Chainflip provides people the ability to earn a liquidity provisioning fee of 0.05% on every swap which uses the liquidity pool the user has deposited funds in.

The way Chainflip liquidity provisioning works is by having limit and range orders in a liquidity pool. While range orders are passive in nature, they are a lower class citizen compared to limit orders. Limit orders need to be actively managed however.

This managing can be done manually via the Chainflip LP Dashboard, or automatically via the Chainflip LP API. The manual approach is still in reach for retail users but requires constant babysitting to place new orders. The automated way is too complicated for retail users.

This bot automates limit orders according to a user their price preferences, automatically placing new limit orders when balance is available.

The orders are limited to the USDT/USDC/arbUSDC stablecoin pools since they do not require tracking index prices and are low risk because there is no impermanent loss risk. A dollar for a dollar.

The Github repository is currently private. **Please view the YouTube video for a live production demo** of the bot as well as the problems it solves. A promise is made to make this code public in case it wins in the hackathon.

Slides are available at [drive.google.com/file/d/1rjavro4P2e0Z0Im4BeMyipPK5Ra-mpb4](https://drive.google.com/file/d/1rjavro4P2e0Z0Im4BeMyipPK5Ra-mpb4/view) Presentation is available at [youtube.com/watch?v=iaxfLvpclaQ](https://www.youtube.com/watch?v=iaxfLvpclaQ)

**Technical Information**

This BUIDL uses:

* [LP API](https://docs.chainflip.io/lp/integrations/lp-api)
* [LP API RPCs](https://docs.chainflip.io/lp/integrations/lp-api#rpc-methods)
* [Node RPCs](https://docs.chainflip.io/lp/integrations/lp-rpcs)

To simplify setup for a user, Docker has been used to bundle the LP API in a preconfigured way with the public Chainflip Node RPC. Thanks to this a user only has to provide their keys to run the Liquidity Bot, and not have to set up an LP API themselves.

The RPC methods are used to retrieve a user their available undeployed balance. Based on the available balance and the configured minimum order amount it then proceeds to place a limit order.

This limit order takes into account the user their share preference between the USDT/USDC and arbUSDC/USDC pools. (solUSDC can easily be added when it launches). Additionally the limit order also respects the configured tick price for each pool.

When an order is placed, a Telegram bot informs the user about this as well.