Ethereum Transaction Reference for Exchange Operations

# Overview

This document outlines the structure and required blockchain data for Ethereum (ETH and ERC-20) transactions used in a Java-based exchange system via web3j and Infura. It details how to simulate and track transactions for deposits, withdrawals, and internal transfers.

# 1. Ethereum Transaction Structure

Ethereum transactions contain the following fields:

- `from`: Sender's address

- `to`: Recipient address or contract

- `nonce`: Number of transactions sent from the sender

- `value`: Amount of ETH (in wei)

- `gasLimit`: Maximum gas to consume

- `gasPrice`: Price per gas unit (legacy)

- `maxPriorityFeePerGas` / `maxFeePerGas`: Used in EIP-1559

- `data`: Input data (e.g., contract method call)

- `hash`: Transaction hash (identifier)

- `blockNumber`: Block where transaction was included

## ETH Transfers

- `to`: Receiver's address

- `value`: Amount in wei

- `data`: Empty

## ERC-20 Transfers

- `to`: Token contract address

- `value`: 0

- `data`: Encoded `transfer(address to, uint256 amount)`

# 2. Exchange Operations & Required Fields

## A. Deposits

## # ETH Deposits (Real On-Chain)

- `from`: User address

- `to`: Exchange deposit address

- `value`: ETH amount

- `transactionHash`, `blockNumber`, `confirmations`

## # ERC-20 Deposits (Real On-Chain)

- `from`: User address

- `contractAddress`: Token contract

- `to`: Exchange address (within `Transfer` event)

- `value`: Token amount

- `transactionHash`, `blockNumber`

## B. Withdrawals

## # ETH Withdrawals (Real On-Chain)

- `from`: Exchange hot wallet

- `to`: User wallet

- `value`: ETH amount

- `nonce`, `gasLimit`, `gasPrice`

- `data`: Empty

- Signed and sent with web3j

## # ERC-20 Withdrawals (Real On-Chain)

- `from`: Exchange hot wallet

- `to`: Token contract

- `data`: Encoded `transfer(userAddress, amount)`

- `value`: 0

## C. Internal Transfers (Simulated Off-Chain)

- `fromUserId`, `toUserId`

- `assetType`, `amount`

- `timestamp`

# 3. Blockchain Data to Collect

For each real transaction (deposit/withdrawal):

- `transactionHash`

- `blockNumber`, `timestamp`

- `from`, `to`, `value`

- `contractAddress` (ERC-20)

- `gasUsed`, `gasPrice`

- `status`, `confirmations`

- Event logs (e.g., `Transfer` events`)

For internal transfers: purely application-level records.

# 4. Real vs. Simulated

| Operation | Type | Blockchain Involved |

| ----------------- | --------- | ------------------- |

| ETH Deposit | Real | Yes |

| ERC-20 Deposit | Real | Yes |

| ETH Withdrawal | Real | Yes |

| ERC-20 Withdrawal | Real | Yes |

| Internal Transfer | Simulated | No |

# 5. Java/web3j Mapping

## Transaction Creation

```java

RawTransaction tx = RawTransaction.createEtherTransaction(

nonce, gasPrice, gasLimit, toAddress, valueWei);

```

## Sending Transactions

```java

EthSendTransaction sentTx = web3j.ethSendRawTransaction(signedTx).send();

String txHash = sentTx.getTransactionHash();

```

## Fetching Receipts

```java

EthGetTransactionReceipt receipt = web3j.ethGetTransactionReceipt(txHash).send();

```

## ERC-20 Transfer Call

```java

Function function = new Function(

"transfer",

Arrays.asList(new Address(userAddress), new Uint256(amount)),

Collections.emptyList());

String data = FunctionEncoder.encode(function);

```