How to store 10KB+ data in a Solana Account?

Asked 2 years, 11 months ago Modified 8 months ago Viewed 1k times

Ask Question

16

I am trying to store 100KB+ data on a Solana Account. In Solana documents I've read that

> "The current maximum size of an account's data is 10 megabytes."

Source: https://docs.solana.com/developing/programming-model/accounts

But when I try to create a space more than 10KB (space = 10240) it gives me error during init tests:

> "Error processing Instruction 0: Failed to reallocate account data"

Here is an example code:

```
#[derive(Accounts)]
pub struct Initialize<'info> {
    #[account(init, payer = user, space=100000)]
     pub example_account: Account<'info, ExampleAccount>,

    #[account(mut)]
    pub user: Signer<'info>,
    pub system_program: Program<'info, System>
}

#[account]
pub struct ExampleAccount {
    pub data: String,
}
```

I need to store 100KB data on that account. How can I do that?

anchor

Share Follow

asked Jul 13, 2022 at 1:21
curiosity
509 ●5 ●19

2 Answers

Sorted by: Highest score (default)

17

While the maximum account size is 10 megabytes, the maximum permitted size of a reallocation in an inner instruction is 10240 bytes. Anchor has to make a CPI to allocate the space for your account, so the maximum space it can allocate is 10240 bytes. If you want to store more than that, you will have to pass in an account you have created with a separate createAccount instruction.

If you are not initializing the account with a CPI, you would instead do:

```
#[account(zero)]
pub example_account: Account<'info, ExampleAccount>,
```

And then when you call Initialize:

```
const create_account_instruction = SystemProgram.createAccount({
    fromPubkey: user,
    newAccountPubkey: exampleAccountKey,
    space: 100000,
    lamports,
    programId: yourProgramId
});
const your_instruction = program.instruction.initialize({
    accounts: {
        exampleAccount: exampleAccountKey,
        user: user,
        systemProgram: systemProgramId
    }
});
const txn = new Transaction().add(create_account_instruction, your_instruction);
```

Share Follow

edited Oct 22, 2024 at 19:06
mikemaccana
3,624 ●1 ●16 ●43

answered Jul 13, 2022 at 2:13
Ellie High
1,114 ●3 ●8

I am confused what does "you will have to pass in an account you have created with a separate createAccount instruction." means. Could you provide an example? – curiosity Jul 13, 2022 at 3:23

Are you familiar with the solana web3 SDK? I've elaborated in my example but you will probably have to be familiar with it to understand what I'm saying. – Ellie High Jul 13, 2022 at 3:38

Don't forget, there's a new realloc method which will allow you to increase by 10kb every transaction – Henry E Jul 13, 2022 at 7:24

Hi, I have tried your code and it works fine, I have created an account with space of 500,000 bytes. My idea is to store 10,000 public keys inside a vector on this account, but with 500K space I can only store 500 pubkeys which is strange because I thought pubkeys just needed 32 bytes, so to store I would need 10,000 * 32 = 320,000 bytes. What am i doing wrong? Ty. – user3688985 Jul 13, 2022 at 18:26

What error are you having? – Ellie High Jul 13, 2022 at 20:42

Program logged: "Error: memory allocation failed, out of memory". And there are just 500 pubkeys in the vector, but the accout has 500KB of space, here's the address of the acc on devnet: 8q8rTUeW9Wn3seaYiCf7HU6bpimzQTt7zEDe3RgVteBc – user3688985 Jul 13, 2022 at 20:44

You're probably running out of heap or stack space. While your account might have 500 kb of storage, your Solana program only has 4 kb of stack space and 32 kb of heap. To work around this limitation you'll probably want to use Anchor's zero-copy feature. – Ellie High Jul 13, 2022 at 20:54

Wow, thank you very much, this is new for me.. I couldn't find many documentation about zero_copy. I tried to implement it but got an error that says "the trait `Copy` may not be implemented" for the vector type.. My approach was the following: #[account(zero_copy)] #[derive(Default)] pub struct PubkeysList { pub pubkeys: Vec<Pubkey>, } – user3688985 Jul 14, 2022 at 0:42

1 @EllieHigh When I hover on programId field it says: "Public key of the program to assign as the owner of the created account". On your example code, should I replace systemProgramId with my custom program Id? If not, why are we making system account owner of program? I am assuming if systemprogram owns that account, I won't be able to edit data inside it. Just like I can't edit my own SOL balance. – curiosity Jul 14, 2022 at 1:19

You're right, my mistake. For this you should definitely set the owner to be your program. – Ellie High Jul 14, 2022 at 21:27

@user3688985 I'd suggest making another post for that question just so people can find the answer and I can write a more in depth explanation: Yeah, ZeroCopy is very finnicky. To use zero_copy your struct has to implement the Pod trait (docs.rs/bytemuck/1.10.0/bytemuck/trait.Pod.html). This is so you can map from struct access straight to byte access in the serialized account without deserializing the whole thing. I would suggest using a [Pubkey; N] (though you will have to manage the expandable array aspect yourself). – Ellie High Jul 14, 2022 at 21:33

@EllieHigh You are right, I made a post for that question. Im still having problems getting the result I need. solana.stackexchange.com/questions/197/... – user3688985 Jul 15, 2022 at 1:32

10

It is possible to allocate a PDA with up to 10 MiB. Use the init instruction to allocate 10240 bytes, this is the maximum that is allowed to be allocated through a CPI. Then you can execute another instruction that calls `AccountInfo::realloc` for every additional 10240 bytes that you need in the account. You can group many of these together in a single transaction. Ensure you have sufficient lamports in the account before reallocating.

The other approach, described in Ellen High's answer, is to initialize an account from a new keypair that signs a transaction containing a transaction-level instruction to the system program to initialize the account up to 10 MiB, plus an instruction that transfers ownership to the program (which can be tx-level or a CPI). But often it is preferable to use a PDA.

Share Follow

edited Jul 14, 2022 at 13:37

answered Jul 13, 2022 at 17:32
Drew Nutter
236 ●1 ●7

Linked

2 How to actually store 10k Pubkeys in a Solana Account
1 How might I send an 87kb transaction to Solana?
0 Why is this function failing to fetch my program account's public key?

Related

2 Is `AccountInfo::realloc` limited to 10kb across all accounts or per account?
1 How to store signatures in a Solana account
0 The data that anchor' account stored surprise me, for a non-PDA account, it store all history input data. Why?
0 How to create data account in web3.js and read&write from custom program?
1 The program expected this account to be already initialized on InitializeCandidate
0 Attempt to load a program that does not exist
1 On Solana smart contract error # [derive(Accounts)]

Hot Network Questions

Your Answer