



School: Campus:

Academic Year: Subject Name: Subject Code:

Semester: Program: Branch: Specialization:

Date:

Applied and Action Learning

(Learning by Doing and Discovery)

Name of the Experiment : Know Your TX – Dissecting a Transaction

Objective/Aim:

To study and understand about TX and how to dissect a transaction.

Apparatus/Software Used:

- Laptop/PC
- Remix IDE
- Ethereum cloud
- Sepolia tesnet
- Internet for research

Theory/Concept:

1. A transaction (TX) in the context of cryptocurrency refers to the movement of assets or data between entities on a blockchain network. It represents the transfer of cryptocurrency from a sender to a receiver, and this exchange is recorded on the blockchain, ensuring transparency and security
2. To dissect a transaction, one can **use a block explorer, which allows users to look up the transaction details and confirmations using the transaction ID (TXID) or transaction hash (Tx Hash)**. The transaction hash is a unique identifier that contains information such as the sender's address, the receiver's address, the amount transferred, the time, and other relevant details.

DEPLOY & RUN TRANSACTIONS

Transactions recorded 6

Deployed Contracts 4

SIMPLESTORAGE AT 0XD88...3

Balance: 0 ETH

set 321

get

0: uint256: 321

storedData

Low level interactions

CALLDATA

Transact

SIMPLESTORAGE AT 0XF8E...9F

SIMPLESTORAGE AT 0X7EF...BC

SIMPLESTORAGE AT 0XDA0...4

1 // SPDX-License-Identifier: MIT

2 pragma solidity ^0.8.0;

3

4 contract SimpleStorage {

5 uint256 public storedData;

6

7 constructor(uint256 storedData) {

8 storedData = storedData; // lte gas 73800 gas

9 }

10

11 function set(uint256 x) {

12 storedData = x; // 22514 gas

13 }

14

15 function get() public view returns (uint256) {

16 return storedData; // 2453 gas

17 }

18 }

19

0

Listen on all transactions

Filter with transaction hash or address

[vm] from: 0x5B3...eddC4 to: SimpleStorage.(constructor) value: 0 wei data: 0x608...0b1d3 logs: 0 hash: 0x90f...2489f

Debug

Sepolia Testnet

Search by Address / Txn Hash / Block / Token

Etherscan

Home Blockchain Tokens NFTs More

Contract 0xDA0bab807633f07013f94DD0E6A4F96F8742B53

Overview

ETH BALANCE
7.302153661442030265 ETH

TOKEN HOLDINGS
\$0.00 (12 Tokens)

More Info

CONTRACT CREATOR
0x5B38Da6a...f56beddC4 | 2 yrs 248 days ago

Multichain Info

N/A

Advertise your brand here!

Transactions

Internal Transactions

Token Transfers (ERC-20)

Contract

Events

Latest 25 from a total of 928 transactions

Download Page Data

Transaction Hash	Method	Block	Age	From	To	Amount	Txn Fee
0xe311d83de5...	Mint	8808893	28 hrs ago	0x3A0B6050...4163E589C	0xDA0bab80...6F8742B53	0 ETH	0.00000003

Account 1

0xDAFe4...CCb81

0.05 SepoliaETH

+ \$0 (+0.00%) Portfolio

Buy/Sell Swap Bridge Send Receive

Solana is now supported

Tokens NFTs Activity

Sepolia

SepoliaETH No conversion rate available 0.05 SepoliaETH

Google Cloud Web3

PYUSD analytics now available in Looker Studio. View dashboard

Home Discover Faucet Web3 AI Events Learn Community Universal Ledger Startup Program

Ethereum Sepolia Faucet BETA

Get free Sepolia ETH sent directly to your wallet. Brought to you by Google Cloud for Web3.

Drip complete

Testnet tokens sent! Check your wallet address.

Network
Ethereum Sepolia

Recipient
0xDAFe4DfDDA3E39e0580D206BcF859b93cAECb81

Transaction hash
0x9dce76ec45a52e1c7144ce977154684115be03166d1bc2d95789881a3ef8e77

Procedure:

1. **Start the transaction:** Initiate the transaction to group multiple database operations into a single unit of work.
2. **Disable auto-commit:** Turn off the auto-commit mode to prevent each SQL statement from being automatically committed.
3. **Execute SQL statements:** Perform the necessary database operations such as updates, inserts, or deletes within the transaction.
4. **Commit the transaction:** If all operations are successful, commit the transaction to make the changes permanent in the database.
5. **Rollback the transaction:** If any operation fails, rollback the transaction to undo all changes and restore the database to its previous consistent state.
6. **Handle exceptions:** Implement error handling to catch any exceptions that occur during the transaction and decide whether to commit or rollback based on the error.
7. **Close the connection:** Ensure that the database connection is properly closed after the transaction is completed.

Observation Table:

1. Identify transactions and source documents: Transactions must be first identified and corroborated with source documents like receipts, invoices, or bank statements.
2. Analyse transactions using the accounting equation: Each transaction is examined to see how it affects the accounting equation: $\text{Assets} = \text{Liabilities} + \text{Owner's Equity}$.
3. Record journal entry: After the analysis, the transaction is recorded in the journal as a journal entry, capturing the debit and credit aspects as per double-entry bookkeeping.
4. Post entry to ledger: Finally, the journal entry is posted to the general ledger where it is categorized into the appropriate accounts

:

ASSESSMENT

Rubrics	Full Mark	Marks Obtained	Remarks
Concept	10		
Planning and Execution/ Practical Simulation/ Programming	10		
Result and Interpretation	10		
Record of Applied and Action Learning	10		
Viva	10		
Total	50		

Signature of the Faculty:**Signature of the Student:**
 Name :
 Regn.No.

