Blog

# Tutorial: Structuring executable messages for Assembly proposals, Part 2: Adding proxy contracts

November 3, 2022 + Technical



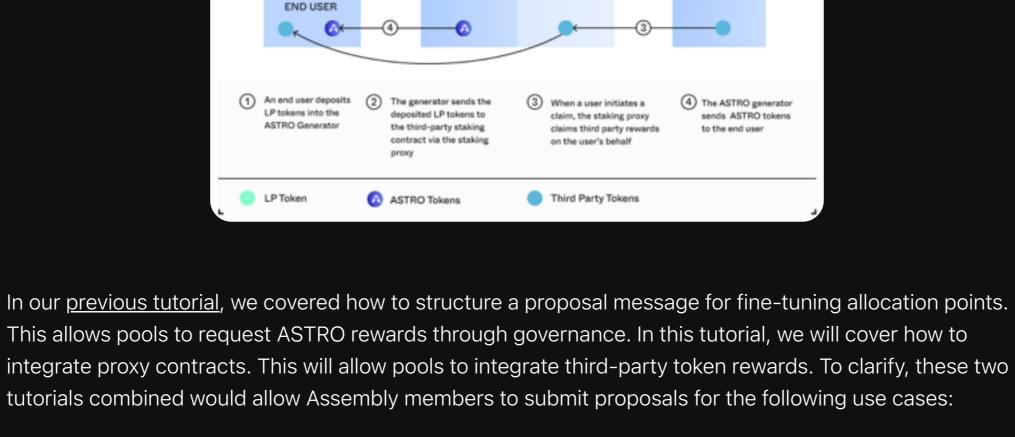
## as the rewards might differ in amount or value. Astroport addresses this problem with special, proxy-based smart contracts that enable simultaneous "dual farming" of ASTRO tokens and governance tokens coming from another community seeking to incentivize LPs.

Intro to Proxy Contracts

Rather than depositing Astroport LP tokens directly in third-party staking contracts, LPs can deposit their tokens into one of Astroport's ASTRO Generators. With the use of proxy contracts, the ASTRO Generators forward LP tokens into relevant third-party staking contracts. Therefore, LPs end up accruing two sets of rewards (both ASTRO and the third-party protocol tokens).

One of the issues that liquidity providers (LPs) face, is the decision of which platform to provide liquidity to,

ASTRO STAKING THIRD PARTY GENERATOR STAKING



Adding dual rewards to a pool (combining steps 1 and 2) Additionally, this tutorial assumes that you have 1) created a liquidity mining mechanism to distribute your governance token to LPs in the Astroport pool. And 2) that you have deployed a generator proxy contract

specific to your liquidity mining mechanism and Astroport pool.

1. Adding ASTRO rewards to a pool (no third-party token rewards)

Adding third-party token rewards to a pool (no ASTRO rewards)

For new projects that want their liquidity mining mechanism to be integrated with Astroport, the simplest path is basing your liquidity mining contracts on the Mirror design. Alternatively, Anchor's liquidity mining

And for the generator proxy contract, Astroport provides a proxy template that you can customize. Astroport also provides a full example proxy contract designed for the Mirror liquidity mining mechanism. If you're ready to integrate proxy contracts, continue on 👇

mechanism is simpler and should not require big changes to the example generator proxy contract.

Code Overview

First, the entire code:

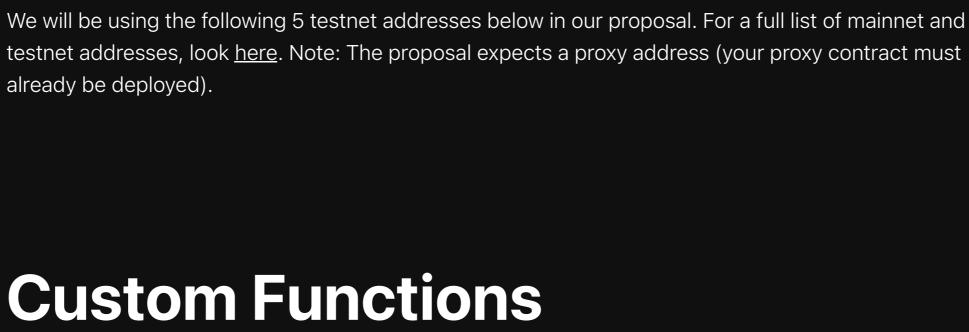
### . . . async function main() { const terra = new LCDClient({



**Contract Addresses** 

async function main() { const terra = new LCDClient({

const mk = new MnemonicKey({



We will be using 2 custom functions. The first will allow us to query any contract, the second will allow us to

1. Allow the proxy contract

2. Stake LP tokens

encode our messages in binary format.

Allow Proxies Msg (1st Msg) Currently, if you want to stake LP tokens to a proxy contract you have to:

This is done in two separate messages which will be combined in our proposal. In this section, we will focus

Next, we use our queryContract custom function to query the generator contract and return its config. The

return value is saved to a config variable, which we update by pushing our new proxy address contract.

## on the first message (allowing the proxy contract). Our msg takes in our updated config variable with our proxy contract address. We encode this message and pass it into our executable message.

**Update Generator Config** 

Move to Proxy Msg (2nd Msg) Next, we move our lp tokens to our proxy contract. We encode our message and pass it into our second executable message.

# **Proposal Msg**

and pass our binary into a "send" message that will be sent to the xASTRO contract.

**Execute and Broadcast Transaction** 

Now that we have our proxy messages, we can create a "submit\_proposal" message and pass in our

messages within an array [allow\_proxies\_msg, move\_to\_proxy\_msg]. We encode our proposal message

## That's it! Simply use the command line and node.js to execute the script and retrieve the transaction hash along with other information such as the proposal id.

complete the combined message.

Lastly, execute and broadcast the transaction.

So far in this series, we have discussed two types of proposals in isolation — finetuning allocation points (ASTRO rewards) and integrating proxy contracts (third-party token rewards). But how exactly would we implement dual rewards or both proposals at the same time?

First, make sure you go through the <u>entire previous tutorial</u> on allocation points. You will need this to

Second, you will need to modify our "messages" field under our "submit\_proposal" message. Remember,

Integrating Dual Rewards

integrating proxy contracts already requires two messages (allowing the proxy contract and staking the LP tokens in the proxy contract). We will be adding a third executable message from our previous tutorial.

You will need to add the executable message itself before your "submit\_proposal" message. The ending of your code will look something like this (look through the previous tutorial for the full message):

Third, you will need to update the order of your messages. Since there are 3 messages now, starting with the astro\_alloc\_message, you will need to update the order of allow\_proxies\_msg and move\_to\_proxies\_msg to "2" and "3" respectively.

Stay tuned for further tutorials and docs updates regarding executable messages and proposal types. + Follow <u>Astroport on Twitter</u> and subscribe to the <u>Astroport email newsletter</u> to get the latest alerts from the mothership.

Remember, Terra 2.0 and Astroport are experimental technologies. This article does not constitute

investment advice and is subject to and limited by the Astroport disclaimers, which you should review

Congrats if you've reached this far and have integrated proxy contracts and/or dual rewards.

before interacting with the protocol.

**DISCLAIMER** 

Astroport builders delay governance

Previous post

token unlocks by six months

Tutorial: Structuring executable messages for Assembly proposals, Part 1: Fine-tuning allocation points

Community

DISCORD

MEDIUM

TWITTER

TELEGRAM

Next post

TERMS OF USE

GOVERNANCE