Multicall

An Arbitrum Stylus version implementation ofSolidity Multi Call contract that aggregates multiple queries using a for loop and RawCall.

Example implementation of a Multi Call contract written in Rust: Here is the interface for TimeLock.

/* * *This file was automatically generated by Stylus and represents a Rust program. * For more information, please seeThe Stylus SDK*. /

// SPDX-License-Identifier: MIT-OR-APACHE-2.0 pragma

solidity

^ 0.8.23 ;

interface

IMultiCall

{ function

multicall (address [ ]

memory addresses ,

bytes [ ]

memory data)

external

view

returns

(bytes [ ]

memory) ;

error ArraySizeNotMatch ( ) ;

error CallFailed (uint256) ; }

src/lib.rs

note This code has yet to be audited. Please use at your own risk.

![cfg_attr(not(feature =

"export-abi"), no_main)] extern

crate

alloc ;

[global_allocator]

static

ALLOC :

mini_alloc :: MiniAlloc

=

mini_alloc :: MiniAlloc :: INIT ;

use

alloy_primitives :: U256 ; use

alloy_sol_types :: sol ; use

stylus_sdk :: { abi :: Bytes ,

alloy_primitives :: Address ,

call :: RawCall ,

prelude :: * } ;

[solidity_storage]

[entrypoint]

pub

struct

MultiCall ;

// Declare events and Solidity error types sol!

{ error ArraySizeNotMatch ( ) ; error CallFailed (uint256 call_index) ; }

[derive(SolidityError)]

pub

enum

MultiCallErrors

{ ArraySizeNotMatch (ArraySizeNotMatch) , CallFailed (CallFailed) , }

[external]

impl

MultiCall

{ pub

fn

multicall (& self , addresses :

Vec < Address

, data :

Vec < Bytes

,)

->

Result < Vec < Bytes

,

MultiCallErrors

{ let addr_len = addresses . len ( ) ; let data_len = data . len ( ) ; let

mut results :

Vec < Bytes

=

Vec :: new ( ) ; if addr_len != data_len { return

Err (MultiCallErrors :: ArraySizeNotMatch (ArraySizeNotMatch

{ })) ; } for i in

0 .. addr_len { let result =

RawCall :: new ( ) . call (addresses [i] , data [i] . to_vec ( ) . as_slice ( )) . map_err (| _ |

MultiCallErrors :: CallFailed (CallFailed

{ call_index :

U256 :: from (i)

})) ? ; results . push (result . into ( )) ; } Ok (results) } }

Cargo.toml

[package] name

=

"stylus-multi-call-contract" version

=

"0.1.5" edition

=

"2021" license

=

"MIT OR Apache-2.0" keywords

=

["arbitrum" ,

"ethereum" ,

"stylus" ,

"alloy"] description

=

"Stylus multi call example"

[dependencies] alloy-primitives

=

"0.3.1" alloy-sol-types

=

"0.3.1" mini-alloc

=

"0.4.2" stylus-sdk

=

"0.5.0" hex

=

"0.4.3"

[dev-dependencies] tokio

=

{

version

=

"1.12.0" ,

features

=

["full"]

} ethers

=

"2.0" eyre

=

"0.6.8"

[features] export-abi

=

["stylus-sdk/export-abi"]

[[bin]] name

=

"stylus-multi-call" path

=

"src/main.rs"

[lib] crate-type

=

["lib" ,

"cdylib"]

[profile.release] codegen-units

=

1 strip

=

true lto

=

true panic

=

"abort" opt-level

=

"s" Edit this page Previous Erc721 Next Vending Machine