

# Build a Wallet Tracker on Solana

What we're building:

Wallet Tracker / Balance Monitor on Solana.

In this short guide, we're going to learn how to create a wallet tracker on Solana with [Helius](#). We'll be watching a number of wallets and alert when they transfer more than 50% of their Solana holdings.

## Setting up the Webhooks

First, we need to setup webhooks. There are two ways to setup your webhooks with Helius:

- Through the Helius UI
- Programatically via the Helius API

Doing it via the UI is much quicker and doesn't require any coding experience, however it can't be updated dynamically. Doing it via the API is much more flexible as you can dynamically add/remove accounts to watch for. We'll use the UI for this tutorial for simplicity.

## Prerequisite: Setting up a Server for the Webhooks

Before we can get started, you must first have a server that will accept calls from Helius webhooks. Go ahead and skip this step if you already have one setup. If you don't, you can easily set one up for testing purposes on [Replit](#). Here is some sample Node JS code to get you started ASAP:

With the server setup, you can proceed to building the actual webhook.

## Setting up Webhooks via Helius UI

First, go to the [Helius Dev Portal](#), connect your Solana wallet, and generate an API key if you haven't done so before. Once you've generated a key, click on the "Webhooks" tab on the left side menu. To proceed, you must have got access to webhooks — if you don't have access, please message us on

[Discord

](<http://discord.gg/6GXdee3gBj>).

Now you can simply click "New Webhook" (shown below).

When you create a new webhook, you'll see three prompts: transaction types, webhook URL, and account addresses.

- Transaction types let you specify which transactions to watch for. We've went ahead and built parsers for a number of common transaction types including NFT mints, NFT sales, NFT listings, DeFi Swaps, Token Transfers, Solana Transfers and much more. You can also select "Any" to watch for all transactions.
- Account addresses let you specify which accounts you'd like to watch. Note that you can dynamically modify which accounts you watch via our Webhook APIs, but not through the UI.
- Webhook URL lets you specify the endpoint which we'll call when the webhook condition is satisfied.

Go ahead and fill these fields out in the UI.

Now, whenever there is a transfer of SOL that involves the address

8cRrU1NzNpjL3k2BwjW3VixAcX6VFc29KHr4KZg8cs2Y

— Helius will hit our test Replit server with the corresponding transaction. To see the exact payload, please refer to.

## Handling the Webhook Call

Once a transfer involving our configured address occurs, our webhook server will be called with that transaction. Let's write some code to handle that case. We want to invoke some alerting function (could be Pager Duty, could be an email, could be a Dialect message) whenever our watched account loses more than 90% of its holdings.

## Conclusion

And that's it! We were able to get a balance monitor up and running with only a few lines of code in a few minutes. For any questions (or corrections) about this guide, please message us on [Discord

](<http://discord.gg/6GXdee3gBj>).

We can't wait to see what you build!