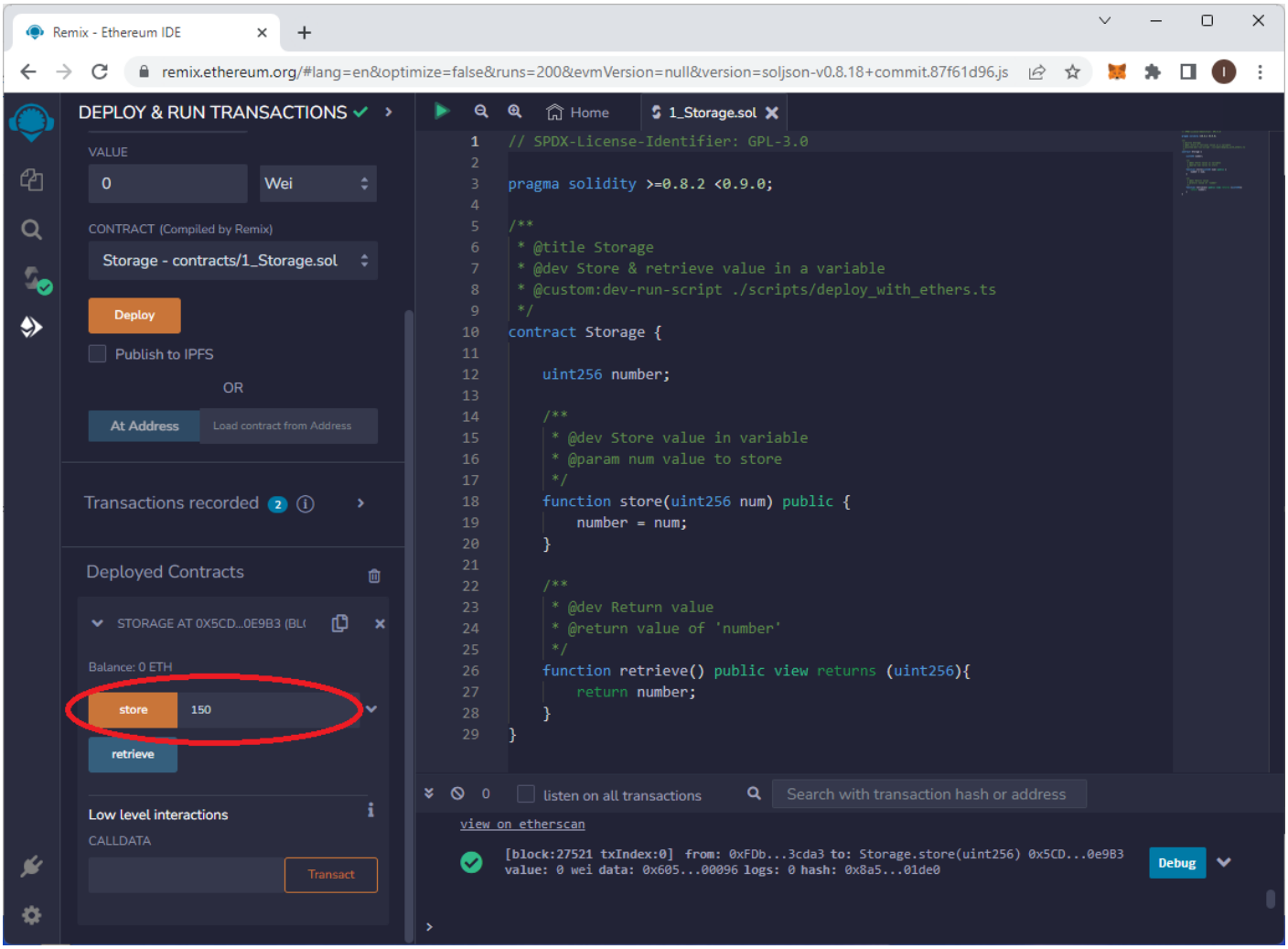
13. You can use the highlighted button to copy the address of your new contract.



14. You can interact with your contract directly from the IDE. Expand the list of contract methods to access them.

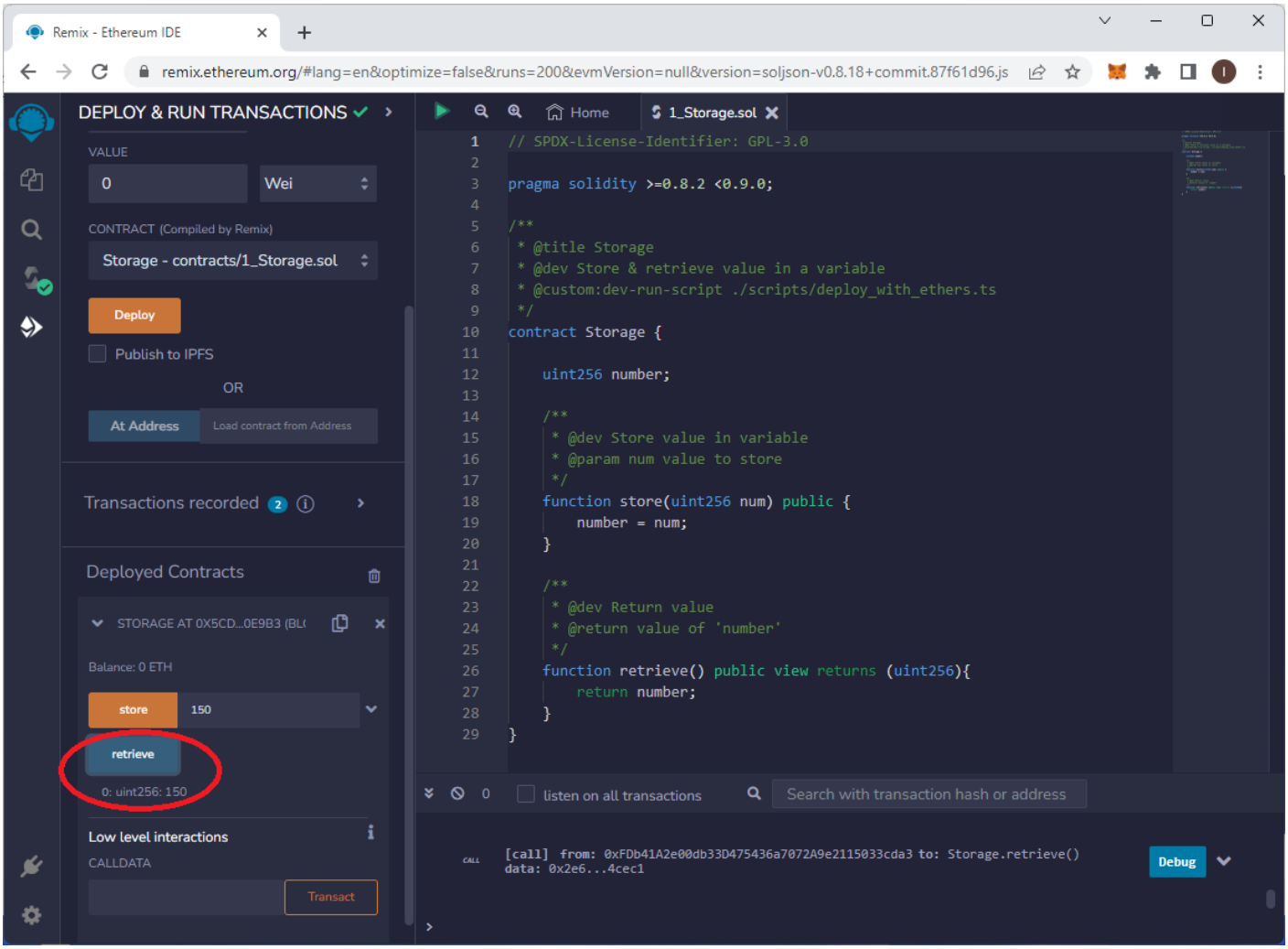


15. Let's call the first function (`store`) to save a number in our sample contract. Enter a random number in the text field next to the "**store**" button, and then press the "**store**" button.



16. Confirm the transaction in your wallet as you did when deploying the contract. Saving data to a smart contract requires a transaction on the blockchain.

17. When the transaction is confirmed, call the second method (`retrieve`) to read the number stored in the smart contract. Click the "**retrieve**" button, and the blockchain will return the value stored in the contract.


Remix - Ethereum IDE
remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96.js
DEPLOY & RUN TRANSACTIONS
VALUE
0
Wei
CONTRACT (Compiled by Remix)
Storage - contracts/1_Storage.sol
Deploy
Publish to IPFS
OR
At Address
Load contract from Address
Transactions recorded 2
Deployed Contracts
STORAGE AT 0X5CD...0E9B3 (BL
Balance: 0 ETH
store
150
retrieve
0: uint256: 150
Low level interactions
CALLDATA
Transact
Home
1_Storage.sol
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.8.2 <0.9.0;
/**
* @title Storage
* @dev Store & retrieve value in a variable
* @custom:dev-run-script ./scripts/deploy_with_ethers.ts
*/
contract Storage {
uint256 number;
/**
* @dev Store value in variable
* @param num value to store
*/
function store(uint256 num) public {
number = num;
}
/**
* @dev Return value
* @return value of 'number'
*/
function retrieve() public view returns (uint256){
return number;
}
}
listen on all transactions
Search with transaction hash or address
[call] from: 0xFDb41A2e00db33D475436a7072A9e2115033cda3 to: Storage.retrieve()
data: 0x2e6...4cec1
Debug