

ChainSafe Gaming SDK

Custom Interactions

Call Custom Blockchains

Connect to any EVM compatible blockchain by providing an RPC. All methods have an optional field to add an RPC URL.

```
1 string chain = "rootstock";
2 string network = "testnet";
3 string account = "0xD4c825203f97984e7867F11eeCc813A036089D1";
4 string rpc = "https://public-node.testnet.rsk.co";
5
6 string balance = await EVM.BalanceOf(chain, network, account, rpc);
7 print(balance);
```

Call Custom Contracts

Call will execute a smart contract method without altering the smart contract state.

Working example: <https://chainsafe.github.io/game-sendContract-example/>

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract AddTotal {
5     uint256 public myTotal = 0;
6
7     function addTotal(uint8 _myArg) public {
8         myTotal = myTotal + _myArg;
9     }
10 }
```

```
1 // set chain: ethereum, moonbeam, polygon etc
2 string chain = "ethereum";
3 // set network mainnet, testnet
4 string network = "rinkeby";
5 // smart contract method to call
6 string method = "myTotal";
7 // abi in json format
8 string abi = "[ { \"inputs\": [ { \"internalType\": \"uint8\", \"name\": \"_myArg\", \"type\": \"uint8\" } ] } ]";
9 // address of contract
10 string contract = "0x7286Cf0F6E80014ea75Dbc25F545A3be90F4904F";
```

```
11 // array of arguments for contract
12 string args = "[]";
13 // connects to user's browser wallet to call a transaction
14 string response = await EVM.Call(chain, network, contract, abi, method, args);
15 // display response in game
16 print(response);
```