

Ethereum

Block Explorer

<https://etherscan.io/blocks>

Etherscan
The Ethereum Block Explorer

HOME BLOCKCHAIN TOKENS RESOURCES MORE

Blocks

Showing Block (#5905396 to #5905372) out of 5905022 total blocks

Height	Age	tx	Uncles	Miner	GasUsed	GasLimit	Avg GasPrice	Reward
5905396	37 mins ago	54	0	DcarPool1	8005515 (+1,17%)	8007798	59.00 Gwei	3.47231 Ether
5905395	37 mins ago	129	1	fpool_2	7987511 (+1,104%)	8003914	51.32 Gwei	3.50367 Ether
5905394	38 mins ago	147	0	miningpoolhub_1	7984705 (+9,311%)	8000029	46.05 Gwei	3.36706 Ether
5905393	38 mins ago	74	0	Nanopool	7905228 (+7,377%)	8000029	56.37 Gwei	3.3949 Ether
5905392	38 mins ago	38	0	Nanopool	7991625 (+9,394%)	8000029	52.97 Gwei	3.42329 Ether
5905391	38 mins ago	50	0	Ethereum	7988322 (+9,354%)	8000029	56.86 Gwei	3.4542 Ether
5905390	39 mins ago	36	0	Ethereum	7983270 (+9,754%)	8000029	50.00 Gwei	3.3992 Ether
5905389	39 mins ago	69	0	fpool_2	8001966 (+9,349%)	8003926	50.47 Gwei	3.40396 Ether
5905388	39 mins ago	26	0	Nanopool	7992945 (+9,514%)	8000029	50.81 Gwei	3.40611 Ether
5905387	39 mins ago	344	0	fpool_2	7993878 (+9,524%)	8000256	86.19 Gwei	3.68896 Ether
5905386	39 mins ago	10	0	Ethereum	5512563 (+6,311%)	7999992	44.85 Gwei	3.24720 Ether
5905385	40 mins ago	372	1	miningpoolhub_1	7954399 (+9,554%)	7992222	92.38 Gwei	3.82858 Ether
5905384	40 mins ago	85	1	DcarPool1	7978337 (+9,524%)	7984452	39.93 Gwei	3.41231 Ether

Blockübersicht

Eine Zeile beschreibt einen Block.
Ein Klick auf die Blocknummer ganz links öffnet die Blockdetails.



Etherscan
The Ethereum Block Explorer

HOME BLOCKCHAIN TOKENS RESOURCES MORE

Block #5905394

Home | Blocks | Block Information

Overview Comments

Block Information

Height:	Prev 5905394 Next
TimeStamp:	22 mins ago (Jul-04-2018 05:43:27 PM +UTC)
Transactions:	147 transactions and 7 contract internal transactions in this block
Hash:	0x1395244b7c2e72b4d5e4109842a247d3275375ec8f6d1a402a0c3b0b
Parent Hash:	0x0db157c2b781c23303995983cc3a68a493a445371bca5e6b1385393
Sha3Uncles:	0x1dcd4de8b2c7507a8b85e67b6cc041ad312451b548a74130a142540a489347
Mined By:	0x2930b35844c230905e1431caa96b643a0347 (miningpoolhub_1) in 9 secs
Difficulty:	3,486,184,042,078,879
Total Difficulty:	5,156,719,510,695,948,171,885
Size:	27962 bytes
Gas Used:	7,984,705 (99.81%)
Gas Limit:	8,000,029
Nonce:	0x029327709a176d85

Blockdetails

Etherscan
The Ethereum Block Explorer

HOME BLOCKCHAIN TOKENS RESOURCES MORE

Transactions for Block 5905394

Home | Transactions

A total of 147 transactions found

TxHash	Block	Age	From	To	Value	(Info)
0x1618e11e3bac10...	5905394	22 mins ago	0xc1bba86003585a...	0x4b317864a05c9f...	0 Ether	0.00194911
0xa1a5a024a4e0b...	5905394	22 mins ago	0x74d56c9ed71e9...	BinanceWallet	0.19977703288882 Ether	0.000756
0x221a2c78a9b7...	5905394	22 mins ago	0x0072e45944a0b...	0x4b317864a05c9f...	0 Ether	0.00194911
0xb6e13a0c03e1...	5905394	22 mins ago	0xfcc127251d95af...	0x59ebb83b72d735...	0 Ether	0.00194336
0x564d419a3c03...	5905394	22 mins ago	0xc0e73d7e133eaf...	0x4b317864a05c9f...	0 Ether	0.00194911
0x430349177ba7a...	5905394	22 mins ago	0x42805ea033ba6...	0xa6278e6879777...	0.09 Ether	0.000891
0x911a03a411674...	5905394	22 mins ago	0x50445d0d887d0c...	0xc0d624a8b63b0...	0.000059229 Ether	0.000891
0x2a00b3a5155d8...	5905394	22 mins ago	0x04a1e7c2296ac...	0x194457773a60b...	0.004 Ether	0.000891
0xa7070e7339a39f...	5905394	22 mins ago	0x2ef80d577664bf...	0x09b60ab045a2f...	0 Ether	0.0007561
0x1f38482205...	5905394	22 mins ago	0x13ab33973d37c...	TronToken	0 Ether	0.00118038
0xc1077b08942...	5905394	22 mins ago	0x26423364a2a6...	0x350a78d09e99d5...	1.098509 Ether	0.000891
0xa119d17a719...	5905394	22 mins ago	0x1272750a5a3f...	0x350a78d09e99d5...	0.27402705 Ether	0.000891
0xb0a1e104a4ac1c...	5905394	22 mins ago	0xd312323ac3a91...	0x0a0a0a471a3321...	0 Ether	0.00177681
0xc0303116073e14...	5905394	22 mins ago	0x35da78d09e99d5...	0xa693a05909a67...	0.01 Ether	0.000891

<https://etherscan.io/tx/0x1f38482205c40356ad0890639750e6d2d308c7eb098ba02c2c31a6a73c>

Transaktionen im Block

The screenshot shows the Etherscan interface for a transaction. At the top, there's a search bar and navigation links. The transaction ID is 0x7cf3bf48259c5a036deb0890639750e8c42d3080c9e9c98bd02c2c3106e73c. The transaction is successful, with a block height of 5905394 and a timestamp of 33 minutes ago. The 'From' field shows the address 0x513ab33973c3704e80073c7726944809640119c, which is linked to TronToken. The 'To' field shows the contract address 0x0230b790e053906d295944390032c0f7b704941735026 (\$395.01). The value transferred is 0 Ether (\$0.00). The gas limit is 400000, and the gas used is 52155. The gas price is 0.000000041 Ether (\$1.02), and the actual transaction fee is 0.002138355 Ether (\$1.02). The input data shows a function call: `function: transfer(address _to, uint256 _value)` with arguments `0x0000000000000000000000000000000000000000000000000000000000000000` and `0x0000000000000000000000000000000000000000000000000000000000000000`.

Transaktionsdetails

Transaktionen können auch einfach Ether zwischen Konten transferieren. Das Beispiel zeigt einen Transaktionsaufruf eines Smart Contracts, hier eine Token-Transfer in einem ERC20-Token-Contract namens „TronToken“.

Token: Ein Token ist eine Cryptowährung, die innerhalb einer anderen Blockchain als Smart Contract implementiert ist. Gegenteil: *Coin:* Diese besitzt eine eigene Blockchain. Ein Token läuft auf der Blockchain-Technologie einer anderen Coin. Warnung: Tokens können sehr einfach erstellt werden. Viele Tokens sind wertlos.

The screenshot shows the Etherscan Contract Overview page for TronToken. The contract address is 0x0230b790e053906d295944390032c0f7b704941735026 (\$395.01). The contract creator is 0x513ab33973c3704e80073c7726944809640119c. The token balances are shown as 1927864 tris. The contract source code is verified (Exact match). The contract name is TronToken, and the compiler version is v0.4.16+commit.d78b1dd9. The source code is displayed in a code editor, showing the contract definition for TronToken, including variables for token name, symbol, decimals, and digit, and functions for mapping addresses to balances and allowances.

Smart Contract: Code

Solidity

Die Programmiersprache für die Smart Contracts in Ethereum.

Online-Editor, -Compiler, -Debugger und -Simulator für Solidity: Remix: <https://remix.ethereum.org>

Dokumentation: <http://solidity.readthedocs.io/>

Oberfläche:

Editor

Compiler und Debugger

The screenshot displays the Remix IDE interface. On the left, the 'browser' and 'config' panels are visible. The main 'Editor' panel shows a Solidity contract named 'KmZaehler.sol' with the following code:

```
1 pragma solidity ^0.4.24;
2
3 /*
4  Einfacher KM-Zähler für ein neues Auto, welches einen Smart Contract verwendet
5  */
6
7 contract KmZaehler {
8     address creator;
9     int aktuelleKmStandInBlockchain;
10    string fahrgestellnummerInBlockchain;
11
12    // Konstruktor: Initialisiert die Daten im Smart Contract
13    // Jedes Auto hat seinen eigenen Smart Contract für den Kilometerstand
14    constructor (string fahrgestellnummer) public {
15        fahrgestellnummerInBlockchain = fahrgestellnummer;
16        aktuelleKmStandInBlockchain = 0;
17        creator = msg.sender;
18    }
19
20    // Hochladen des aktuellen Kilometerstandes
21    function hochladenKM (int aktuelleKM) public {
22        // nur eigenes Auto darf eigene km hochladen
23        require (msg.sender == creator, "Du bist nicht das Auto dieses Contracts");
24
25        // neue km müssen höher sein als davor
26        require (aktuelleKM > aktuelleKmStandInBlockchain, "Das kann nicht der richtige KM-Stand sein");
27
28        aktuelleKmStandInBlockchain = aktuelleKM;
29    }
30
31    // Lesen des Kilometerstandes aus der Blockchain
32    // read-only und gibt int zurück: constant returns(int)
33    function lesenAktuelleKMinBlockchain() public constant returns(int) {
34        return aktuelleKmStandInBlockchain;
35    }
36 }
```

On the right, the 'Compiler und Debugger' panel is shown. The 'Compile' tab is active, displaying the 'KmZaehler' contract. The 'Start to compile' button is highlighted with a blue arrow. Below the compiler panel, the 'Ausgabe des Simulators' (Simulator Output) panel is visible, showing a search bar and a list of transactions.

Ausgabe des Simulators

This screenshot shows the deployment process in the Remix IDE. The 'Compiler und Debugger' panel is active, and the 'Deploy' button is highlighted with a blue arrow. The 'Environment' section shows the account '0xca3...a73c (100 ether)' and the gas limit '3000000'. The 'Value' is set to '0'. The 'Contract' dropdown is set to 'KmZaehler', and the 'MainAuto' constructor parameter is entered. The 'Deploy' button is highlighted with a blue arrow. The 'Load contract from Address' button is also visible. The '0 pending transactions' and '0 contract instances' status is shown at the bottom.

Ethereum-Konto im Simulator, welches den Contract anlegt.

Contract anlegen durch Eingabe des Konstruktor-Parameters (in Anführungszeichen) und „Deploy“ ausführen.

Weitere Beispiele:

Simple Auction:

<http://solidity.readthedocs.io/en/v0.4.24/solidity-by-example.html#simple-open-auction>

The screenshot shows the Solidity documentation interface. On the left is a sidebar with a search bar and a navigation menu. The main content area is titled 'Simple Open Auction' and contains a paragraph explaining the auction's general idea. Below the text is a code block showing the Solidity contract for a simple auction.

Solidity
v0.4.24

Search docs

Introduction to Smart Contracts
Installing the Solidity Compiler

▢ Solidity by Example

- ▢ Voting
- ▢ Blind Auction
 - Simple Open Auction
 - Blind Auction
- Safe Remote Purchase
- Micropayment Channel

Solidity in Depth
Security Considerations
Using the compiler
Contract Metadata
Application Binary Interface Specification
Joyfully Universal Language for (Inline) Assembly
Style Guide
Common Patterns

Read the Docs v: v0.4.24 ▼

Simple Open Auction

The general idea of the following simple auction contract is that everyone can send their bids during a bidding period. The bids already include sending money / ether in order to bind the bidders to their bid. If the highest bid is raised, the previously highest bidder gets her money back. After the end of the bidding period, the contract has to be called manually for the beneficiary to receive his money - contracts cannot activate themselves.

```
pragma solidity ^0.4.22;

contract SimpleAuction {
    // Parameters of the auction. Times are either
    // absolute unix timestamps (seconds since 1970-01-01)
    // or time periods in seconds.
    address public beneficiary;
    uint public auctionEnd;

    // Current state of the auction.
    address public highestBidder;
    uint public highestBid;

    // Allowed withdrawals of previous bids
    mapping(address => uint) pendingReturns;

    // Set to true at the end, disallows any change
    bool ended;

    // Events that will be fired on changes.
    event HighestBidIncreased(address bidder, uint amount);
    event AuctionEnded(address winner, uint amount);

    // The following is a so-called natspec comment,
    // recognizable by the three slashes.
    // It will be shown when the user is asked to
    // confirm a transaction.

    /// Create a simple auction with `_biddingTime`
    /// seconds bidding time on behalf of the
    /// beneficiary address `_beneficiary`.
    constructor(
        uint _biddingTime,
```

Die eigene Kryptowährung:

Einen sehr einfachen eigenen Token mit Solidity programmieren

<https://www.ethereum.org/token>

The screenshot shows the Ethereum documentation page for creating a token. It includes a header with the Ethereum logo and the text 'Create your own crypto-currency'. The main content is titled 'The Coin' and explains the purpose of the token. Below this is a section titled 'MINIMUM VIABLE TOKEN' which describes the standard token contract. A code block shows the Solidity contract for 'MyToken'.

ETHEREUM » Create your own crypto-currency NEXT

The Coin

We are going to create a digital token. Tokens in the Ethereum ecosystem can represent any fungible tradable good: coins, loyalty points, gold certificates, IOUs, in-game items, etc. Since all tokens implement some basic features in a standard way, this also means that your token will be instantly compatible with the Ethereum wallet and any other client or contract that uses the same standards.

MINIMUM VIABLE TOKEN

The standard token contract can be quite complex. But in essence a very basic token boils down to this:

```
pragma solidity ^0.4.20;

contract MyToken {
    /* This creates an array with all balances */
    mapping (address => uint256) public balanceOf;

    /* Initializes contract with initial supply tokens to the creator of the contract */
    function MyToken(
        uint256 initialSupply
    ) public {
        balanceOf[msg.sender] = initialSupply; // Give the creator all initial tokens
    }

    /* Send coins */
    function transfer(address _to, uint256 _value) public {
        require(balanceOf[msg.sender] >= _value); // Check if the sender has enough
        require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
        balanceOf[msg.sender] -= _value; // Subtract from the sender
        balanceOf[_to] += _value; // Add the same to the recipient
    }
}
```

OPEN CHAT