

How To Find ABI Of Smart Contract

The Application Binary Interface (ABI) in Solidity is a crucial standard that defines how to interact with smart contracts on the Ethereum blockchain, specifying the methods and data types used for communication.

The Application Binary Interface (ABI) of a smart contract gives a contract the ability to communicate and interact with external applications and other smart contracts. Receiving data from external sources can be critical for completing the goals of the application and the user.

The screenshot shows the REMIX IDE version 1.4.1. On the left, there's a sidebar with icons for Deploy & Run Transactions, CONTRACT, Transactions recorded, Deployed Contracts, CHANGETWEETLENGTH, and a Scam Alert button. The main area displays the Solidity code for the Twitter contract:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract Twitter {
    uint16 public MAX_TWEET_LENGTH = 280; // I change change 300, after deployment using
    struct Tweet {
        address author;
        string content;
        uint256 timestamp;
        uint256 likes;
    }
    mapping(address => Tweet[]) public tweets;
    address public owner;

    constructor() {
        owner = msg.sender;
    }

    modifier onlyOwner() {
        require(msg.sender == owner, "YOU ARE NOT THE OWNER!");
    }
}
```

Below the code, there's an Explain contract button and a transaction history section with a call to Twitter.getAllTweets. At the bottom, there's a Did you know? note about the Recorder feature.

After deploy and verify.
Copy contract address from remix ide or etherscan.

↓
Open repo.sourcify.dev and select your testnet.
↓

https://repo.sourcify.dev

sourcify.eth

Sourcify Verified Contract Repository

Chain

Ethereum Testnet Sepolia (11155111)

Address

0xB4170b0C99D0c96D219FD21224c65De24Cbd7300

View Contract

Example Contracts

NAME	ADDRESS
ERC1967Proxy	0x78f7C79d8aE156A6C68c67d0393d1Cc97df3Bd1
Uniswap UniversalR...	0x3fc91A3af7d70395C496C647d5a6C97d4B2h7FA

Most Used Contracts
by growthepie

Select Chain

Ethereum Mainnet (1)

Verified contracts can be viewed in Sourcify

Paste your contract address here.
And click on view contract



sourcify.eth

0xB4170b0C99D0c96D219FD21224c65De24Cbd7300

on Ethereum Testnet Sepolia (11155111)

Exact Match Runtime Bytecode Creation Bytecode

Contract Name	Twitter
Compilation Target	twitter.sol:Twitter
Language	Solidity
Compiler	solc 0.8.0+commit.c7dfd78e
EVM Version	default
Verified At	2025-12-22 05:55:35 UTC
Deployer	0x3Ca52f62e32751791eD5637A08d1063Cb5852c80
Deployment Transaction	0xb15044147891680034459d670ec9622e6c8ba1a647ae9f5b241d65cfa59262ac
Block Number	9890998
Transaction Index	0

ReadWrite Contract on: [OpenZeppelin](#)

ABI

Expand to view

Click on down arrow

↓

Read/Write Contract on: [OpenZeppelin](#)

ABI

Formatted [JSON](#)

`constructor`

`function MAX_TWEET_LENGTH`

`function changeTweetLength`

`function createTweet`

`function getAllTweets`

`function getTweet`

`function owner`

`function tweets`

Click On JSON



Copy the all ABI

Formatted [JSON](#)

```
1 [  
2 {  
3   "type": "constructor",  
4   "inputs": [],  
5   "stateMutability": "nonpayable"  
6 },  
7 {  
8   "name": "MAX_TWEET_LENGTH",  
9   "type": "function",  
10  "inputs": [],  
11  "outputs": [  
12    {  
13      "name": "",  
14      "type": "uint16",  
15      "internalType": "uint16"  
16    }  
17  ],  
18  "stateMutability": "view"  
19 },  
20 {  
21   "name": "changeTweetLength",  
22   "type": "function",  
23   "inputs": [  
24     {  
25       "name": "newTweetLength",  
26       "type": "uint16",  
27       "internalType": "uint16"  
28     }  
29   ],  
30   "outputs": [],  
31   "stateMutability": "nonpayable"
```