Writing to Network

Now that we've explored reading from the Solana network, let's learn how to write data to it. On Solana, we interact with the network by sending transactions made up of instructions. These instructions are defined by programs, which contain the business logic for how accounts should be updated.

Let's walk through two common operations, transferring SOL and creating a token, to demonstrate how to build and send transactions. For more details, refer to theTransactions and Instructions andFees on Solana pages.

Transfer SOL#

We'll start with a simple SOL transfer from your wallet to another account. This requires invoking the transfer instruction on the System Program.

Open Example 1#

Click thislink to open the example in Solana Playground. You'll see this code:

```
client.ts import { LAMPORTS_PER_SOL, SystemProgram, Transaction, sendAndConfirmTransaction, Keypair, } from "@solana/web3.js" ;

const sender = pg.wallet.keypair; const receiver = new Keypair ();

const transferInstruction = SystemProgram. transfer ({ fromPubkey: sender.publicKey, toPubkey: receiver.publicKey, lamports: 0.01 * LAMPORTS_PER_SOL , });

const transaction = new Transaction (). add (transferInstruction);

const transactionSignature = await sendAndConfirmTransaction ( pg.connection, transaction, [sender], );

console. log ( "Transaction Signature:" , https://solana.fm/tx{ transactionSignature }?cluster=devnet-solana , );
```

Explanation

Run Example 1#

Run the code using therun command.

Terminal run Click on the output link to view the transaction details on the SolanaFM explorer.

Output

Transfer SOL

You've just sent your first transaction on Solana! Notice how we created an instruction, added it to a transaction, and then sent that transaction to the network. This is the basic process for building any transaction.

Create a Token#

Now, let's create a new token by creating and initializing a Mint account. This requires two instructions:

- Invoke the System Program to create a new account
- Invoke the Token Extensions Program to initialize the account data

Open Example 2#

Click thislink to open the example in Solana Playground. You'll see the following code:

```
client.ts import { Connection, Keypair, SystemProgram, Transaction, clusterApiUrl, sendAndConfirmTransaction, } from "@solana/web3.js" ; import { MINT_SIZE, TOKEN_2022_PROGRAM_ID, createInitializeMint2Instruction, getMinimumBalanceForRentExemptMint, } from "@solana/spl-token" ;

const wallet = pg.wallet; const connection = new Connection ( clusterApiUrl ( "devnet" ), "confirmed" );

// Generate keypair to use as address of mint account const mint = new Keypair ();

// Calculate minimum lamports for space required by mint account const rentLamports = await getMinimumBalanceForRentExemptMint (connection);
// Instruction to create new account with space for new mint account const createAccountInstruction = SystemProgram. createAccount ({ fromPubkey: wallet.publicKey, newAccountPubkey: mint.publicKey, space: MINT_SIZE , lamports: rentLamports, programId: TOKEN_2022_PROGRAM_ID , });

// Instruction to initialize mint account const initializeMintInstruction = createInitializeMint2Instruction ( mint.publicKey, 2 , // decimals wallet.publicKey, // mint authority wallet.publicKey, // freeze authority TOKEN_2022_PROGRAM_ID , );

// Build transaction with instructions to create new account and initialize mint account const transaction = new Transaction (). add ( createAccountInstruction, initializeMintInstruction, );

const transactionSignature = await sendAndConfirmTransaction ( connection, transaction, [ wallet.keypair, // payer mint, // mint address keypair ], );

console. log ( " \n Transaction Signature:" ,https://solana.fm/tx{ transactionSignature }?cluster=devnet-solana , );

console. log ( " \n Mint Account:" ,https://solana.fm/address{ mint . publicKey }?cluster=devnet-solana , );
```

Explanation

Run Example 2#

Run the code using therun command.

Terminal run You'll see two links printed to the Playground terminal:

- One for the transaction details
- One for the newly created mint account

Click the links to inspect the transaction details and the newly created mint account on SolanaFM.

Output

Create Token

Mint Account

Notice how we built a transaction with multiple instructions this time. We first created a new account and then initialized its data as a mint. This is how you build more complex transactions that involve instructions from multiple programs.

Previous «Reading from Network Next Deploying Programs»